Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

Допущено до захисту
Завідувач кафедри
комп'ютерних наук,
кандидат технічних наук, доцент
Ірина Вікторівна Машкіна

_____
« ___ » _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «бакалавр»
Спеціальність 122 Комп'ютерні науки
Освітня програма 122.00.01 Інформатика

Тема: РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ РЕЄСТРАЦІЇ БЕЗРОБІТНИХ В ЦЕНТРІ ЗАЙНЯТОСТІ ТА ІНТЕГРАЦІЯ З БАЗОЮ ДАНИХ ВАКАНСІЙ

Виконав
студент групи Інб-1-21-4.0д
Соломаха Олександр Віталійович

_____ _____

Науковий керівник
Кандидат педагогічних наук
С. М. Співак

_____

Київ – 2025

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики Кафедра комп'ютерних наук «Затверджую»

Завідувач кафедри комп'ютерних наук,
кандидат технічних наук,доцент
Ірина Вікторівна Машкіна

_____

« ___ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Виконавець – студент групи Інб-1-21-4.0д,
Соломаха Олександр Віталійович
спеціальності: 122 «Комп'ютерні науки»,
освітньої програми: 122.00.01 Інформатика

1. **Вихідні дані:** Законодавство та нормативно-правові акти України в інформаційній сфері та за напрямом дослідження, зокрема, Закон України "Про електронні документи та електронний документообіг" від 22 травня 2003 № 851-IV , "Про загальнообов'язкове державне соціальне страхування на випадок безробіття" : Закон України від 02.03.2000 № 1533-III : станом на 19 груд. 2024 р., Закон України "Про зайнятість населення" від 05.07.2012 № 5067-VI : станом на 1 січ. 2025 р., Постанова Каб. Міністрів України від 19.09.2018 № 792 : станом на 10 трав. 2023 р.; Наказ М-ва розвитку економіки, торгівлі та сіл. госп-ва України від 16.12.2020 № 2663, Постанова Каб. Міністрів України від 05.03.2014 № 90 : станом на 12 берез. 2020 р.

2. **Основні завдання:**

- Здійснити аналіз сучасних тенденцій розвитку процесів працевлаштування та найму, а також, існуючих методів роботи Державних центрів зайнятості. Дослідити технічні виклики,

обґрунтувати вибір концепції програмного додатку інформаційної системи регіонального центру зайнятості та засобів його розробки.

- Спроєктувати та реалізувати програмний додаток інформаційної системи регіонального центру зайнятості, провести його тестування задля визначення коректності його функціонування.

3. Пояснювальна записка: Обсяг – до 45 стор. формату А4 комп’ютерного набору з дотриманням вимог стандарту і методичних рекомендацій кафедри.

4. Графічні матеріали: за потребою.

5. Додатки: за потребою.

6. Строк подання роботи на кафедру: «___» _____ 2025 р.

Науковий керівник: Виконавець:

Кандидат педагогічних наук Соломаха О.В.

Співак Світлана Михайлівна

РЕФЕРАТ бакалаврської роботи

Кваліфікаційна робота: 88 с., 29 рис., 3 табл., 40 посилань.

Актуальність: впровадження цифрових послуг значно покращить роботу державних центрів зайнятості та зробить їхні послуги доступнішими та зручнішими для користувачів, що як наслідок призведе до автоматизації роботи цієї державної структури, а саме до збільшення кількості працевлаштованих безробітних та скорочення витрат України на допомогу такій категорії громадян.

Мета роботи: дослідити сучасні тенденцій розвитку процесів працевлаштування та найму, спроєктувати та програмно реалізувати інформаційну систему регіонального центру зайнятості.

Об'єкт дослідження: процеси автоматизація працевлаштування шукачів роботи за сприяння Державних центрів зайнятості засобами сучасних інформаційних технологій.

Предмет дослідження: розробка програмного додатку, який автоматизує процеси співпраці безробітних - шукачів роботи, як з Державним центром зайнятості, так і з роботодавцями.

Завдання:

- Здійснити огляд публікацій за темою роботи.
- Здійснити аналіз існуючих методів роботи державних центрів зайнятості.
- Обґрунтувати вибір концепції програмного додатку та засобів його розробки.
- Спроектувати та реалізувати програмний додаток, провести його тестування для оцінки його функціонування.

ОСНОВНІ РЕЗУЛЬТАТИ:

- Спроектовано та реалізовано програмний додаток на основі NextJs з використанням серверних API роутів Next.js та NextAuth

для аутентифікації, проведено тестування коректності функціонування програмного додатку.

- побудована база даних, яка містить інформацію про безробітних, роботодавців, вакансії та направлення. Також побудовані ER-діаграма бази даних, діаграма активності системи та макети основних інтерфейсних вікон програми. Розроблено інтерфейс користувача системи, який складається з таких основних форм: вхід, головна сторінка, облік осіб, вакансії, роботодавці та направлення. Окремо розроблені додаткові форми для всіх побічних форм.

Практичне значення дослідження: Впровадження створеного web-застосунку забезпечить підвищення рівня цифровізації центрів зайнятості України та сприятиме автоматизації ключових процесів у межах державної політики зайнятості населення.

Ключові слова: АВТОМАТИЗАЦІЯ, ДЕРЖАВНА СЛУЖБА ЗАЙНЯТОСТІ, ДІДЖИТАЛІЗАЦІЯ, ПРОЦЕС ПРАЦЕВЛАШТУВАННЯ, РЕГІОНАЛЬНИЙ ЦЕНТР ЗАЙНЯТОСТІ, NextJs.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ……..8
ВСТУП……..9
1 ДОСЛІДЖЕННЯ СУЧАСНИХ ТРАНСФОРМАЦІЙ У СФЕРІ ПОШУКУ РОБОТИ ТА ПІДБОРУ ПЕРСОНАЛУ ……... 12
1.1 Тренди сучасного ринку праці ……..12
1.2 Основні завдання Державної служби зайнятості……. 18
1.3 Основні механізми роботи регіонального центру зайнятості……. 21
1.4 Огляд наявних цифрових інструментів найму та керування персоналом..24
1.5 Постановка задачі дослідження…….30
Висновки до розділу 1……...32
2 ТЕХНОЛОГІЇ ТА ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕГІОНАЛЬНОГО ЦЕНТРУ ЗАЙНЯТОСТІ……..34
2.1 Обґрунтування вибору мови програмування TypeScript для розробки веб-додатку……34
2.2 Архітектура та компонентний підхід React для створення інтерактивних інтерфейсів……35
2.3 Використання Next.js як повнофункціонального фреймворку для повного стеку веб-розробки…… 37
2.4 Застосування Prisma як сучасного ORM для типобезпечної взаємодії з базами даних…….. 38
2.5 Використання PostgreSQL як масштабованої та надійної системи управління базами даних…….. 40
2.6 Реалізація системи аутентифікації та авторизації за допомогою NextAuth.js……..42
2.7. Оптимізація та стратегії розгортання веб-додатків на основі Next.js……44
Висновки до розділу 2……...46

3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕГІОНАЛЬНОГО ЦЕНТРУ ЗАЙНЯТОСТІ……49
3.1 Структура ІС РЦЗ…… 49
3.1.1. Загальна інформація про структуру web-додатку…….49
3.1.2 Навігація та ролі користувачів…….50
3.1.3 Наповнення (контент).....51
3.1.4 Дизайн…….....51
3.2 Проектування та створення бази даних ІС РЦЗ.....52
3.3 Проектування серверної частини ІС РЦЗ.....63
3.3.1 Архітектура API маршрутів Next.js.....66
3.3.2 Структура серверної частини.....67
3.3.3 Реалізація доступу до даних через Prisma.....65
3.4 Створення клієнтської частини розробленого додатку інформаційної системи регіонального центру зайнятості……...67
3.4.1 Реалізація функціональності для різних ролей користувачів……...67
3.4.2 Взаємодія з API та авторизація……68
3.5 Запуск та тестування розробленого програмного продукту.....70
3.5.1 Вимоги до функціонування web-додатку 70
3.5.2 Чек-лист функціональних вимог.....70
3.5.3 Заплановані тест-кейси.....71
3.5.4 Реалізовані тест-кейси.....76
Висновки до розділу 3…….....80
ВИСНОВКИ ……..82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ……..84

ПЕРЕЛІК СКОРОЧЕНЬ

Служба - Державна служба зайнятості

Голова Служби - Голова Центрального апарату Служби

ІТ-компаніях - Компанія у сфері інформаційних технологій

ІС РЦЗ - Інформаційна система регіонального центру зайнятості

РЦЗ - районний центр зайнятості

Фонд - Фонд загальнообов‘язкового державного соціального страхування України на випадок безробіття

ШІ - штучний інтелект HR - Human Resources

ВСТУП

У сучасному світі ринок праці динамічно розвивається, роботодавці створюють нові робочі місця, і як наслідок, є постійна потреба у кваліфікованих фахівцях, які повинні відповідати вимогам нового цифрового суспільства. Сьогодні існують численні способи пошуку персоналу, проте Державні центри зайнятості України і до тепер не втрачають важливої ролі у цьому процесі, оскільки, крім обліку безробітних та нарахування державної допомоги, надають послуги з пошуку роботи, профорієнтації та перепідготовки кадрів. Однак традиційні методи роботи цих центрів, засновані на паперовому документообігу та ручному опрацюванні інформації, наразі є застарілими та не відповідають потребам цифрового суспільства та меті українського уряду створити цифрову економіку та максимально діджиталізувати державні сервісні функції [7, 18, 26]. Нагадаємо, що основними завданнями Державної служби зайнятості України є:

- реалізація державної політики у сфері зайнятості населення та трудової міграції, соціального захисту від безробіття;

- здійснення аналізу стану ринку праці;

- сприяння громадянам у підборі відповідної до кваліфікації роботи;

- надання роботодавцям послуг з добору працівників;

- участь в організації проведення громадських та інших робіт тимчасового характеру;

- сприяння громадянам в організації підприємницької діяльності;

- участь у реалізації заходів, спрямованих на запобігання масовому вивільненню працівників, профілактика настання страхового випадку, сприяння мобільності робочої сили та зайнятості населення;

- організація підготовки, перепідготовки та підвищення кваліфікації безробітних з урахуванням поточної та перспективної потреб ринку праці, підтвердження результатів неформального професійного навчання;

- проведення професійної орієнтації населення;

- додаткове сприяння у працевлаштуванні окремих категорій громадян, які неконкурентоспроможні на ринку праці [23].

Безсумнівно, що впровадження автоматизації послуг може значно покращити роботу Державних центрів зайнятості та зробити їхні послуги доступнішими та зручнішими для користувачів, що як наслідок призведе до підвищення ефективності їхньої роботи, а саме до збільшення кількості працевлаштованих безробітних та скорочення витрат України на допомогу такій категорії громадян, що і зумовлює **актуальність** обраної для бакалаврської роботи теми.

Метою кваліфікаційної бакалаврської роботи дослідити сучасні тенденцій розвитку процесів працевлаштування та найму, спроєктувати та програмно реалізувати інформаційну систему регіонального центру зайнятості.

Відповідно до поставленої мети сформульовано **наступні завдання** дослідження:

1. Здійснити аналіз сучасних тенденцій розвитку процесів працевлаштування та найму, а також, існуючих методів роботи Державних центрів зайнятості.

2. Дослідити технічні виклики, обґрунтувати вибір концепції програмного додатку інформаційної системи регіонального центру зайнятості та засобів його розробки.

3. Спроєктувати та реалізувати програмний додаток інформаційної системи регіонального центру зайнятості, провести його тестування задля визначення коректності його функціонування.

Об’єкт дослідження: процеси автоматизація працевлаштування шукачів роботи за сприяння Державних центрів зайнятості засобами сучасних інформаційних технологій.

Предмет дослідження: розробка програмного додатку, який автоматизує процеси співпраці безробітних - шукачів роботи, як з Державним центром зайнятості, так і з роботодавцями.

Методи дослідження: Теоретичною основою дослідження став огляд наукової літератури та систематизація даних кар'єрних платформ, що дозволило визначити найоптимальніші способи розгортання програмного рішення. У процесі дослідження було проведено порівняльний аналіз технологічних інструментів веб-розробки. Для клієнтської частини було обрано Next.js та React з використанням системи стилізації CSS Tailwind, що пропонує типобезпечну розробку та адаптивний інтерфейс залежно від корпоративного стилю. Серверна частина була реалізована за допомогою Next.js API Routes з системою ORM Prisma для операцій з базами даних та PostgreSQL. Система автентифікації була реалізована за допомогою NextAuth, що пропонує гнучкі механізми авторизації користувачів. Програмне забезпечення було розроблено та розгорнуто за допомогою інтегрованого середовища розробки Cursor. Структура системи була розроблена відповідно до концепції розподілу відповідальності, а об'єднання отриманих результатів дозволило розробити інтегровану платформу, що враховує потреби шукачів роботи, роботодавців та кар'єрних консультантів.

Структура та обсяг роботи. Бакалаврське дослідження складається 88 сторінок, 3 Розділів, містить 3 таблиці, 29 рисунки, 37 джерел.

1 ДОСЛІДЖЕННЯ СУЧАСНИХ ТРАНСФОРМАЦІЙ У СФЕРІ ПОШУКУ РОБОТИ ТА ПІДБОРУ ПЕРСОНАЛУ

1.1 Тренди сучасного ринку праці

Аналізуючи сучасні тенденції рекрутингу, окреслимо мегатренди, які мають значний вплив на процеси, що відбулися на світовому ринку праці у 2024 році та збережуть тенденції у 2025 році.

Рецесія провідних національних економік у європейських країнах, як от у Німеччині, та пов’язані з цим ускладнення соціально-економічної ситуації, вплив демографічних факторів, створюють нові умови, з якими доводиться стикатися міжнародним та регіональним компаніям. Як наслідок, постає актуальне питання: як компаніям та корпораціям адаптувати власні стратегії найму та збереження персоналу в часи світової невизначеності та глобальних змін, щоб успішно конкурувати, знайти та утримати кваліфікованих робітників?

Аналіз дозволяє визначити п’ять трендів рекрутингу станом на 2024 рік [29]:

Тренд рекрутингу 1: використання штучного інтелекту і автоматизація з метою запобігання професійного вигорання;

Тренд рекрутингу 2: пошук талантів на глобальному рівні, що призводить до мобільності фахівців та розвитку міжнародного рекрутингу;

Тренд рекрутингу 3: врахування досвіду роботи фахівців та залучення таких співробітників нетрадиційними засобами, як от опціони в ІТ-компаніях;

Тренд рекрутингу 4: наймання на основі суміжних кваліфікації та орієнтація на навчання впродовж життя;

Тренд рекрутингу 5: абстрагування від популярних раніше обмежень, що впливали на найм працівників, а саме гендерних, вікових, локаційних тощо.

Проаналізуємо кожен з вище означених трендів.

Тренд рекрутингу 1: використання штучного інтелекту і автоматизація з метою запобігання професійного вигорання.

Глобальний дефіцит кваліфікованих фахівців в останні роки підвищив вимоги рекрутерів до пошукачів у процесі заповнення ними відкритих вакансій, проте одночасно призвів так званого “професійного вигоранням”, адже наймані працівники часто працюють понаднормово, у вихідні та святкові дні, оскільні отримані завдання необхідно виконати у стислі терміни. Саме тому автоматизація та використання штучного інтелекту (ШІ) під час верифікації первинних документів, поданих пошукачем, і далі у процесі планування співбесід і зв’язку потенційного кандидата з роботодавцем є необхідністю сучасності на шляху спрощення роботи рекрутів у довгостроковій перспективі. Безсумнівно, діджиталізація вже сьогодні полегшує та прискорює обробку персональних даних, що зменшує робоче навантаження, економить час та запобігає простою компаній, і як наслідок втрати конкурентоспроможності та прибутків. Саме тому на часі автоматизація рутинних дії засобом ШІ, який оптимізує та вивільнить час рекрутів. Згідно з опитуванням, рекрутери витрачають в середньому 30 годин на тиждень на рутинні завдання, як от написання оголошень про наявні вакансії або надсилання підтвердження про отримання заявок [28]. Так наприклад, фахівці Stepstone Austria (платформи з пошуку роботи) переконані, що рутинну діяльність можна зробити ефективною саме за рахунок автоматизації, адже “зв'язок потенційних кандидатів з відповідною до їхньої кваліфікації посадою та компанією матиме вирішальне значення в майбутньому. Саме тому в The Stepstone Group інвестуються значні кошти в інтелектуальні технології, які пришвидшують встановлення зв’язку компанії з потрібними шукачами роботи та звільняють рекрутерів для виконання стратегічних завдань і вирішення міжособистісних питань” [29].

Тренд рекрутингу 2: пошук талантів на глобальному рівні, що призводить до мобільності фахівців та розвитку міжнародного рекрутингу. Упродовж останніх років міжнародний рекрутинг набуває дедалі більшого значення, що зумовлено зростанням цінності багатоманітності професійних підходів та спеціалізованого досвіду. У країнах із високим рівнем економічного розвитку, зокрема Західної Європи та Північної Америки, спостерігається зростаюча залежність від імміграції кваліфікованих кадрів, що є відповіддю на демографічні виклики, зокрема старіння населення та дефіцит робочої сили. В умовах глобалізованої та цифрової економіки конкуренція за висококваліфікованих фахівців суттєво загострюється. Компанії дедалі частіше виходять за межі національних ринків праці у пошуках найкращих спеціалістів, що сприяє розширенню практики цільового міжнародного рекрутингу. Цей процес передбачає цілеспрямований підбір персоналу з-за кордону для заповнення вакансій, які потребують специфічної експертизи. Такий підхід не лише задовольняє потреби роботодавців у висококваліфікованих кадрах, а й відкриває нові можливості для фахівців із країн, що розвиваються. Вони отримують шанс на релокацію та працевлаштування за кордоном, де рівень оплати праці та якість життя є значно вищими, ніж на батьківщині. Таким чином, міжнародний рекрутинг виступає ефективною відповіддю на виклики сучасного ринку праці, одночасно задовольняючи інтереси як роботодавців, так і фахівців у глобальному контексті.

Відповідно до поточного дослідження спеціалізованої платформи Stepstone [29] 23 відсотки компаній Австрії вже сьогодні використовують міжнародний рекрутинг як стратегію боротьби з нестачею кваліфікованих фахівців. Зазначимо, що міжнародний пошук працівників має також низку неочевидних переваги: так закордонні пошукачі отримали освіту у своїх країнах, яка часто має свої позитивні переваги, та далі набули професійний досвід на національному ринку праці, і як наслідок пропонують

високоефективні новації та креативні ідеї, підвищуючи інноваційну складову компанії, на яку вони починають працювати, адже “ця передача ноу-хау, методів роботи та ідей є рушійною силою розвитку компанії” [29]. “Міжнародний підхід також оптимізує відносини компанії за кордоном, саме тому, що співробітники, які знають країну та мову, можуть налагодити позитивні відносини з іноземними партнерами” [29]. Варто підкреслити, що релоковані спеціалісти виступають важливими агентами у процесі інтернаціоналізації бізнесу, зокрема під час виходу компаній на нові ринки або розширення їхньої присутності за кордоном. Їхній професійний досвід та культурна обізнаність сприяють більш ефективній адаптації організацій до умов зовнішнього ринку, що, у свою чергу, дозволяє оптимізувати витрати на етапі масштабування. Формування мультикультурної, висококваліфікованої команди позитивно впливає на репутацію роботодавця, демонструючи його відкритість до культурного розмаїття та прихильність до принципів інклюзії й толерантності. Такі ціннісні орієнтири підвищують привабливість компанії як роботодавця, сприяють залученню нових фахівців і зміцненню лояльності вже існуючого персоналу, що є ключовим чинником стабільного організаційного розвитку.

Тренд рекрутингу 3: врахування досвіду роботи фахівців та залучення таких співробітників нетрадиційними засобами, як от опціони в IT-компаніях. Зазначимо, що протягом багатьох років професійний досвід співробітників є центром уваги в роботі HR, адже цей досвід відображає професійний шлях співробітника взагалі і у певні компанії зокрема - від найму, зростання по професійній ієрархії компанії, до звільнення. Аналіз чинників, що призводять до звільнення фахівця, дозволяє HR-команді зробити рекомендації щодо створення середовища в межах компанії, в якому професіонали почувалися б комфортно, що є важливою передумовою для їхньої подальшої роботи, та дозволяє зберегти цінних фахівців. Однією з провідних тенденцій сучасного управління персоналом є концепція залучення працівників (employee

engagement), яка дедалі більше визнається як критичний фактор організаційного успіху. Компанії, що формують емоційний зв’язок із персоналом шляхом визнання індивідуального внеску співробітника у розвиток підприємства, та водночас забезпечують не лише конкурентоспроможну систему оплати праці, але й сприятливе робоче середовище, здатні ефективно утримувати кваліфіковані кадри у довгостроковій перспективі. Цілеспрямовані стратегії залучення персоналу сприяють формуванню позитивного організаційного клімату, що, у свою чергу, забезпечує стійке зростання продуктивності, інноваційності та рівня задоволеності працівників. Таким чином, високий рівень залученості є не лише показником внутрішньої згуртованості колективу, а й індикатором стратегії сталого розвитку компанії. Згідно з даними дослідження *Stepstone Engagement Study*, ключовими детермінантами формування емоційного залучення працівників є, з одного боку, надання можливостей для безперервного професійного розвитку та навчання, а з іншого — ефективна комунікація з боку керівництва, що передбачає чітке, послідовне та позитивне бачення стратегічних орієнтирів організації [29].

Тренд рекрутингу 4: наймання на основі суміжних кваліфікації та орієнтація на навчання впродовж життя. Прагнучи розширити резерв пошуку талантів, значна кількість великих компаній знижують вимоги до обов’язковості вищої освіти у пошукачів та визначають у противагу необхідні практичні навички та вміння, набуті впродовж попередньому професійного досвіду, а також враховується навчання, отримане альтернативним методами. Слід очікувати, що такий підхід та вимоги до пошукачів у подальшому стануть нормою, адже це не лише сприятиме інклюзивності, але й підвищуватиме здатність компаній до інновацій та адаптації. Крім того, наразі з’явилася нова тенденція: залучання фахівці не тільки виключно з професійним досвід роботи відповідно до відкритої вакансії, а й оцінка потенціалу пошукача щодо здатності навчатися та розвиватися у професійній

сфері, адже знайти ідеального фахівця часто важко, довготривалий пошук та очікування є нерентабельними, у противагу до того, що у короткій перспективі певний пошукач набуде необхідних компанії навичок та вмінь. Саме тому зростає важливість цільових інвестиції в навчання та програми розвитку персоналу, адже часто для пошукача це є важливим фактором у виборі місця роботи саме в певній компанії. Процеси перекваліфікації та підвищення кваліфікації персоналу відіграють ключову роль у стратегічному розвитку будь-якої організації. Інвестування в професійне зростання співробітників сприяє масштабуванню бізнесу, підвищенню операційної ефективності та, як наслідок, зростанню доходів як компанії, так і її працівників. Такий підхід водночас формує позитивний імідж роботодавця, який демонструє цінність людського капіталу та взаємозв'язок між розвитком компанії й професійною еволюцією її фахівців. Це, у свою чергу, сприяє залученню нових висококваліфікованих кадрів та зміцненню мотивації вже залучених професіоналів [29].

Тренд рекрутингу 5: абстрагування від популярних раніше обмежень, що впливали на найм працівників, а саме гендерних, вікових, локаційних тощо. Незважаючи на те, що різноманітність в усіх сферах, рівні можливості та інтеграція є центром уваги у процесі найму фахівців протягом останніх років, у майбутньому ця різноманітність продовжуватиме набувати більшого значення з огляду на дефіцит кваліфікованих фахівців у найбільших національних економіках світу, як от у США, Німеччини тощо. Інтернаціоналізація робочого середовища, збільшення можливостей працювати віддалено, залучення жінок до нетипових професій - це те, що відбувається сьогодні. Так 78 відсотків опитаних згідно звіту Stepstone Diversity Report зазначили [29], що хотіли б працювати в середовищі, складовою якого є відкритість та різноманітність. У сучасних умовах глобального ринку праці принцип різноманітності набуває стратегічного значення як для процесів найму, так і для вибору місця роботи. З огляду на

демографічні виклики, зокрема старіння населення у високорозвинених країнах, їхній подальший економічний розвиток значною мірою залежить від залучення кваліфікованих кадрів із третіх країн. Водночас специфіка культурних норм, релігійних переконань та соціальних практик таких фахівців нерідко суттєво відрізняється від традиційних європейських цінностей. У зв'язку з цим виникає об'єктивна необхідність формування толерантного підходу до індивідуальних відмінностей працівників, з акцентом на їхню професійну компетентність, а не на етнокультурні або інші особистісні характеристики. Прийняття принципів інклюзії та підтримка різноманітності дозволяють створити сприятливе середовище для продуктивної міжкультурної взаємодії та командної співпраці. У майбутньому робочі колективи ставатимуть усе більш різноманітними за такими ознаками, як вік, гендер, національність, релігія тощо, однак визначальним критерієм залишатиметься рівень професійної ефективності та здатність до інтегративної взаємодії задля забезпечення сталого та інноваційного розвитку організацій. "У рекрутингу це, зокрема, означає: відбір персоналу на основі компетенцій і навичок, а не зв'язків" [29]. "Ця реорганізація забезпечує ефективний і орієнтований на майбутнє відбір персоналу, який відповідає різноманітним вимогам робочого світу багатьох поколінь" [29]. Отже, глобальний ринок праці загалом, а ринок праці України, зокрема, в перспективі повинен врахувати вище зазначені тенденції, до яких додаються і зовнішні чинники, як от політичні події в різних куточках світу, війни, природні катаклізми, які також враховуються компаніями. Усе це призводить до необхідності адаптувати стратегії найму персоналу до вищезазначених подій.

1.2 Основні завдання Державної служби зайнятості

Україна та її ринок праці є частиною глобального ринку, тому усі зазначені вище тенденції стосуються і нас. Безумовно, існують різноманітні

способи пошуку роботи, проте найбільш очевидним і простим є звернення до Державного центру зайнятості, структурні підрозділи якого представлені регіональними представництвами в усіх територіальних громадах України. Зазначимо, що усі регіональні представництва Державного центру зайнятості працюють за єдиною схемою надання послуг. Клієнти - пошукачі роботи - можуть звернутися до будь-якого з них та отримати всі передбачені законодавством соціальні послуги, пов'язані з допомогою по безробіттю, перенавчанням та подальшим працевлаштуванням, адже існує уніфікована оперативна база вакансій для пошукачів роботи та можливості проходження професійного навчання по всій країні. Саме це дозволяє розширити можливості пошуку роботи для безробітних не тільки в межах району чи області, а й всій державі в цілому [23].

Розкриємо особливості функціонування типового районного центру зайнятості. Отже, це державна установ у централізованій системі державних установ державної служби зайнятості (далі - РЦЗ) [22]. Саме РЦЗ є головною установою в первінній роботі з безробітними та шукачами роботи в певному районі. РЦЗ у своїй діяльності керується Конституцією та Законами України, постановами Верховної Ради України, прийнятими відповідно до Конституції та Законів України, актами Президента України, Кабінету Міністрів України та Міністерства соціальної політики України, рішеннями правління Фонду загальнообов'язкового державного соціального страхування України на випадок безробіття (далі – Фонд), іншими нормативно-правовими актами, наказами і дорученнями Голови Центрального апарату Служби (далі – Голова Служби) та директорами обласного центру зайнятості, а також Положенням РЦЗ [4].

РЦЗ є неприбутковою державною установою, фінансування якої відбувається коштом Фонду. Доходи, отримані РЦЗ відповідно до частини першої статті 16 закону України "Про загальнообов'язкове державне соціальне страхування на випадок безробіття" [19] використовуються

виключно на цілі, визначені частиною другою цієї статті. РЦЗ як юридична особа публічного права має самостійний баланс, рахунки в органах Державної казначейської служби України та банківських установах, кутовий штамп і печатку із зображенням Державного Герба України, своїм найменуванням та ідентифікаційним кодом згідно з ЄДРПОУ, інші печатки та штампи (за потреби) [22].

РЦЗ виконує завдання і функції у сфері зайнятості населення, трудової міграції та соціального захисту від безробіття, а також функції виконавчої дирекції Фонду загальнообов‘язкового державного соціального страхування України на випадок безробіття, визначені Законами України “Про загальнообов’язкове державне соціальне страхування на випадок безробіття” [19], “Про зайнятість населення” [20], до внесення відповідних змін до цих законів (пункт 2 постанови Кабінету Міністрів України від 05 березня 2014 року № 90 “Деякі питання державного управління у сфері зайнятості населення” [6]) .

РЦЗ, викуючи покладені на нього завдання, взаємодіє з іншими органами виконавчої влади, органами місцевого самоврядування, підприємствами, установами та організаціями. Керівником РЦЗ є директор, призначення та звільнення якого Головою Служби та погоджується з головою відповідної районної державної адміністрації. Структура та штатний розпис РЦЗ затверджується директор обласного центру зайнятості відповідно до затверджених директором обласного центру зайнятості граничної чисельності. Відомості про лінейно-функціональну структуру Служби відображені в наступній Таблиці 1.1.

Таблиця 1.1

Структура РЦЗ

Заступник директора – технолог
Начальник відділу організації працевлаштування населення
Начальник відділу активної підтримки безробітних

Начальник відділу взаємодії з роботодавцями
Юрисконсульт

Дослідження ефективності діяльності служби зайнятості відбуваються постійно, що уможливлює її вдосконалення. Окреслимо коло тем наукових розвідок щодо шляхів збільшення її ефективності за останні роки:

- удосконалення структури Державного центру зайнятості та показники ефективності структурних підрозділів [4; 5; 6; 15; 16];

- шляхи вдосконалення рейтингового оцінювання та організації системи ідентифікації потреб незайнятих осіб [2; 7];

- висвітлення методів оцінювання ефективності діяльності служб зайнятості в європейських країнах [24; 25; 27].

- питання осучаснення роботи таких установ [11; 14; 24; 25];

- проблеми працевлаштування певних категорій громадян [9; 10; 11];

- проблеми функціонування центрів зайнятості в період невизначеності [1; 11];

- проблеми автоматизації діяльності [5;12; 28; 30].

1.3 Основні механізми роботи регіонального центру зайнятості

Можна констатувати, що жодній державі державі не вдалося досягти 100 % зайнятості населення: безробіття є складовою ринку праці, яке визначено як “соціально-економічне явище, за якого частина осіб не має змоги реалізувати своє право на працю та отримання заробітної плати (винагороди) як джерела існування” [20]. Чинним законодавством визначенопорядок процедури реєстрації, перереєстрації безробітних та ведення обліку пошукувачів робот, який здійснюють регіональні центри Державної служби зайнятості. Також визначені учасники цього процесу, а саме “суб’єкти ринку праці - особи, які шукають роботу, зареєстровані

безробітні, працівники, роботодавці, що отримують послуги, передбачені цим законами" [20] та шляхи їхньої співпраці [20].

Під час проведення реєстрації працівник центру зайнятості заповнює в Єдиній інформаційно-аналітичній системі державної служби зайнятості персональну картку, в якій зазначаються персональні дані безробітного (прізвище, ім'я та по батькові, зареєстроване місце проживання чи перебування, число, місяць та рік народження, реєстраційний номер облікової картки платника податків або серія та номер паспорта (для осіб, які через свої релігійні переконання відмовилися від прийняття реєстраційного номера облікової картки платника податків, повідомили про це відповідний орган державної податкової служби і мають відповідну відмітку в паспорті), відомості про останнє місце роботи або вид діяльності згідно із частиною першою статті 4 Закону України "Про зайнятість населення", підстава для припинення трудових відносин відповідно до запису в трудовій книжці або іншого виду діяльності, що підтверджено документально. Форма персональної картки та додатки до неї затверджуються Мінсоцполітики [20].

Структуру роботи РЦЗ можна відобразити наступною схемою «(рис.1.1)».

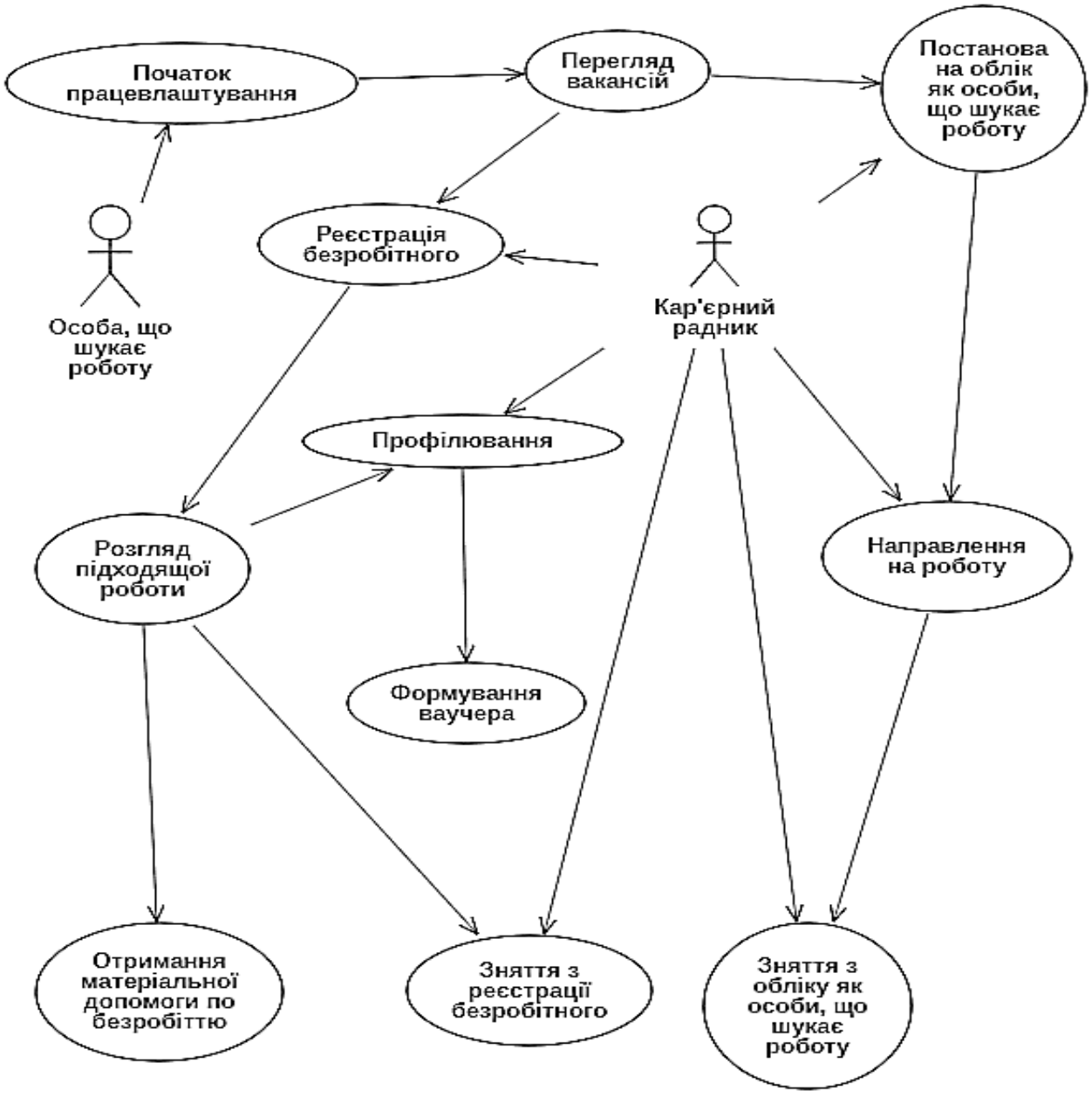

Рисунок. 1.1. Загальна діаграма прецедентів всіх подій, пов’язаних з роботою центру зайнятості (Use Case)

Отже, аналіз чинного законодавства та підзаконних актів та документів дозволяє виділити наступні функції Державної служба зайнятості:

- інформування зацікавлених громадян про вільні вакансії на підприємствах та установах усіх форм власності;

- облік осіб, що шукають роботу;

- державна реєстрація безробітних у Єдиній інформаційно-аналітичної системи державної служби зайнятості з метою нарахуванням матеріальної допомоги відповдно до чинного законаодвства ;

- акумулювання вакансій, поданих роботодавцями усіх форм власності та посередництво між усіма суб'єктами ринку праці ;

- допомога шукачам роботи, у тому числі у перекваліфікації, підвищення кваліфікації, відкритті власної справи тощо.

Таким чином, сучасні регіональні центри зайнятості виконують не лише функцію прямого працевлаштування, а й практично реалізують державну політику у сфері зайнятості населення. Враховуючи вищезазначене, для ефективної та продуктивної роботи Державній службі зайнятості загалом, а регіональним центрам зокрема, необхідно збирати та обробляти великі масиви інформації, а саме:

- інформацію про безробітних;
- інформацію про вакансії;
- інформацію про роботодавців;
- інформацію про надані направлення;
- інформацію про кар’єрних радників.

Саме тому питання автоматизації збору та обробки інформації є нагальним та потребує подальшого удосконалення з метою діджиталізації рутиних процесів працівників регіональних центрів зайнятості, яким необхідно щоденно співпрацювати з усіма суб'єктами ринку праці.

1.4 Огляд наявних цифрових інструментів найму та керування персоналом

Проаналізуємо існуючі системи управління персоналом, які створені для співпраці суб'єктів ринку праці. Пропонована компанією Colorisoft програма Coloris HRM є системою управління персоналом, що дозволяє

автоматизувати процеси в сфері HR, на всіх етапах працевлаштування та звільнення [31].

Основний функціонал від Coloris HRM [31]:

- ведення кадрового обліку;
- управління наймом;
- звільнення та переведення;
- управління вакансіями;
- онбординг;
- навчання та розвиток;
- оцінка та атестація;
- управління талантами.

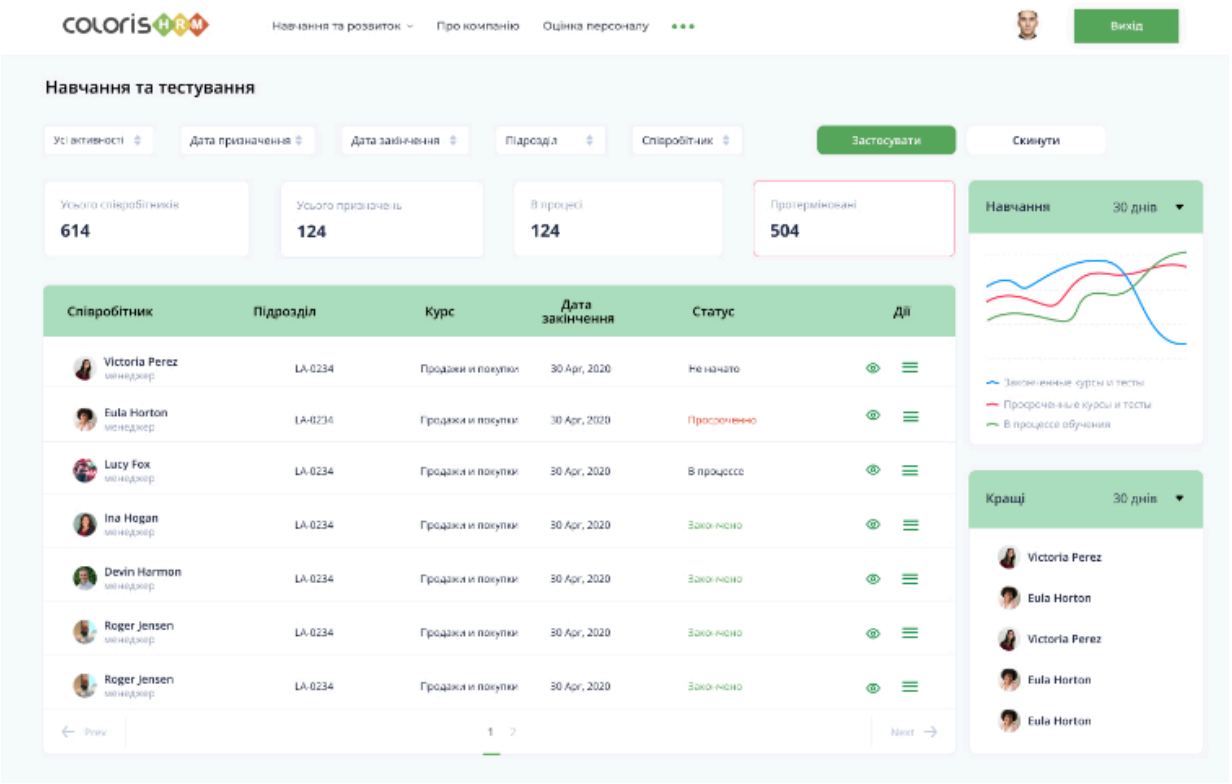

Рисунок. 1.2. Скріншот роботи програми Coloris HRM [31]

Аналіз пропонованої системи «(рис.1.2)» дозволяє виділити наступні переваги Coloris HRM:

- автоматизує рутинні HR-задачі, оптимізуючи час роботи HR-фахівцям для реалізації стратегічних завдань;

- оптимізує HR-процеси, призводячи до підвищення ефективності роботи як HR-команди, так і компанії загалом, зменшуючи оперативні витрати;

- ефективний доступ до персональних даних співробітників, що уможливлює прийняття кращих управлінських рішень.

Сучасною та достатньо універсальною програмою рекрутингу є сервіс Ripplin [35], який є оптимальним для середніх за чисельністю компаній (від 75 до 2000 співробітників). Серед функціоналу надається система відстеження кандидатів (ATS), керування зарплатою та бонусами, моніторинг робочого часу, керування онбордингом і підвищення кваліфікації, а також інструментарій для професійної організації роботодавців. Зазначимо, що сервіс позиціонує, що використання його призведе до зростання продуктивності, адже автоматизує ланку рутинних адміністративних завдань.

Рекрутери, що обирали Rippling, звертають увагу на його багатофункціональність та гнучкість налаштувань, що робить цей сервіс одним кращих програмних рішень для HR «(рис.1.3)».

Рисунок. 1.3. Скріншот роботи [35]

Аналіз сервісу дозволяє виділити такі переваги:

- в один клік автоматичний збір даних відпрацьованого часу співробітників, його обробка, з подальшим розрахунком заробітної плати та заповнення податкових декларації;

- гнучка автоматизація робочих процесів на основі тригерів різного порядку;

- уможливлює дистанційне керування менеджерами пристроями підлеглих з метою встановлення або оновлення програмного забезпечення на них;

- вбудований модуль навчання персоналу інтегрований з іншими hr-сервісами платформи;

- уможливлює автоматичний облік кандидатів (ATS), які надіслані резюме на оголошення про вакансії;

- функція онлайн-онбордингу та хмарного зберігання документів;

- розширені інструменти звітності та аналітики.

Слід зазначити, що Державна служба зайнятості, орієнтуючись на сучасні підходи цифровізації державного управління, почала впроваджувати електронні сервіси як одну з провідних форм надання соціальних послуг населенню та роботодавцям через цифрові засоби, тому в 2018 році в межах цієї стратегії було запущено інноваційну платформу - web-сайт “Єдине соціальне середовище зайнятості” [36], який:

- уможливив самостійну реєстрацію безробітних засобом інформаційних технологій [17].

- генерує базу поданих вакансій та резюме по всій території України;

- надає зареєстрованим клієнтам особисті кабінети для самообслуговування [17];

- дозволяє зареєстрованим роботодавцям подавати звітність онлайн [8], пропонує доступ до актуальної аналітики та статистики ринку праці;

- передбачає можливість проходження профорієнтаційного тестування [37] та функціонує на комп’ютерах та на мобільних пристроях із системами Android та Apple.

Головною метою запровадження цифрових сервісів є підвищення якості обслуговування та зручності у взаємодії з державною службою зайнятості.

Зокрема, “Електронний кабінет роботодавця” дозволяє зменшити потребу у фізичному відвідуванні центрів зайнятості, забезпечуючи дистанційний доступ до інформації про реквізити підприємства, перебіг заповнення вакансій, направлення кандидатів, укладені договори тощо. Також передбачена можливість подавати статистичну звітність у реальному часі.

Аналогічно, безробітні, зареєстровані у системі, можуть створювати резюме, шукати роботу, дізнаватися про програми навчання та тимчасового працевлаштування, а також отримувати детальну інформацію про візити до центру, заходи служби зайнятості та фінансову допомогу.

Платформа “Єдине соціальне середовище зайнятості” частково реалізує функцію автоматичного поєднання роботодавців та безробітних, але в повному сенсі автоматичний підбір кандидатів **(**як у сучасних системах на основі штучного інтелекту) поки що є обмеженим.

Функціональні можливості платформи:

- пошук за фільтрами: платформа дозволяє роботодавцям переглядати резюме кандидатів за заданими критеріями (професія, регіон, досвід тощо);
- підбір вакансій для безробітного: система надає зареєстрованим користувачам список відповідних вакансій на основі вказаних характеристик;
- автоматизована аналітика: доступна базова статистика щодо вакансій, запитів, зайнятості тощо.

Недостатній функціонал для сучасного цифрового суспільства:

- відсутність інтелектуального матчінгу (matching) за глибокими критеріями, такими як м’які навички, поведінкові патерни або рекомендації системи;

- відсутність автоматичного надсилання пропозицій без участі людини (наприклад, якщо роботодавець створив вакансію, система автоматично розсилає її відповідним кандидатам);

- самонавчальних алгоритмів, які покращують результат з часом. Після аналізу можна констатувати, що повна автоматизація процесу працевлаштування та автоматичне поєднання шукачів роботи і роботодавців ще не досягнуто, що ми і пропонуємо зробити.

У Таблиці 1.2. узагальнимо можливості проаналізованих аналогів.

Таблиця 1.2.

Порівняння HRM-систем та цифрових платформ для працевлаштування в Україні та світі

Характеристика	**Rippling**	**Coloris HRM**	**Єдине соціальне середовище зайнятості**
Походження	США	Україна	Україна (державна ініціатива)
Модулі	HR, IT, фінанси, управління пристроями	HR, рекрутинг, оцінка, навчання	Резюме, вакансії, електронні кабінети, послуги зайнятості
Інтеграції	500+ додатків (Slack, Zoom, Salesforce тощо)	Обмежена кількість, локальні системи	Немає інтеграцій із зовнішніми комерційними системами
Локалізація	Глобальна	Повна локалізація під український ринок	Український ринок
Підтримка української мови	немає	є	є
Цінова політика	Комерційна, помодульна оплата	Гнучка, адаптована до українського ринку	Безкоштовна (державна платформа)
Цільова аудиторія	Бізнес (від малого до великого)	Українські компанії та HR-відділи	Громадяни України та роботодавці

Що робить система	Автоматизує HR, IT, фінанси, пристрої, доступи	Оптимізує HR-процеси, оцінює персонал, навчає	Допомагає у пшуку роботу, розміщує вакансії, подає заявки
Платформа/дост уп	Хмарна, web + мобільна	Web-платформа	Web-платформа + мобільна адаптація
Складність впровадження	Середня / Висока (потребує тех. налаштувань)	Середня (вимагає супровід впровадження)	Низька (реєстрація через BankID або код доступу

1.5 Постановка задачі дослідження

Грунтуючись на вищезазначеному, визначимо завдання на розробку власного програмного продукту.

Назва продукту: Програмний комплекс для центрів зайнятості.

Мета: Автоматизація процесу реєстрації безробітних в центрі зайнятості та інтеграції з базою даних вакансій з метою підвищення ефективності працевлаштування громадян.

Цільовою аудиторією є працездатні особи у віці від 18 до 60 років, які проживають в Україні та мають доступ до інтернету.

Кінцевий продукт - web-сайт, що має зручний інтерфейс та численні сервісні функції. Розроблюваний web-додаток має бути загальнодоступним у мережі Інтернет. Права доступу до інформації розмежовані за групами користувачів: адміністратор, кар’єрний радник, особа, яка шукає роботу, та роботодавець.

Програмний додаток матиме зазначені нижче функціональні можливості. Ведення бази даних:

- зберігання інформації про безробітних: анкетні дані, освіта, досвід роботи, навички, шукані вакансії;

- зберігання інформації про вакансії: опис вакансії, вимоги до кандидатів, контактні дані роботодавця;

- зберігання інформації про роботодавців: реєстраційні дані, контактні дані, сфера діяльності, кількість вакансій;

- зберігання інформації про надані направлення: дата направлення, вакансія, безробітний, результат направлення;

- зберігання інформації про кар’єрних радників: анкетні дані, контактні дані, спеціалізація.

Пошук вакансій:

- можливість пошуку вакансій за різними критеріями: назва вакансії, ключові слова, сфера діяльності, місце роботи, рівень зарплати, тип зайнятості (постійне, тимчасове, стажування);

- персоналізований підбір вакансій на основі анкетних даних та навичок безробітного.

Постановка на облік осіб, що шукають роботу:

- реєстрація безробітних онлайн або за допомогою кар’єрного

радника;

- заповнення анкетних даних безробітного;

- призначення кар’єрного радника.

Реєстрація вакансій:

- можливість розміщення вакансій роботодавцями онлайн або за допомогою кар’єрного радника;

- заповнення інформації про вакансію;

- підтвердження вакансії кар’єрним радником;

Надання направлення на вакансії:

- підбір вакансій, що відповідають персональним даним, компетенціям та практичним навичкам безробітного; - направлення безробітного на вакансію; - відстеження результатів направлення. Звітність:

- формування звітів про кількість зареєстрованих безробітних, кількість вакансій, кількість виданих направлень, результати працевлаштування;

Інші можливості: - онлайн-консультації з кар’єрним радником; - навчальні курси та семінари з пошуку роботи; - бібліотека матеріалів з пошуку роботи. Технічні вимоги. Програмний комплекс має бути:

- розроблений з використанням сучасних технологій та інструментів;
- сумісний з різними веб-браузерами та операційними системами;
- захищений від несанкціонованого доступу;
- простим у використанні та мати зручний інтерфейс.

Очікувані результати від використання web-додатку:

- збільшення кількості працевлаштованих громадян;
- зниження рівня безробіття;
- підвищення ефективності роботи центрів зайнятості;
- покращення якості надання послуг центрами зайнятості.

Висновки до розділу 1

Аналіз сучасних тенденцій у сфері працевлаштування та рекрутингу, проведений у контексті 2024-2025 років, виявив ключові напрями трансформації управління людськими ресурсами, які відображають глобальні виклики та технологічний прогрес. Застосування штучного інтелекту, автоматизація процесів, акцент на навички, глобальна мобільність талантів, а також увага до досвіду працівників і їхньої залученості стають основними рушіями змін. Водночас управління віковою та соціокультурною різноманітністю, а також концепція навчання впродовж життя набувають дедалі більшого значення. Ці тенденції вказують на необхідність адаптації організацій до нових реалій, де гнучкість, інклюзивність і технологічна інтеграція відіграють вирішальну роль у формуванні ефективних стратегій рекрутингу та утримання кадрів.

Окремий акцент у дослідженні зроблено на діяльності регіональних центрів зайнятості (РЦЗ) та аналізі програмного забезпечення, що підтримує процеси працевлаштування. РЦЗ виконують важливі функції, такі як облік безробітних, надання інформації про вакансії та сприяння працевлаштуванню, однак потребують подальшої автоматизації для підвищення ефективності. Платформи, такі як “Єдине соціальне середовище

зайнятості”, Coloris HRM і Rippling, демонструють різні підходи до автоматизації HR-процесів, але мають обмеження, зокрема щодо локалізації, складності впровадження чи відсутності автоматичного зіставлення вакансій і кандидатів. Отримані результати аналізу створюють підґрунтя для розробки вдосконалення інформаційної системи РЦЗ, яка враховуватиме сучасні тенденції та локальні потреби, сприяючи ефективнішому управлінню процесами працевлаштування.

2 ТЕХНОЛОГІЇ ТА ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕГІОНАЛЬНОГО ЦЕНТРУ ЗАЙНЯТОСТІ

2.1 Обґрунтування вибору мови програмування TypeScript для розробки веб-додатку

TypeScript — це статично типізований надмножинний набір для JavaScript, який надає кодерам можливість створювати будь-які типи безпечних та стабільних веб-додатків. За допомогою TypeScript можна створювати клієнтські веб-додатки, серверні API, мікросервіси та інтерактивні інтерфейси користувача. Інструменти редактора коду, такі як Visual Studio Code, включають вбудований редактор коду, вбудовану підтримку TypeScript, вбудований налагоджувач та багато, багато іншого для полегшення створення додатка.

Розглянемо ключові можливості TypeScript, а саме:

1. Статична типізація.

- Система типів, яка перевіряється під час компіляції.
- Інтерфейси для визначення контрактів між фрагментами коду.
- Дженерики для створення компонентів, що можуть бути повторно використані та адаптовані.
- Перевірка сумісності типів, щоб уникнути помилок під час виконання.

2. Об'єктно-орієнтовані функції.

- Класи з модифікаторами доступу: public, private, protected.
- Успадкування та поліморфізм для гнучких структур класів.
- Абстрактні класи та абстрактні методи для визначення контрактів реалізації.
- Метапрограмування та аспектна орієнтація за допомогою декораторів.

3. Додаткові функції.

- Стрілкові функції зберігають контекст this.
- Деструктуризація для легшої роботи з масивами та об'єктами.

- Асинхронне програмування з async/await.
- Типи об'єднання та перетину для визначення розширених структур даних.

4. Інструменти розробки маю такі властивості:

- Інтеграція з IDE через службу мов.
- IntelliSense для автодоповнення коду.
- Рефакторинг та автоматичне виправлення помилок
- Перевірка коду та виявлення помилок на етапі розробки.

Процес звітування про помилки під час компіляції оптимізовано під час розробки для підвищення якості коду. TypeScript перетворює код на звичайний JavaScript, так що його можна було використовувати на будь-якій платформі, що підтримує JavaScript. Процес компіляції за допомогою tsconfig.json забезпечує гнучкість в ефективній обробці процесу компіляції, включаючи встановлення цільового ECMAScript, суворість перевірки типів та інші конфігурації.

2.2 Архітектура та компонентний підхід React для створення інтерактивних інтерфейсів

React — високопродуктивна, багаторазова та декларативна бібліотека для створення користувацьких інтерфейсів на JavaScript, яку підтримує та розробляє Facebook (Meta), та базується на компонентах, з акцентом на багаторазові компоненти користувацького інтерфейсу та створення складних інтерфейсів.

Уточнимо архітектурні елементи React, серед яких важливими є:

1. Декларативна парадигма

- Оголошення бажаного стану користувацького інтерфейсу замість імперативної маніпуляції DOM.
- Автоматичне оновлення інтерфейсу при зміні даних.
- Передбачувана поведінка компонентів, що спрощує налагодження.

2. Компонентний підхід:

- Атомарні, незалежні блоки інтерфейсу.
- Композиція компонентів для побудови складних інтерфейсів користувача.
- Інкапсуляція стану та логіки в компонентах.
- Спільне використання компонентів між кількома частинами програми.

3. Односторонній потік даних:

- Спільне використання даних від батьківських компонентів до дочірніх компонентів за допомогою props.
- Регульований метод обробки форм.
- Передбачуваний потік даних, що покращує налагодження.
- Керування станом за допомогою React Hooks (useState, useReducer). *Віртуальний DOM*:
- Покращений процес оновлення для живого DOM.
- Об'єднання оновлень для підвищення продуктивності.
- Алгоритм узгодження зменшує кількість операцій DOM.
- Використання архітектури Fiber для асинхронного рендерингу.

4. Модель реактивного оновлення:

- Компоненти автоматично відповідають на зміни стану або властивостей.
- Оптимізація за допомогою memo, useMemo та useCallback
- Правильна обробка життєвого циклу компонента за допомогою useEffect.

Можливість вказувати власні перехоплення для захоплення логіки React тепер використовує функціональні компоненти та перехоплення, що спрощує написання та підтримку коду порівняно з компонентами класів. Перехоплення дозволяють використовувати стан та інші функції React без необхідності написання класів, що спрощує написання та тестування коду. React не передбачає конкретних шаблонів для обробки станів або структури коду, і ви можете використовувати будь-який інший метод відповідно до

вимог проекту - від локального стану до складних менеджерів станів, таких як Redux, Zustand або Jotai.

2.3 Використання Next.js як повнофункціонального фреймворку для повного стеку веб-розробки

Next.js — це високорівневий фреймворк для React, який забезпечує багату інфраструктуру для створення високопродуктивних веб-додатків. Він додає серверний рендеринг, генерацію статичних даних, маршрутизацію файлової системи та інші оптимізації одразу після встановлення.

Уточнимо загальні функції Next.js для користувача:

1. Гібридний метод рендерингу
- Генерація статичного сайту (SSG) для статичного контенту.
- Серверний рендеринг (SSR) для динамічних сторінок.
- Інкрементальна статична регенерація (ISR) для кешування з обмеженим часом.
- Клієнтський рендеринг багатих, інтерактивних елементів.
- Потокова передача компонентів для покращеного UX.
2. Маршрутизація додатків (маршрутизація на основі файлової системи):
- Інтуїтивна організація маршрутів через структуру каталогів.
- Вкладені макети для повторюваних елементів інтерфейсу користувача.
- Паралельні шляхи для одночасного відображення кількох сторінок.
- Перехоплення маршрутів модального вікна та спеціальні взаємодії.
- Групування маршрутів для організації семантичного коду, яка не впливає на URL-адреси.
3. Апаратне забезпечення та операції сервера.
- Розділення логіки клієнта та сервера.
- Мінімізований JavaScript, що доставляється клієнту.
- Прямий доступ до ресурсів сервера (наприклад, бази даних, файлової системи).

- Операції сервера для обробки форм без API.
- Реактивні оновлення інтерфейсу користувача через мутації сервера

4. Оптимізація та продуктивність:

- Автоматична оптимізація зображень та шрифтів.
- Підтримка найновішого формату зображень WebP/AVIF.
- Попереднє завантаження ресурсів для швидшої навігації.
- Автоматичне розділення коду для зменшення розміру пакета.
- Додана інтеграція ESLint для якості коду.
- Оптимальна компіляція для успішного впровадження

5. API та взаємодія з даними:

- Кінцеві точки API на стороні сервера для оголошення кінцевих точок REST або GraphQL.
- ПЗ проміжного шару для обробки запитів та відповідей.
- Включає вбудовану підтримку CORS та захист CSRF.
- Кешування на рівні маршруту та валідація.
- Оптимізація API для зовнішніх компонентів за допомогою Fetch API.

Слід зазначити, Next.js ідеальний для комплексного стеку розробки, оскільки дозволяє писати код для фронтенду та бекенду в одному проекті, що спрощує написання програми, її тестування та остаточне розгортання, а також дозволяє тісніший зв'язок між різними частинами системи. Архітектура Next.js розроблена для створення сучасних веб-застосунків, що відповідають сучасним вимогам продуктивності, доступності та SEO. Vercel підтримує Next.js та надає оптимізовану платформу для розміщення застосунку у випадку Next.js.

2.4 Застосування Prisma як сучасного ORM для типобезпечної взаємодії з базами даних

Prisma — це ORM нового покоління на Node.js та TypeScript, пропонує типобезпечний доступ до бази даних та спрощує процес створення

веб-застосунків на стороні сервера. Prisma відрізняється від інших ORM тим, що вона є декларативною у визначенні схеми даних, а також має потужний API для запитів.

Уточнимо ключові характеристики Prisma:

1. Декларативна схема даних та міграції:
- Мова схеми Prisma (PSL) для явного визначення моделей даних.
- Строго типізовані поля моделі та зв'язки між моделями.
- Автоматичне створення та виконання міграцій зі змін схеми.
- Історія міграцій для відстеження змін у схемі бази даних.
- Підтримка перейменування таблиць та полів без втрати даних
2. Типобезпечний клієнт бази даних:
- TypeScript клієнт автоматично генерується зі схеми
- Повна підтримка автодоповнення в IDE.
- Перевірка типів під час компіляції.
- Запобігання атакам SQL-ін'єкцій та невідповідності типів.
- Статичний аналіз запитів для виявлення проблем
3. Продуктивний API для запитів та маніпулювання даними:
- Fluent API для створення складних запитів.
- Можливість звуження запитів за допомогою певних полів.
- Вкладені операції та запити для маніпулювання пов'язаними даними.
- Підтримується групування даних та агрегації.
- Розширена фільтрація з підтримкою логічних операторів.
4. Транзакції та атомарні операції:
- Підтримка транзакцій ACID.
- Вкладені транзакції для складних операцій.
- Атомні операції для узгодженості даних.
- Оптимізація з'єднань з базою даних шляхом пулінгу.
- Обробка помилок з підтримкою транзакцій для відкату.
5. Інтеграція з Next.js та подібними фреймворками:

- Плавна робота серверних компонентів Next.js.
- Оптимальна поведінка сервера та використання кінцевих точок API.
- Підтримка GraphQL за допомогою Nexus або type-graphql.
- Кешування результатів для покращення продуктивності.
- Управління для оптимального використання ресурсів.

Отже, Prisma значно спрощує роботу з базами даних у проектах TypeScript, забезпечуючи високу безпеку типів та уникаючи поширених помилок через невідповідність типів або неправильні запити. У декларативному підході схема бази даних є єдиним джерелом достовірності для схеми, а код бази даних та код програми залишатимуться синхронізованими. Prisma Studio — інтерактивна консоль для редагування та аналізу даних — надає розширені функції розробки та налагодження, дозволяючи візуалізувати та редагувати дані бази даних кінчиками пальців, без необхідності писати SQL-запити.

2.5 Використання PostgreSQL як масштабованої та надійної системи управління базами даних

PostgreSQL як передова об'єктно-реляційна система керування базами даних з відкритим кодом відповідає стандартам та розширюється для SQL. PostgreSQL з моменту створення понад 30 років тому перетворився на просунуту та багатофункціональну систему зберігання даних. Ми зосередимося на архітектурних та функціональних аспектах PostgreSQL:

1. Цілісність та надійність даних:

- Суворе дотримання специфікацій ACID (атомірність, узгодженість, ізоляція, довговічність)
- Реєстрація попередньої запису (реєстрація перед писанням).
- Відновлення в момент часу (відновлення в момент часу).
- Асинхронна та синхронна реплікація для високої доступності.
- Запобігання пошкодженню даних через збій обладнання

2. Розширена система типів даних та структур даних:
- Розширена підтримка типів даних (масиви, JSON/JSONB, XML).
- Геопросторові типи даних для ГІС-додатків (PostGIS).
- Визначені користувачем типи та домени даних
- Підтримуються UUID, HSTORE, перечислителі та діапазони.
- Повний текстовий пошук з підтримкою лінгвістичних функцій
3. Оптимізація та продуктивність:
- Різні типи індексів (B-дерево, хеш, GiST, SP-GiST,)
- Паралельні запити для багатоядерних систем.
- Оптимізатор запитів на основі вартості.
- Розділення таблиці для обробки великих обсягів дані.
- Матеріалізовані представлення для кешування результатів запитів.
4. Програмованість та розширюваність:
- Збережені процедури та функції різними мовами програмування (PL/pgSQL, PL/Python, PL/Perl тощо).
- Тригер для автоматичної відповіді на зміни даних.
- Система розширень для додавання нових функцій.
- Правила та політики доступу для реалізації складної логіки безпеки.
- Користувацькі агрегатні та віконні функції
5. Безпека та контроль доступу:
- Детальна система контролю доступу на основі ролей (RBAC).
- Шифрування даних на різних рівнях.
- SSL/TLS для безпечного з'єднання.
- Аутентифікація за допомогою різних методів (пароль, сертифікат, LDAP).
- Аудит та ведення журналу для відстеження дій
6. Інтеграція з сучасними технологіями:
- Оболонка зовнішніх даних для підключення до зовнішніх джерел даних.

- Слухачі та асинхронні повідомлення (LISTEN/NOTIFY).
- Підтримка JSON для інтеграції з архітектурами NoSQL.
- Логічна реплікація для інтеграції з іншими СУБД.
- Масштабування читання за допомогою підказок для читача

Отже, PostgreSQL ідеально поєднується з Prisma, яка забезпечує типобезпечний доступ до бази даних та оптимізує SQL-запити для максимальної продуктивності. Таке поєднання технологій дозволяє ефективно працювати з реляційними даними, зберігаючи безпеку типів та продуктивність. Порівняно з іншими СУБД, PostgreSQL пропонує оптимальний баланс між надійністю, функціональністю та продуктивністю, що робить його чудовим вибором для веб-застосунків будь-якого розміру, від стартапів до високонавантажених корпоративних систем.

2.6 Реалізація системи аутентифікації та авторизації за допомогою NextAuth.js

NextAuth.js - бібліотека аутентифікації з відкритим кодом, що широко застосовується, для проектів Next.js, яка дозволяє легко використовувати різні форми автентифікації без необхідності створювати власну інфраструктуру. Вона постачається з готовою реалізацією для контролю доступу, токенів та сесій.

Зупинимося на архітектурі та функціях NextAuth.js:

1. Система багатопровайдерної автентифікації

- Інтеграція з популярними соціальними мережами (Google, Facebook, Twitter, GitHub).
- Підтримка провайдерів OAuth 1.0 та OAuth 2.0
- Автентифікація через електронну пошту та пароль.
- Автентифікація без пароля за допомогою магічних посилань.
- Підтримка сертифікатів клієнта та сервера
- Можливість вказувати власних провайдерів

2. Безпечна обробка сесій
- Підтримка JWT (JSON Web Tokens) для безсерверної архітектури.
- Сесії, що підтримуються базою даних, для розширених функцій.
- Шифрування конфіденційної інформації.
- Автоматичне оновлення сесій та токенів.
- Налаштовувані механізми зберігання сесій (sessionStorage, localStorage, cookie).
3. Авторизація та контроль доступу:
- Система ролей та прав.
- Безпека маршрутів на сервері та на клієнті.
- Проміжне програмне забезпечення контролю доступу до сторінок.
- Умовний рендеринг компонентів залежно від стану автентифікації.
- Контекст автентифікації для доступу до інформації користувача
4. Безпека та захист від атак:
- Захист від CSRF (міжсайтових запитів)
- Запобігання XSS (міжсайтових скриптів)
- Захист від різних форм підробки сеансу.
- Суворі налаштування безпеки файлів cookie.
- Обмеження на кількість спроб входу.
5. Інтеграція з маршрутизатором додатків Next.js:
- Серверні модулі для перевірки автентифікації
- Клієнтські перехоплювачі для доступу до даних сеансу.
- Проміжне програмне забезпечення для захисту групи маршрутів.
- Збільшена кількість дій сервера з розширеною інтеграцією.
- Інтегрована підтримка альтернативних режимів рендерингу (SSR, SSG, ISR).
6. Розширюваність та гнучкість:
- Зворотні виклики для модифікації процесу аутентифікації.
- Події, пов'язані зі змінами стану автентифікації

- Адаптери для інших баз даних (Prisma, MongoDB, Firebase).
- Налаштування компонентів інтерфейсу користувача.
- Підтримка багатьох мов та локалізації.

Отже, NextAuth.js забезпечує безпечну та просту аутентифікацію користувачів, щоб розробники могли зосередитися на створенні функцій програми, а не на складній системі безпеки. Модуль регулярно оновлюється для відповідності сучасним вимогам безпеки, а також для сумісності з новими. технології.

Зазначимо, що NextAuth.js у поєднанні з Prisma та PostgreSQL дозволяє створити цілу систему користувачів із безпечним сховищем облікових даних, обробкою сеансів та контролем доступу для різних частин програми.

2.7. Оптимізація та стратегії розгортання веб-додатків на основі Next.js

Розробка сучасного веб-додатку вимагає не лише якісного коду, але й оптимальних стратегій розгортання та доставки контенту. Next.js надає сучасні інструменти для оптимізації продуктивності та спрощення процесу розгортання.

Зробимо огляд переваг щодо комплексного підходу до оптимізації та розгортання:

1. Оптимізація веб-вітальних показників (Web Vitals):

- Автоматична оптимізація зображень (розміри, формати, відкладене завантаження).
- Оптимізація шрифтів з попереднім завантаженням та відображенням системних шрифтів.
- Стратегії попереднього завантаження ресурсів для критичного шляху рендерингу.
- Розділення коду для зменшення розміру початкового пакета JavaScript.

- Оптимізація метрик LCP, FID, CLS для покращення користувацького досвіду.

2. Стратегії кешування та доставки контенту:
    - Статична генерація для кешованого контенту.
    - Інкрементальна статична регенерація для обмеженого в часі кешування.
    - Стратегії кешування на рівні браузера з HTTP-заголовками.
    - Інтеграція з CDN для географічно розподіленої доставки контенту.
    - Оптимізація на рівні Edge для швидкої доставки динамічного контенту.
3. Масштабування та надійність:
    - Безсерверна архітектура для автоматичного масштабування.
    - Контейнеризація через Docker для однорідних середовищ.
    - Системи оркестрації (Kubernetes) для управління великими розгортаннями.
    - Балансування навантаження для розподілу запитів.
    - Стратегії відмовостійкості та самовідновлення.
4. Автоматизація процесу розгортання:
    - Конвеєри CI/CD для автоматичного тестування та розгортання.
    - Перевірки якості коду та продуктивності перед розгортанням.
    - Автоматичне розгортання пов'язане з гілками Git (Preview Deployments).
    - Поетапне розгортання для мінімізації ризиків..
    - Автоматичне відкачування змін у випадку проблем.
5. Моніторинг та аналітика:
    - Збір метрик продуктивності у реальному часі.
    - Відстеження помилок та виключень.
    - Аналіз використання ресурсів та оптимізація витрат.

- Моніторинг доступності та швидкодії по регіонах.
- Аналітика користувацького досвіду та поведінки

6. Оптимізація для пошукових систем (SEO):
    - Генерація статичних метаданих для пошукових систем.
    - Автоматичне створення карт сайту та robots.txt.
    - Структуровані дані для розширеного відображення в результатах пошуку.
    - Оптимізація для соціальних мереж (Open Graph, Twitter Cards).
    - Прогресивне покращення для забезпечення доступності контенту.

Слід зазначити, що платформи, як от Vercel від Next.js, надають оптимізоване середовище розгортання, налаштоване для Next.js додатків, пропонуючи автоматичну оптимізацію на рівні Edge, глобальне розгортання через CDN та безсерверну архітектуру, що суттєво спрощує процес запуску та масштабування веб-додатків. Саме комплексний підхід до оптимізації та розгортання забезпечує високу продуктивність, доступність та масштабованість веб-додатків, побудованих на Next.js, що є критичним фактором для сучасних цифрових продуктів.

Висновки до розділу 2

Розглянуті технології TypeScript, React, Next.js, Prisma, PostgreSQL та NextAuth.js формують сучасний та ефективний стек для розробки високопродуктивних веб-додатків. Проаналізовано та доведено переваги кожної технології та їх синергетичний ефект при спільному використанні.

TypeScript забезпечує надійність та безпеку коду через статичну типізацію, зменшуючи кількість помилок виконання та покращуючи підтримуваність коду. React пропонує гнучкий та ефективний підхід до створення інтерактивних інтерфейсів завдяки компонентній архітектурі та декларативній парадигмі.

Next.js розширює можливості React, додаючи оптимізації для продуктивності, серверний рендеринг та статичну генерацію, що значно покращує початкове завантаження та SEO-оптимізацію. Архітектура App Router з серверними компонентами дозволяє ефективно розділяти серверну та клієнтську логіку.

Prisma забезпечує типобезпечний доступ до бази даних PostgreSQL, спрощуючи процес взаємодії з даними та запобігаючи поширеним помилкам. PostgreSQL, як потужна та надійна система управління базами даних, забезпечує високу продуктивність, масштабованість та розширену функціональність для зберігання та обробки даних.

NextAuth.js надає комплексне рішення для аутентифікації та авторизації, інтегруючись з Next.js та Prisma для створення безпечної та гнучкої системи управління користувачами. Оптимізація та стратегії розгортання, специфічні для Next.js, забезпечують високу продуктивність та доступність веб-додатків.

Порівняльний аналіз з альтернативним стеком технологій (ASP.NET Core/Angular) показав, що обраний стек має переваги в контексті швидкості розробки, оптимізації продуктивності, SEO та гнучкості розгортання. Хоча ASP.NET Core з Angular є потужним рішенням для корпоративних додатків з високими вимогами до безпеки та інтеграції з екосистемою Microsoft, обраний стек пропонує більш сучасний та оптимізований підхід до веб-розробки.

Узагальнюючи, можна стверджувати, що TypeScript, React, Next.js, Prisma, PostgreSQL та NextAuth.js формують ефективну екосистему для створення сучасних, швидких та масштабованих веб-додатків, які відповідають актуальним вимогам щодо продуктивності, безпеки, доступності та зручності використання. Ця комбінація технологій забезпечує оптимальний баланс між швидкістю розробки, якістю коду та

користувацьким досвідом, що робить її ідеальним вибором для реалізації веб-проектів різної складності.

3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РЕГІОНАЛЬНОГО ЦЕНТРУ ЗАЙНЯТОСТІ

3.1 Структура ІС РЦЗ

3.1.1. Загальна інформація про структуру web-додатку

Веб-додаток ІС РЦЗ представляє собою сучасний інтерфейс з боковою навігацією, яку можна згорнути для збільшення робочого простору. Основні елементи структури:

- Початкова сторінка аутентифікації: Перше, що бачить користувач – це сторінка входу або реєстрації з можливістю авторизації через Google або електронну пошту (з використанням NextAuth.js).
- Бокова навігація: Після входу в систему користувач має доступ до бокової панелі навігації, яка містить всі розділи, доступні відповідно до його ролі. Панель можна згорнути для оптимізації робочого простору.
- Профіль користувача: Після першого входу користувач потрапляє на сторінку заповнення персональних даних. Повноцінний доступ до функціоналу системи відкривається тільки після внесення необхідної інформації про себе.
- Дешборд: Основна робоча область, що відображає релевантну інформацію в залежності від ролі користувача (статистика, вакансії, кандидати тощо).
- Робоча область: Центральна частина екрану, де відображається активний розділ (список вакансій, деталі профілю, форми редагування тощо).
- Елементи управління профілем: У верхній частині інтерфейсу розміщені елементи для швидкого доступу до налаштувань профілю, сповіщень та виходу з системи.

Структура додатку розроблена згідно з підходом "спочатку мобільні пристрої" (mobile-first), забезпечуючи адаптивне відображення на різних пристроях. Бокова навігація автоматично згортається на малих екранах і може бути розгорнута користувачем за потреби.

3.1.2 Навігація та ролі користувачів

Для зручної навігації по веб-додатку використовується бокова панель, що забезпечує швидке та зручне переміщення між розділами системи. Система підтримує різні ролі користувачів, кожна з яких має доступ до специфічного набору функцій та сторінок:

1. ADMIN (Адміністратор системи):
    - Дешборд з аналітикою системи.
    - Сторінка управління користувачами (перегляд, редагування ролей).
    - Список всіх вакансій з можливістю модерації.
    - Список всіх роботодавців (EMPLOYER).
    - Список всіх консультантів (CAREER_ADVISOR).
    - Налаштування системи.
2. CAREER_ADVISOR (Консультант центру зайнятості):
    - Дешборд з інформацією про активні вакансії та заявки.
    - Список всіх вакансій, створених роботодавцями.
    - Інтерфейс для перевірки та затвердження вакансій.
    - Календар консультацій з шукачами роботи.
    - Профілі шукачів роботи.
3. EMPLOYER (Роботодавець):
    - Профіль компанії з можливістю редагування інформації.
    - Сторінка створення нової вакансії.Список створених вакансій з їхнім статусом (на розгляді, активна, закрита).
    - Статистика переглядів та відгуків на вакансії.

- Список кандидатів, що відгукнулись на вакансії.

4. JOB_SEEKER (Шукач роботи):
    - Персональний дешборд з прогресом заповнення профілю.
    - Профіль з можливістю редагування особистої інформації, досвіду, навичок та освіти.
    - Сторінка пошуку вакансій з фільтрами та можливістю збереження.
    - Збережені вакансії.
    - Історія відгуків на вакансії.
    - Можливість перегляду профілів інших користувачів

Система також підтримує аутентифікацію через Google OAuth та електронну пошту з використанням NextAuth.js, що забезпечує безпечний та зручний процес реєстрації та входу в систему.

3.1.3 Наповнення (контент)

За додавання контенту відповідають адміністратори, які у подальшому можуть редагувати власний контент. Консультанти центру зайнятості (CAREER_ADVISOR) модерують вакансії, створені роботодавцями, перевіряючи їх на відповідність вимогам та політиці центру зайнятості.

3.1.4 Дизайн

Дизайн ІС РЦЗ виконаний у сучасному та мінімалістичному стилі. Використано компоненти з React, стилізовані для створення консистентного інтерфейсу користувача. Адаптивний дизайн забезпечує коректне відображення на різних пристроях, від мобільних телефонів до настільних комп'ютерів.

Шрифт підібраний таким чином, щоб користувачу було зручно та комфортно переглядати сторінки сайту. Елементи web-сайту розміщені

інтуїтивно зрозуміло, що дозволяє користувачам легко орієнтуватися в системі та ефективно використовувати всі функції.

Архітектура додатку базується на сучасному стеку технологій:

1. Frontend: Next.js з App Router, React компоненти, TypeScript для типобезпеки
2. Backend: Серверні компоненти та серверні дії Next.js, API Routes для роботи з даними
3. База даних: PostgreSQL для зберігання даних, Prisma як ORM для типобезпечного доступу
4. Аутентифікація: NextAuth.js з підтримкою Google OAuth та аутентифікації через email

Використання серверних компонентів Next.js дозволяє ефективно розділити логіку між сервером і клієнтом, зменшуючи об'єм JavaScript, що надсилається клієнту, і покращуючи продуктивність додатку.

3.2 Проектування та створення бази даних ІС РЦЗ

В результаті дослідження всіх аспектів роботи розроблюваної інформаційної системи розроблена наступна діаграма прецедентів, яка описує роботу програмного модулю «(рис. 3.1)».

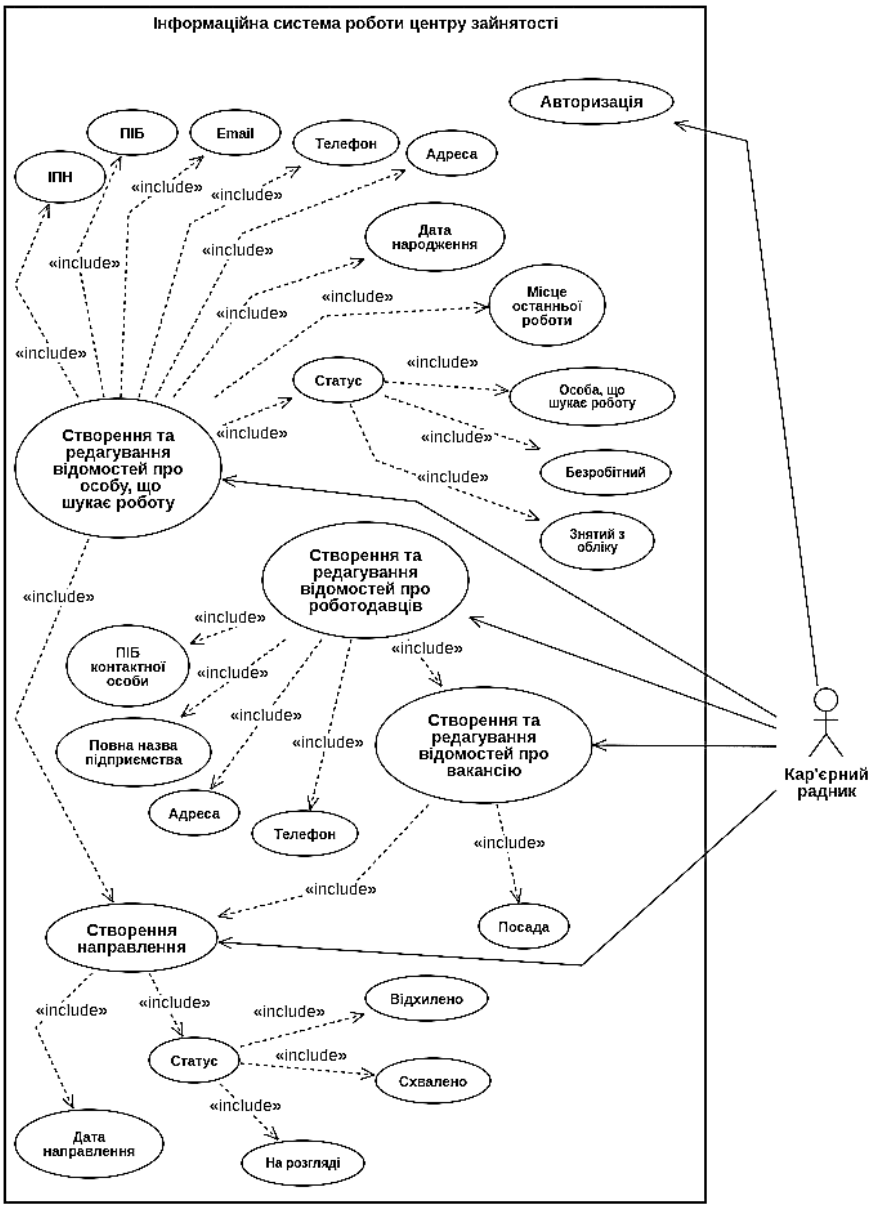

Рисунок. 3.1. Діаграма прецедентів роботи програмного модулю ІС центру зайнятості (Use Case).

На діаграмі прецедентів для подальшої розробки бази даних додатку виявлені наступні сутності, представлені на рисунках «(рис.3.2)», «(рис.3.3)», «(рис.3.4)», «(рис.3.5)», «(рис.3.6)» :

Рисунок. 3.2. Сутність Безробітний

Рисунок. 3.3. Сутність Роботодавець

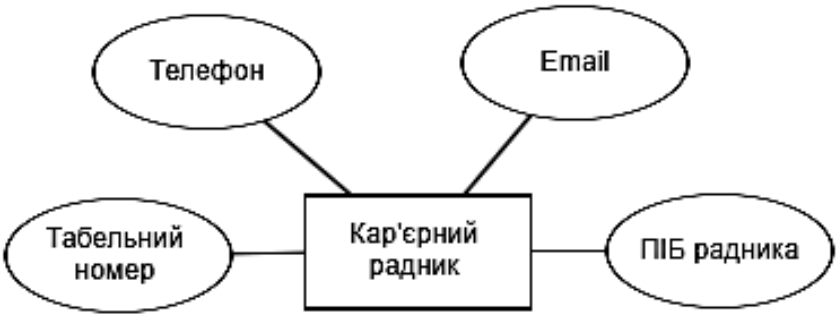

Рисунок. 3.4. Сутність Кар’єрний радник

Рисунок. 3.5. Сутність Вакансія

Рисунок. 3.6. - Сутність Направлення

Між даними сутностями встановлюємо зв'язки «(рис.3.7)».

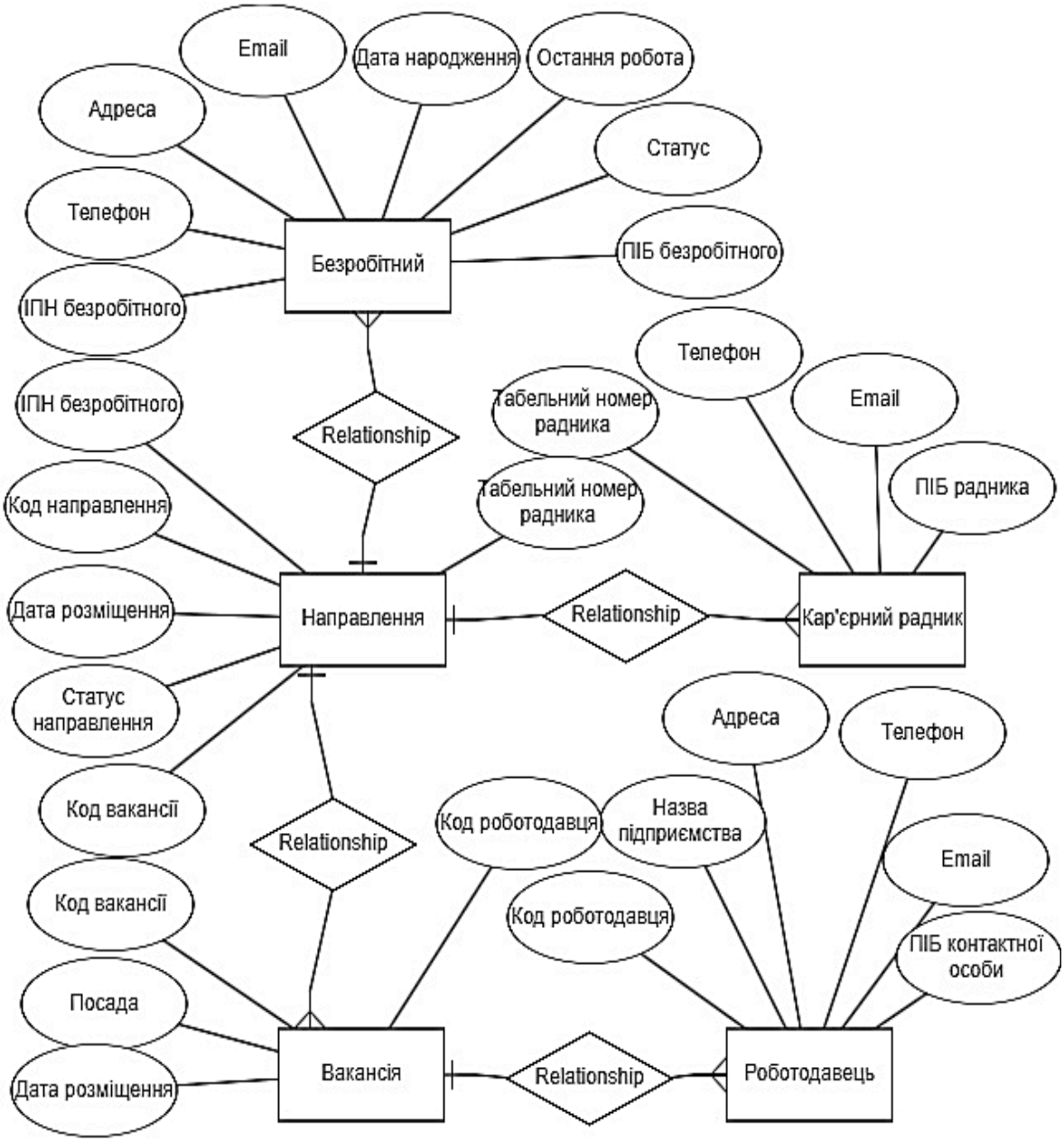

Рисунок. 3.7. Концептуальна схема предметної області.

Далі складаємо таблиці на підставі розробленої концептуальної моделі бази даних з використанням Prisma Schema.

Таблиця User

Таблиця відповідає за зберігання основної інформації про користувачів системи незалежно від їх ролі.

Привласнимо ім'я User й введемо наступні імена атрибутів:

1. id - String - дане поле є первинним ключем таблиці, що однозначно ідентифікує користувача, використовується UUID;
2. email - String - символьне поле, що містить електронну пошту користувача, має обмеження унікальності;
3. name - String - символьне поле, що містить ім'я користувача;
4. password - String? - символьне поле, що містить хешований пароль користувача (опціональне, оскільки користувач може авторизуватися через Google);
5. image - String? - символьне поле, що містить URL-адресу аватара користувача;
6. role - Enum - перелічувальне поле, що містить роль користувача в системі (ADMIN, CAREER_ADVISOR, EMPLOYER, JOB_SEEKER);
7. createdAt - DateTime - часове поле, що містить дату та час створення облікового запису;
8. updatedAt - DateTime - часове поле, що містить дату та час останнього оновлення профілю;
9. profile - Profile - зв'язок з таблицею Profile (один до одного);
10. jobs - Job[] - зв'язок з таблицею Job (один до багатьох) для користувачів з роллю EMPLOYER;
11. applications - Application[] - зв'язок з таблицею Application (один до багатьох) для користувачів з роллю JOB_SEEKER;
12. savedJobs - SavedJob[] - зв'язок з таблицею SavedJob (один до багатьох) для користувачів з роллю JOB_SEEKER;

13. approvedJobs - Job[] - зв'язок з таблицею Job (один до багатьох) для користувачів з роллю CAREER_ADVISOR.

Таблиця Profile

Таблиця зберігає розширену інформацію про користувача в залежності від його ролі.

Привласнимо ім'я Profile й введемо наступні імена атрибутів:

1. id - String - дане поле є первинним ключем таблиці;
2. userId - String - зовнішній ключ, що пов'язує профіль з користувачем (таблиця User);
3. bio - String? - символьне поле, що містить біографію або опис користувача;
4. phone - String? - символьне поле, що містить контактний телефон користувача;
5. address - String? - символьне поле, що містить адресу користувача;
6. dateOfBirth - DateTime? - часове поле, що містить дату народження користувача;
7. skills - String[] - масив рядків, що містить перелік навичок користувача;
8. resumeUrl - String? - символьне поле, що містить посилання на резюме (для JOB_SEEKER);
9. companyName - String? - символьне поле, що містить назву компанії (для EMPLOYER);
10. companyDescription - String? - символьне поле, що містить опис компанії (для EMPLOYER);
11. completionPercentage - Int - числове поле, що відображає відсоток заповнення профілю;
12. user - User - зв'язок з таблицею User (один до одного);
13. experiences - Experience[] - зв'язок з таблицею Experience (один до багатьох);

14. educations - Education[] - зв'язок з таблицею Education (один до багатьох).

Таблиця Experience

Таблиця містить інформацію про досвід роботи користувача.

Привласнимо ім'я Experience й введемо наступні імена атрибутів:

1. id - String - дане поле є первинним ключем таблиці;
2. profileId - String - зовнішній ключ, що пов'язує досвід з профілем користувача;
3. company - String - символьне поле, що містить назву компанії;
4. position - String - символьне поле, що містить посаду;
5. description - String? - символьне поле, що містить опис обов'язків;
6. startDate - DateTime - часове поле, що містить дату початку роботи;
7. endDate - DateTime? - часове поле, що містить дату закінчення роботи (опціональне для поточної роботи);
8. isCurrent - Boolean - логічне поле, що вказує, чи є це поточним місцем роботи;
9. profile - Profile - зв'язок з таблицею Profile (багато до одного).

Таблиця Education

Таблиця містить інформацію про освіту користувача.

Привласнимо ім'я Education й введемо наступні імена атрибутів:

1. id - String - дане поле є первинним ключем таблиці;
2. profileId - String - зовнішній ключ, що пов'язує освіту з профілем користувача;
3. institution - String - символьне поле, що містить назву навчального закладу;
4. degree - String - символьне поле, що містить отриманий ступінь;
5. field - String - символьне поле, що містить область/спеціальність навчання;
6. startDate - DateTime - часове поле, що містить дату початку навчання;

7. endDate - DateTime? - часове поле, що містить дату закінчення навчання (опціональне для поточного навчання);
8. isCurrent - Boolean - логічне поле, що вказує, чи є це поточним місцем навчання;
9. profile - Profile - зв'язок з таблицею Profile (багато до одного).

Таблиця Job

Таблиця містить інформацію про вакансії, створені роботодавцями.

Привласнимо ім'я Job й введемо наступні імена атрибутів:

1. id - String - дане поле є первинним ключем таблиці;
2. title - String - символьне поле, що містить назву вакансії;
3. description - String - символьне поле, що містить детальний опис вакансії;
4. requirements - String - символьне поле, що містить вимоги до кандидата;
5. salary - Int? - числове поле, що містить інформацію про зарплату (опціональне);
6. location - String - символьне поле, що містить місце роботи;
7. type - Enum - перелічувальне поле, що вказує тип зайнятості (FULL_TIME, PART_TIME, CONTRACT, INTERNSHIP);
8. employerId - String - зовнішній ключ, що пов'язує вакансію з роботодавцем (User з роллю EMPLOYER);
9. status - Enum - перелічувальне поле, що вказує статус вакансії (PENDING, APPROVED, CLOSED);
10. approvedById - String? - зовнішній ключ, що пов'язує вакансію з консультантом, який її затвердив;
11. createdAt - DateTime - часове поле, що містить дату та час створення вакансії;
12. updatedAt - DateTime - часове поле, що містить дату та час останнього оновлення вакансії;
13. employer - User - зв'язок з таблицею User (багато до одного);

14. approvedBy - User? - зв'язок з таблицею User (багато до одного, опціональний);
15. applications - Application[] - зв'язок з таблицею Application (один до багатьох);
16. savedBy - SavedJob[] - зв'язок з таблицею SavedJob (один до багатьох).

Таблиця Application

Таблиця містить інформацію про відгуки на вакансії (направлення).

Привласнимо ім'я Application й введемо наступні імена атрибутів:

1. id - String - дане поле є первинним ключем таблиці;
2. jobId - String - зовнішній ключ, що пов'язує відгук з вакансією;
3. jobSeekerId - String - зовнішній ключ, що пов'язує відгук з шукачем роботи (User з роллю JOB_SEEKER);
4. status - Enum - перелічувальне поле, що вказує статус відгуку (PENDING, REVIEWED, ACCEPTED, REJECTED);
5. coverLetter - String? - символьне поле, що містить супровідний лист;
6. createdAt - DateTime - часове поле, що містить дату та час створення відгуку;
7. updatedAt - DateTime - часове поле, що містить дату та час останнього оновлення відгуку;
8. job - Job - зв'язок з таблицею Job (багато до одного);
9. jobSeeker - User - зв'язок з таблицею User (багато до одного).
10. Таблиця SavedJob
11. Таблиця містить інформацію про збережені вакансії шукачами роботи.
12. Привласнимо ім'я SavedJob й введемо наступні імена атрибутів:
13. id - String - дане поле є первинним ключем таблиці;
14. jobId - String - зовнішній ключ, що пов'язує запис зі збереженою вакансією;
15. userId - String - зовнішній ключ, що пов'язує запис з користувачем, який зберіг вакансію;

16. createdAt - DateTime - часове поле, що містить дату та час збереження вакансії;
17. job - Job - зв'язок з таблицею Job (багато до одного);
18. user - User - зв'язок з таблицею User (багато до одного).

Таблиця Account

Таблиця містить інформацію про облікові записи авторизації користувачів через зовнішні провайдери (наприклад, Google).

Привласнимо ім'я Account й введемо наступні імена атрибутів:

1. id - String - дане поле є первинним ключем таблиці;
2. userId - String - зовнішній ключ, що пов'язує запис з користувачем;
3. type - String - символьне поле, що вказує тип авторизації (наприклад, "oauth");
4. provider - String - символьне поле, що вказує провайдера авторизації (наприклад, "google");
5. providerAccountId - String - символьне поле, що містить ідентифікатор користувача в системі провайдера;
6. refresh_token - String? - символьне поле, що містить токен оновлення;
7. access_token - String? - символьне поле, що містить токен доступу;
8. expires_at - Int? - числове поле, що вказує час закінчення дії токена;
9. token_type - String? - символьне поле, що вказує тип токена;
10. scope - String? - символьне поле, що містить доступні області дії токена;
11. id_token - String? - символьне поле, що містить ідентифікаційний токен;
12. session_state - String? - символьне поле, що містить стан сесії;
13. user - User - зв'язок з таблицею User (багато до одного).

Таблиця Session

Таблиця містить інформацію про активні сесії користувачів.

Привласнимо ім'я Session й введемо наступні імена атрибутів:

1. id - String - дане поле є первинним ключем таблиці;

2. sessionToken - String - символьне поле, що містить унікальний токен сесії;
3. userId - String - зовнішній ключ, що пов'язує сесію з користувачем;
4. expires - DateTime - часове поле, що вказує час закінчення сесії;
5. user - User - зв'язок з таблицею User (багато до одного).

Таблиця VerificationToken

Таблиця містить інформацію про токени верифікації для підтвердження електронної пошти та скидання пароля.

Привласнимо ім'я VerificationToken й введемо наступні імена атрибутів:

1. identifier - String - символьне поле, що містить ідентифікатор користувача;
2. token - String - символьне поле, що містить токен верифікації;
3. expires - DateTime - часове поле, що вказує час закінчення дії токена.

Дана схема бази даних на основі Prisma Schema забезпечує повноцінну підтримку всіх функцій системи, включаючи управління користувачами, вакансіями, направленнями та аутентифікацією через різні провайдери. Використання Prisma ORM дозволяє ефективно працювати з даними, забезпечуючи типобезпеку та оптимізовані запити до бази даних.

Після розробки схеми Prisma, необхідно підняти сервер бази даних та виконати міграції для створення структури бази даних.

Запуск PostgreSQL сервера через Docker:

1. Створив файл docker-compose.yml з наступним вмістом:

```
version: '3'
services:
  postgres:
    image: postgres:14
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: job_center
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data

volumes:
  postgres_data:
```

2. Запустив PostgreSQL сервер з командою “docker-compose up -d”
3. Згенерував міграцію на основі схеми Prisma командою “npx prisma migrate dev --name init”
4. Згенерував Prisma Client для типобезпечного доступу до бази даних “npx prisma generate”
5. для перегляду бази даних використовую Beekeeper Studio

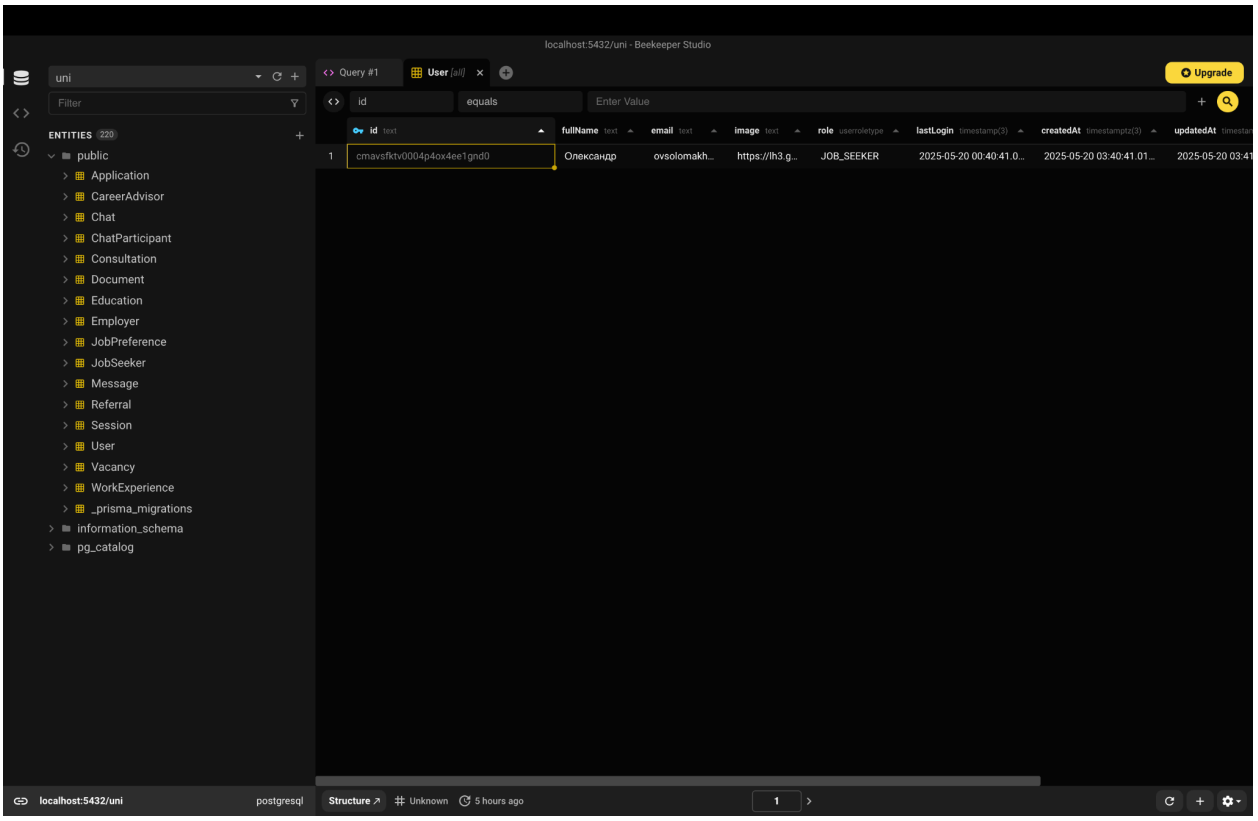

Рисунок. 3.8. Перегляд бази даних через Beekeeper Studio

3.3 Проектування серверної частини ІС РЦЗ

3.3.1 Архітектура API маршрутів Next.js

Серверна частина інформаційної системи регіонального центру зайнятості побудована на основі Next.js App Router, який забезпечує повну інтеграцію серверних компонентів, API маршрутів та серверних дій для роботи з даними.

Таблиця 3.1.

Основні функції веб-API.

API маршрут	Опис	Метод запиту	Тип відповіді
/api/users	Отримання списку користувачів	GET	Масив користувачів
/api/users/[id]	Отримання даних користувача	GET	Об'єкт користувача
/api/users	Додавання нового користувача	POST	Створений об'єкт користувача
/api/users/[id]	Оновлення даних користувача	PATCH	Оновлений об'єкт користувача
/api/users/[id]	Видалення користувача	DELETE	Статус успішного видалення
/api/jobs	Отримання списку вакансій	GET	Масив вакансій
/api/jobs/[id]	Отримання даних вакансії	GET	Об'єкт вакансії
/api/jobs	Створення нової вакансії	POST	Створений об'єкт вакансії
/api/jobs/[id]	Оновлення вакансії	PATCH	Оновлений об'єкт вакансії
/api/jobs/[id]	Видалення вакансії	DELETE	Статус успішного видалення
/api/applications	Робота з відгуками на вакансії	GET, POST, PATCH, DELETE	Відповідний об'єкт або статус

3.3.2 Структура серверної частини

На наступному рисунку показано структуру серверної частини додатку.

```
src/
├── app/
│   ├── api/                  # API маршрути
│   │   ├── auth/             # Аутентифікація через NextAuth.js
│   │   │   └── [...nextauth]/   # Конфігурація NextAuth
│   │   ├── users/            # API для роботи з користувачами
│   │   │   └── [id]/         # Операції з конкретним користувачем
│   │   ├── jobs/             # API для роботи з вакансіями
│   │   │   └── [id]/         # Операції з конкретною вакансією
│   │   └── applications/       # API для роботи з відгуками
│   │       └── [id]/         # Операції з конкретним відгуком
│   └── (private)/            # Захищені маршрути (потребують авторизації)
```

```
│       ├── profile/          # Сторінка профілю користувача
│       ├── dashboard/        # Дашборд у відповідності до ролі
│       ├── jobs/             # Сторінки для роботи з вакансіями
│       └── admin/            # Адміністративна панель
├── lib/                      # Утиліти та хелпери
│   ├── prisma.ts             # Клієнт Prisma для роботи з БД
│   └── auth.ts               # Утиліти для авторизації
└── utils/                    # Додаткові службові функції
    └── routes.ts             # Визначення маршрутів додатку
```

3.3.3 Реалізація доступу до даних через Prisma

Доступ до бази даних здійснюється через Prisma ORM, яка забезпечує типобезпечну взаємодію з PostgreSQL. Для взаємодії з базою даних створено єдиний екземпляр Prisma Client, який використовується у серверних компонентах та API маршрутах:

```ts
// src/lib/prisma.ts
import { PrismaClient } from '@prisma/client'

const globalForPrisma = globalThis as unknown as { prisma: PrismaClient }

export const prisma = globalForPrisma.prisma || new PrismaClient()

if (process.env.NODE_ENV !== 'production') globalForPrisma.prisma = prisma
```

Рисунок. 3.9 Prisma Client

```
// src/app/api/jobs/route.ts
import { NextResponse } from 'next/server'
import { prisma } from '@/lib/prisma'
import { getServerSession } from 'next-auth'
import { authOptions } from '../auth/[...nextauth]/route'

export async function GET(request: Request) {
  const { searchParams } = new URL(request.url)
  const page = parseInt(searchParams.get('page') || '1')
  const limit = parseInt(searchParams.get('limit') || '10')
  const status = searchParams.get('status')
  const search = searchParams.get('search')

  const where = {}
  if (status) where.status = status
  if (search) where.title = { contains: search, mode: 'insensitive' }

  const jobs = await prisma.job.findMany({
    where,
    include: { employer: { select: { name: true, image: true } } },
    skip: (page - 1) * limit,
    take: limit,
    orderBy: { createdAt: 'desc' }
  })

  const total = await prisma.job.count({ where })

  return NextResponse.json({
    items: jobs,
    total,
    page,
    pages: Math.ceil(total / limit)
  })
}
```

Рисунок. 3.10 Приклад реалізації API маршруту для отримання списку вакансій

```
// src/app/api/jobs/route.ts
export async function POST(request: Request) {
  const session = await getServerSession(authOptions)

  if (!session || session.user.role !== 'EMPLOYER') {
    return NextResponse.json({ error: 'Unauthorized' }, { status: 401 })
  }

  const data = await request.json()

  const job = await prisma.job.create({
    data: {
      title: data.title,
      description: data.description,
      requirements: data.requirements,
      salary: data.salary,
      location: data.location,
      type: data.type,
      status: 'PENDING',
      employer: { connect: { id: session.user.id } }
    }
  })

  return NextResponse.json(job, { status: 201 })
}
```

Рисунок. 3.10 Створення нової вакансії

3.4 Створення клієнтської частини розробленого додатку інформаційної системи регіонального центру зайнятості

3.4.1 Реалізація функціональності для різних ролей користувачів

Клієнтська частина реалізує різний інтерфейс та функціональність залежно від ролі користувача:

ADMIN

1. Дашборд з аналітикою системи
2. Управління користувачами (зміна ролей, блокування)
3. Моніторинг та модерація вакансій
4. Перегляд статистики використання системи

CAREER_ADVISOR

1. Дашборд з останніми заявками на розгляд вакансій
2. Список вакансій для модерації
3. Інтерфейс перевірки та затвердження вакансій
4. Управління направленнями шукачів роботи

EMPLOYER

1. Дашборд з інформацією про статуси вакансій
2. Сторінка профілю компанії з можливістю редагування
3. Форма створення та редагування вакансій
4. Перегляд відгуків на вакансії від шукачів

JOB_SEEKER

1. Дашборд з прогресом заповнення профілю
2. Сторінка профілю з формами для редагування
3. Пошук вакансій з можливістю фільтрації та сортування
4. Перегляд та управління відгуками на вакансії

3.4.2 Взаємодія з API та авторизація

Для взаємодії з API використовуються серверні дії (Server Actions) Next.js та хуки для роботи з даними на клієнтській стороні. Авторизація реалізована через NextAuth.js з різними стратегіями входу:

```
// src/hooks/useJobs.ts
'use client'

import { useState, useEffect } from 'react'
import { useSession } from 'next-auth/react'

export function useJobs(initialFilters = {}) {
  const [jobs, setJobs] = useState([])
  const [loading, setLoading] = useState(true)
  const [filters, setFilters] = useState(initialFilters)
  const { data: session } = useSession()

  useEffect(() => {
    const fetchJobs = async () => {
      setLoading(true)
      try {
        const queryParams = new URLSearchParams()
        Object.entries(filters).forEach(([key, value]) => {
          if (value) queryParams.append(key, value.toString())
        })

        const response = await fetch(`/api/jobs?${queryParams}`)
        const data = await response.json()
        setJobs(data.items)
      } catch (error) {
        console.error('Failed to fetch jobs:', error)
      } finally {
        setLoading(false)
      }
    }

    fetchJobs()
  }, [filters])

  return { jobs, loading, setFilters }
}
```

Рисунок. 3.11 Приклад взаємодії з API

Загальна архітектура клієнтської частини забезпечує оптимальний користувацький досвід завдяки використанню серверних компонентів для швидкого початкового рендерингу та клієнтських компонентів для інтерактивності. Використання Prisma ORM забезпечує типобезпечну роботу з даними, а NextAuth.js гарантує надійну аутентифікацію та авторизацію користувачів різних ролей.

3.5 Запуск та тестування розробленого програмного продукту

3.5.1 Вимоги до функціонування web-додатку

Для забезпечення стабільної роботи web-додатку вимагається web-браузер Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

Рекомендовані вимоги до конфігурації персонального комп'ютеру:

- тип процесора Intel Pentium 1600 MHz або краще;
- обсяг оперативного запам'ятовуючого пристрою 512 Мб і більше;
- обсяг вільного місця на жорсткому диску 475 Мб і більше;
- операційна систем Microsoft Windows 10, 11.

3.5.2 Чек-лист функціональних вимог

Авторизація користувача в залежності від статусу:

- авторизація в якості адміністратора веб-застосунку;
- авторизація в якості кар’єрного радника;
- авторизація в якості особи, що шукає роботу;
- авторизація в якості роботодавця.

Постановка на облік осіб, що шукають роботу:

- заповнення анкетних даних безробітного;
- реєстрація безробітних онлайн;
- призначення кар’єрного радника.

Реєстрація вакансій:

- розміщення вакансій роботодавцями онлайн;
- заповнення інформації про вакансію;
- підтвердження вакансії кар’єрним радником.

Надання направлення на вакансії:

- Направлення безробітного на вакансію;
- Відстеження результатів направлення.

Ведення бази даних:

- зберігання інформації про безробітних: анкетні дані, освіта, дата останньої роботи, резюме;

- зберігання інформації про вакансії: посада, сфера діяльності, вимоги до кандидатів, рівень зарплати, тип зайнятості, контактні дані роботодавця;

- зберігання інформації про роботодавців: реєстраційні дані, контактні дані, сфера діяльності, кількість вакансій;

- зберігання інформації про надані направлення: дата направлення, вакансія, безробітний, результат направлення;

- зберігання інформації про кар'єрних радників: анкетні дані, контактні дані, спеціалізація.

Пошук вакансій:

- можливість пошуку вакансій за різними критеріями: назва вакансії, сфера діяльності, місце роботи, рівень зарплати, тип зайнятості (постійне, тимчасове, стажування);

3.5.3 Заплановані тест-кейси

Тест-кейси для авторизації користувачів

1. Авторизація адміністратора

Тест-кейс 1.1: Успішна авторизація з правильним логіном і паролем.

Передумови: Адміністратор зареєстрований в системі.

Слід зазначити, що Державна служба зайнятості, орієнтуючись на сучасні підходи цифровізації державного управління, впроваджує електронні сервіси як одну з провідних форм надання соціальних послуг населенню та роботодавцям через цифрові засоби, тому у 2018 році в межах цієї стратегії було запущено інноваційну платформу - web-сайт «Єдине соціальне середовище зайнятості», який:

- уможливив самостійну реєстрацію безробітних засобом інформаційних технологій;

- генерує базу поданих вакансій та резюме по всій території України;

- надає зареєстрованим клієнтам особисті кабінети для самообслуговування;

- дозволяє зареєстрованим роботодавцям подавати звітність онлайн;

- пропонує доступ до актуальної аналітики та статистики ринку праці;

- передбачає можливість проходження профорієнтаційного тестування та функціонує на комп'ютерах та на мобільних пристроях із системами Android та Apple.

Передумови: Кар'єрний радник зареєстрований в системі.

Кроки: Ввести змінений логін та змінений пароль, натиснути кнопку "Увійти".

Очікуваний результат: Користувач перенаправляється на панель кар'єрного радника з усіма доступними функціями.

3. Авторизація шукача роботи

Тест-кейс 3.1: Успішна авторизація з правильними даними.

Передумови: Шукач роботи зареєстрований в системі.

Кроки: Ввести правильний логін та пароль, натиснути кнопку "Увійти".

Очікуваний результат: Користувач перенаправляється до особистого кабінету шукача роботи з усіма доступними функціями. Тест-кейс 3.2: Неуспешна авторизація з невірно введеними даними. Передумови: Шукач роботи зареєстрований в системі.

Кроки: Ввести неіснуючий логін та будь-який пароль, натиснути кнопку "Увійти".

Очікуваний результат: З'являється повідомлення про помилку "Користувач не знайдений".

4. Авторизація роботодавця

Тест-кейс 4.1: Успішна авторизація з правильними даними.

Передумови: Роботодавець зареєстрований в системі.

Кроки: Ввести правильний логін та пароль, натиснути кнопку "Увійти".

Очікуваний результат: Користувач перенаправляється до особистого кабінету роботодавця роботи з усіма доступними функціями. Тест-кейс 4.2: Неуспешна авторизація з невірно введеними даними. Передумови: Роботодавець зареєстрований в системі.

Кроки: Ввести неіснуючий логін та будь-який пароль, натиснути кнопку "Увійти".

Очікуваний результат: З'являється повідомлення про помилку "Користувач не знайдений".

Тест-кейси для заповнення анкетних даних безробітного

Тест-кейс 5.1: Заповнення всіх обов'язкових полів анкетних даних безробітного.

Передумови: Користувач перейшов на сторінку заповнення анкети шукача роботи.

Кроки: Користувач послідовно заповнює всі обов'язкові поля анкети (ПІБ, контактна інформація, освіта, досвід роботи тощо) коректними даними.

Очікуваний результат: Анкета зберігається, користувач перенаправляється на сторінку з повідомленням про успішне збереження.

Тест-кейс 5. 2: Заповнення анкетних даних безробітного з помилками

Передумови: Користувач перейшов на сторінку заповнення анкети.

Кроки: Користувач залишає деякі обов'язкові поля порожніми або вводить некоректні дані (наприклад, невірний формат дати народження).

Очікуваний результат: Система видає повідомлення про помилку, що вказані не всі дані або дані введені неправильно.

Тест-кейс 5.3: Завантаження резюме

Передумови: Користувач заповнив основні дані анкети.

Кроки: Користувач завантажує файл резюме у підтримуваному форматі (doc, docx, pdf).

Очікуваний результат: Файл успішно завантажується і прикріплюється до анкети.

Тест-кейс 5.4: Успішна реєстрація з унікальним email

Передумови: Користувач перейшов на сторінку реєстрації.

Кроки: Користувач вводить унікальну електронну пошту, пароль і повторює пароль, натискає кнопку "Зареєструватися".

Очікуваний результат: Користувач отримує повідомлення про успішну реєстрацію і переходить на сторінку авторизації або активації акаунту.

Тест-кейс 5.5: Спроба реєстрації з існуючим email

Передумови: Користувач з таким email вже зареєстрований

Кроки: Користувач намагається зареєструватися з існуючим email.

Очікуваний результат: Система видає повідомлення про те, що користувач з таким email вже існує.

Тест-кейс 5.6: Автоматичне призначення радника

Передумови: Налаштовано автоматичне призначення радника за певними критеріями.

Кроки: Користувач успішно реєструється.

Очікуваний результат: Систем автоматично призначає користувачу кар'єрного радника відповідно до налаштувань.

6. Тест-кейси для реєстрації вакансій

Тест-кейс 6.1: Успішне створення вакансії з заповненням всіх обов'язкових полів

Передумови: Роботодавець авторизований в системі.

Кроки: Роботодавець заповнює всі обов'язкові поля форми для створення вакансії (назва вакансії, опис, вимоги, зарплата, контактна інформація тощо) коректними даними.

Очікуваний результат: Вакансія успішно зберігається в системі, роботодавець бачить повідомлення про успішне створення.

Тест-кейс 6.2: Спроба створення вакансії з незаповненими обов'язковими полями

Передумови: Роботодавець авторизований в системі.

Кроки: Роботодавець залишає деякі обов'язкові поля порожніми і натискає кнопку "Створити".

Очікуваний результат: Система видає повідомлення про помилку, що не всі поля заповнені.

Тест-кейс 6.3: Успішна перевірка та підтвердження вакансії

Передумови: Кар'єрний радник авторизований в системі, вакансія створена роботодавцем.

Кроки: Кар'єрний радник переглядає інформацію про вакансію, вносить необхідні корективи (якщо потрібно) і підтверджує вакансію.

Очікуваний результат: Статус вакансії змінюється на "Підтверджена", вакансія стає доступною для шукачів роботи.

Тест-кейс 6.4: Відхилення вакансії з вказанням причини

Передумови: Кар'єрний радник авторизований в системі, вакансія створена роботодавцем.

Кроки: Кар'єрний радник відхиляє вакансію, вказавши причину відхилення.

Очікуваний результат: Статус вакансії змінюється на "Відхилена", роботодавець отримує повідомлення про причину відхилення.

Тест-кейс 6.5: Пошук підтвердженої вакансії шукачем роботи

Передумови: Шукач роботи авторизований в системі, вакансія підтверджена кар'єрним радником.

Кроки: Шукач роботи використовує функціонал пошуку вакансій і знаходить потрібну вакансію.

Очікуваний результат: Вакансія відображається в результатах пошуку.

7. Тест-кейси для надання направлення на вакансію.

Тест-кейс 7.1: Успішне створення направлення

Передумови: Існує активна вакансія і зареєстрований безробітний.

Кроки: Кар'єрний радник вибирає вакансію та безробітного, створює направлення.

Очікуваний результат: Система створює запис про направлення, в профілі безробітного та вакансії відображається інформація про направлення.

Тест-кейс 7.2: Спроба створення направлення на вже закриту вакансію

Передумови: Існує вакансія зі статусом "Закрито".

Кроки: Кар'єрний радник намагається створити направлення на цю вакансію.

Очікуваний результат: Система видає повідомлення про те, що вакансія недоступна для направлення.

Тест-кейс 7.3: Зміна статусу направлення кар'єрним радником Передумови: Існує активне направлення.

Кроки: Кар'єрний радник змінює статус направлення (наприклад, "Пройшов співбесіду", "Відмовився від вакансії").

Очікуваний результат: Статус направлення оновлюється в системі, роботодавець і безробітний отримують відповідні повідомлення (якщо передбачено).

3.5.4 Реалізовані тест-кейси

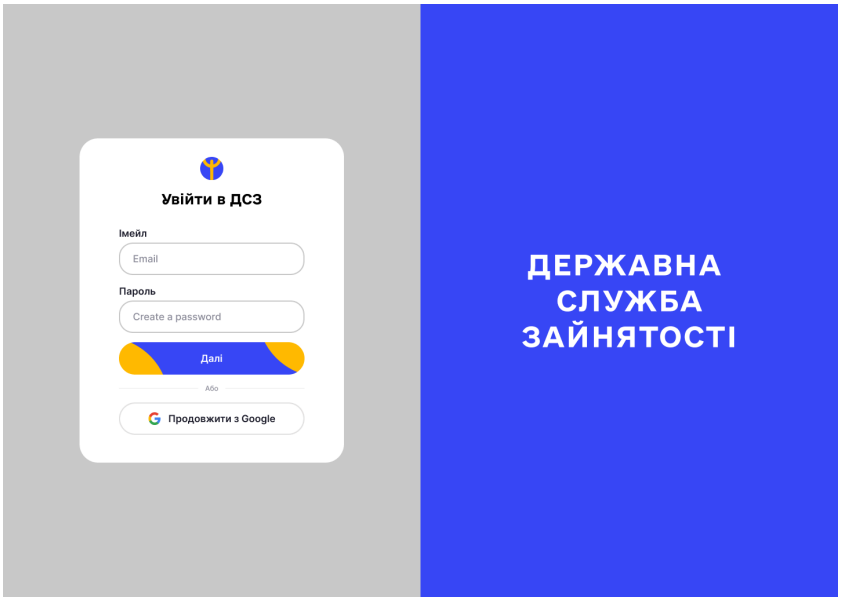

Рисунок. 3.22 Скріншот запуску додатку з появою вікна авторизації

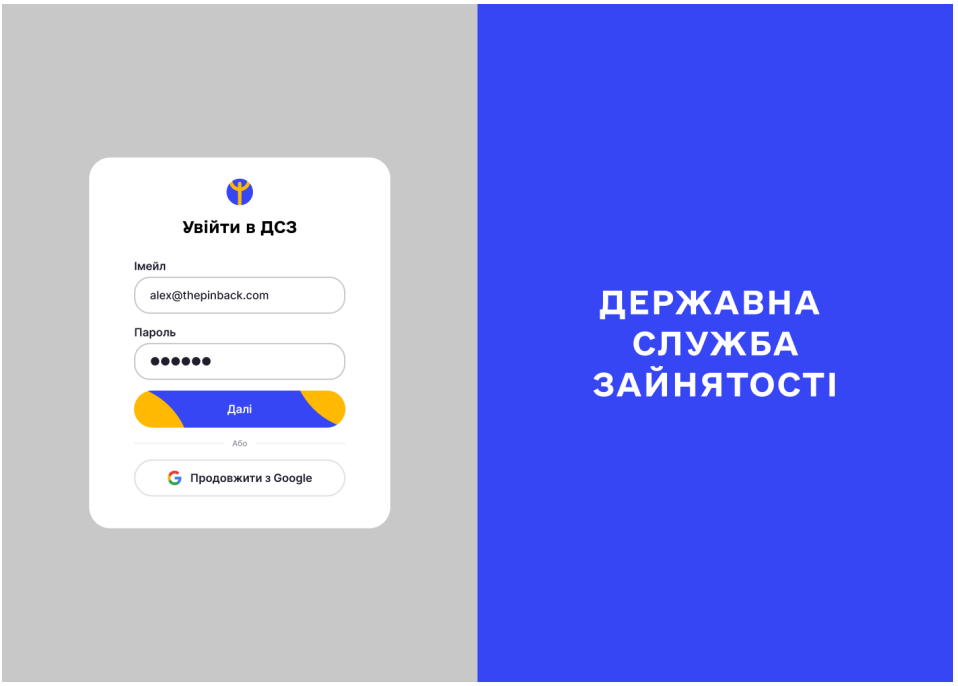

Рисунок. 3.23 Скріншот введення паролю та логіну

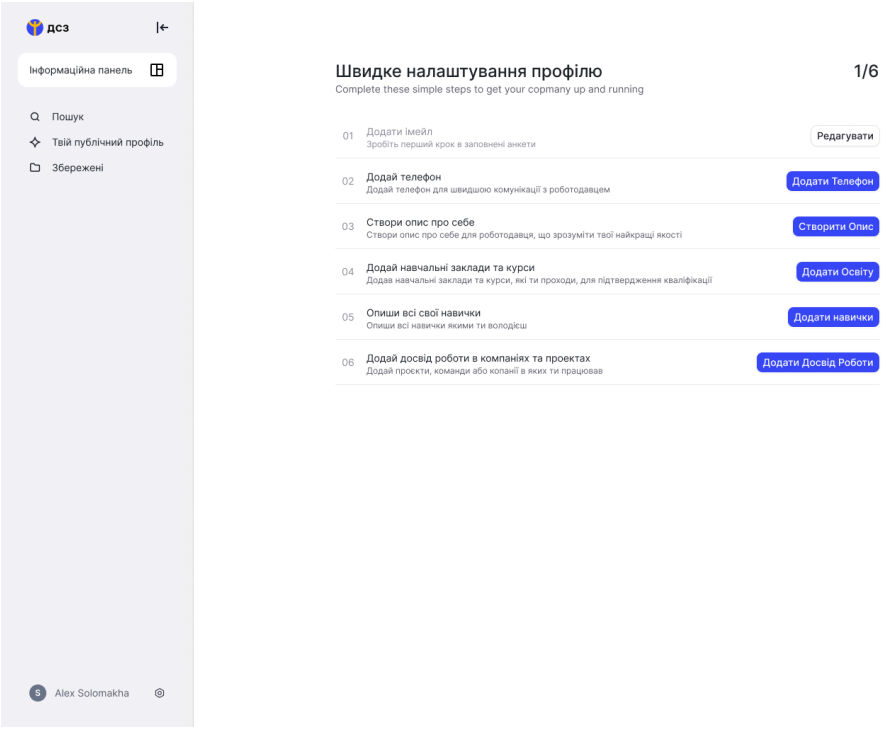

Рисунок. 3.24 Скріншот входу до головної сторінки

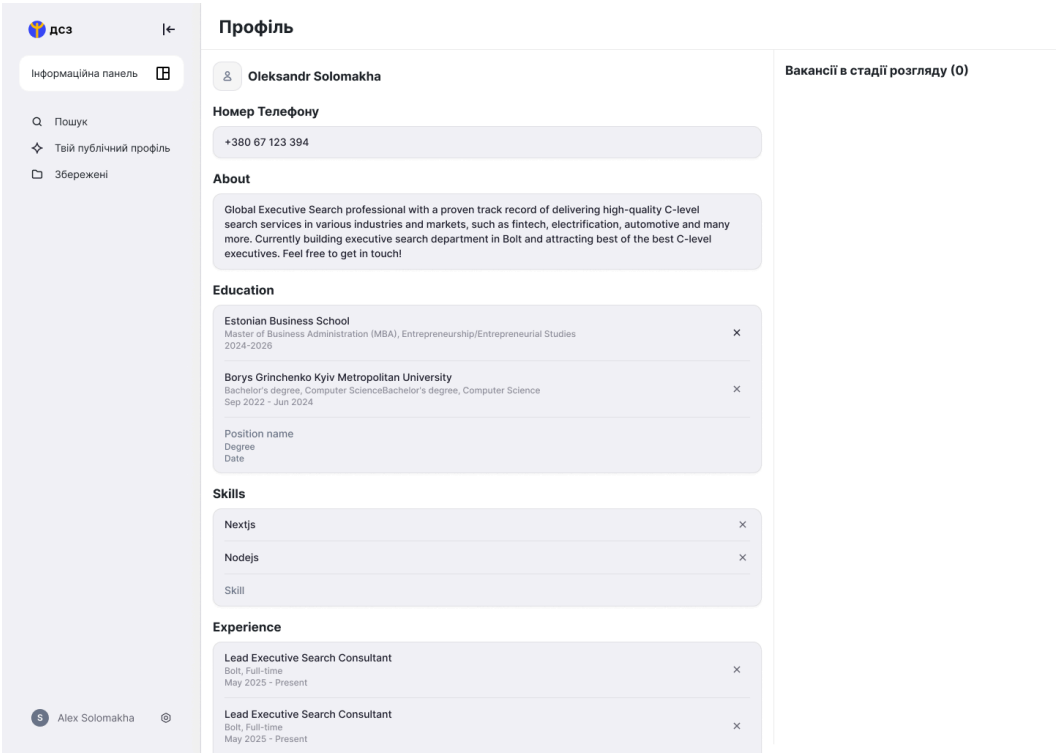

Рисунок. 3.25 Скріншот профілю шукача роботи

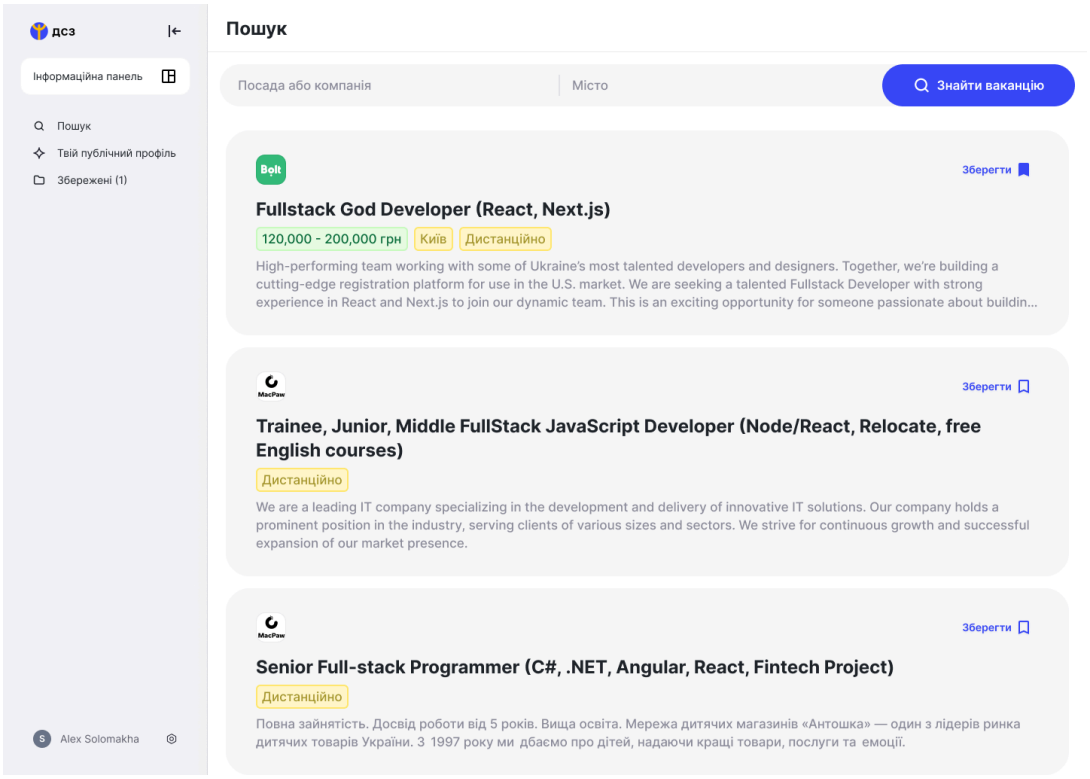

Рисунок. 3.26. Скріншот пошуку вакансій

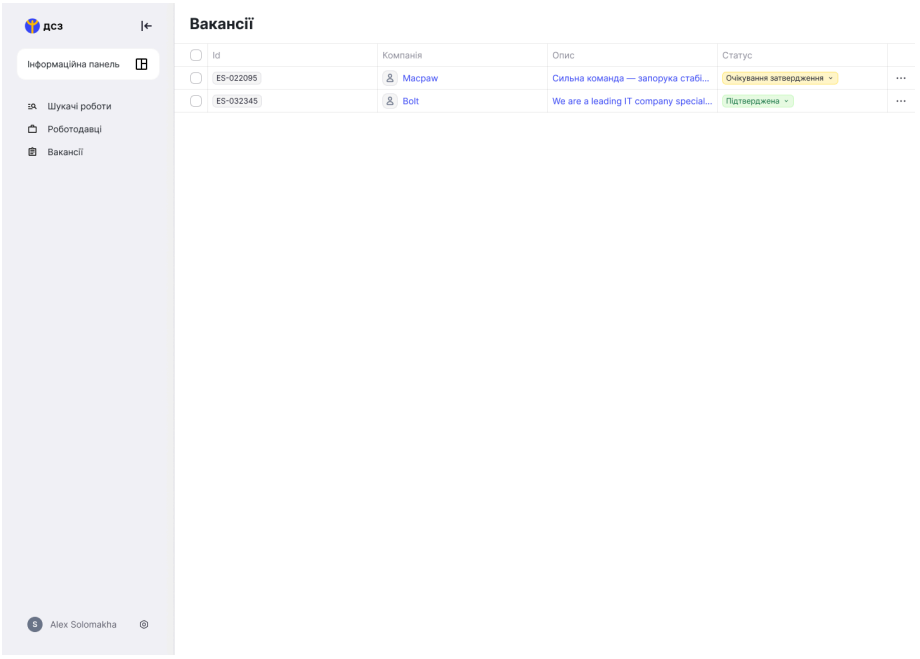

Рисунок. 3.27. Скріншот переходу до сторінки про вакансії

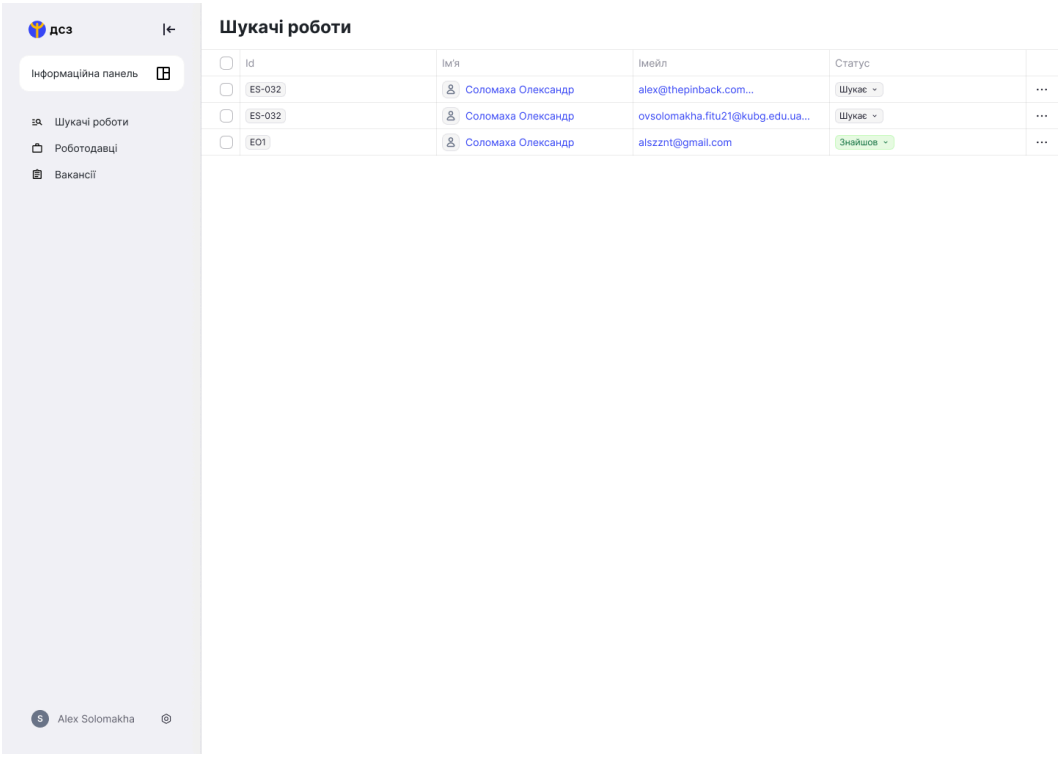

Рисунок. 3.28. Скріншот адміністратора з стиском шукачів роботи.

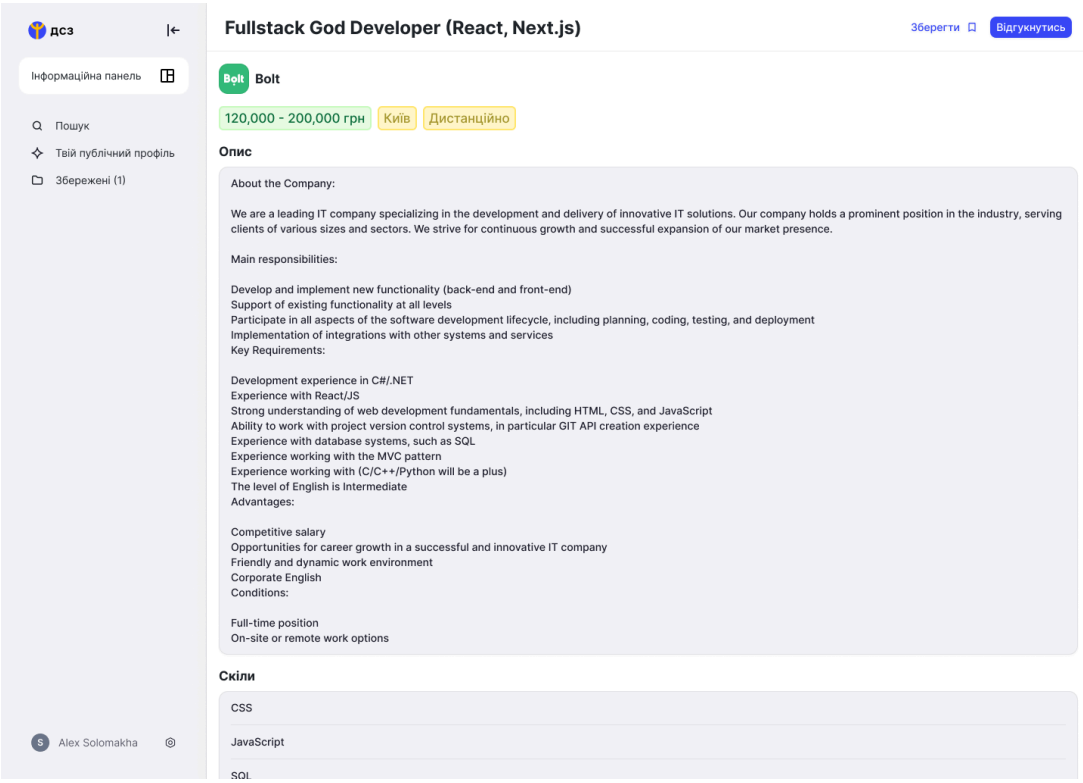

Рисунок. 3.29. Скріншот сторінки вакансії

Висновки до розділу 3

В ході розробки додатку додано наступні функціональні можливості:

- реєстрація та авторизація користувачів;
- ведення обліку безробітних;
- додавання, редагування та видалення записів.

Додаток має наступні технічні характеристики:

- повний стек: Next.js з App Router;
- клієнтська частина: React з TypeScript;
- серверна частина: API маршрути та серверні компоненти Next.js;
- система управління базами даних: PostgreSQL;
- фреймворк ORM: Prisma;
- аутентифікація: NextAuth.js з підтримкою Google OAuth та email.

Програма працює за певною інструкцією:

- введення логіну та паролю;

- вибір розділу, з яким треба працювати: "Особи", "Роботодавці", "Вакансії", "Направлення";

- використання кнопок "Додати", "Редагувати", "Видалити" для керування записами.

У разі впровадження, пропонований додаток автоматизує процес реєстрації безробітних у РЦЗ, зробить процес пошуку роботи простішим, адже простота використання та необхідні функції є основою ефективного ведення обліку пошукачів роботи та співпраці з ними.

ВИСНОВКИ

У бакалаврському дослідженні здійснено аналіз сучасних тенденцій розвитку процесів працевлаштування та найму, уточнено п’ять трендів рекрутингу 2024 та 2025 році.

Проаналізовано та уточнено завдання Державних служби зайнятості у сфері зайнятості населення загалом, а саме структуру, особливості функціонування типового РЦЗ, його функції та механізми роботи.

На основі аналізу наукових досліджень щодо ефективності діяльності служби, дійшли висновку, що автоматизація та діджиталізація роботи РЦЗ відповідають вимогам цифрового суспільства та сприятимуть ефективному використанню людського ресурсу.

Проведено аналіз існуючих аналогів програмного забезпечення, що функціонують у сфері працевлаштування, такі як державна платформа “Єдине соціальне середовище зайнятості”, Coloris HRM, платформи Rippling, складено порівняльну таблицю та визначено недоліки кожної.

Досліджено технічні виклики, а саме зроблено порівняльний аналіз інструментарію: мову програмування TypeScript, фреймворк Next.js, React для побудови інтерфейсу, Prisma як ORM, PostgreSQL як систему управління базами даних, та середовище розробки Visual Studio Code. Обрано метод розробки та реалізації програмного додатку інформаційної системи регіонального центру зайнятості.

На основі проведеного аналізу предметної області засобами TypeScript була побудована база даних з використанням Prisma Schema, яка містить інформацію про безробітних, роботодавців, вакансії та направлення. Також побудовані моделі даних Prisma, діаграма активності системи та макети основних інтерфейсних компонентів програми.

Був розроблений інтерфейс користувача системи з використанням React компонентів, який складається з таких основних сторінок: автентифікація, дешборд, профіль користувача, вакансії, роботодавці та направлення. Окремо

розроблені додаткові компоненти для різних ролей користувачів (адміністратор, кар'єрний радник, роботодавець, шукач роботи).

Система використовує наступні алгоритми:

- алгоритм автентифікації: використовує NextAuth для безпечної перевірки доступу користувачів;
- алгоритм додавання даних: додає нові дані до бази даних через Prisma ORM;
- алгоритм оновлення даних: оновлює дані в базі даних за допомогою Prisma mutations;
- алгоритм видалення даних: видаляє дані з бази даних використовуючи Prisma API.

Було спроектовано та реалізовано програмний додаток на основі веб-фреймворку Next.js з використанням серверних API роутів Next.js та Prisma ORM для взаємодії з базою даних PostgreSQL, проведено тестування коректності функціонування програмного додатку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Богуш Л. Г. Сфера зайнятості в умовах масштабних протиепідемічних заходів: здобутки, проблеми, недоліки регулювання. *Підприємництво та інновації,* 2021. № 21. С. 76–83. URL: https://doi.org/10.37320/2415-3583/21.13 (дата звернення: 25 жовтня 2024 року).

2. Боярчук Л. В., Холодницька А. В. Забезпечення ефективного управління роботою служби зайнятості України в умовах невизначеності. *Вісник Харківського національного технічного університету сільського господарства*, вип. 162 «Економічні науки». 2015. С. 107-113.

3. Гасюк І., Баюк М. Можливості інтелектуальних систем для автоматизації рутинних задач та процесів публічного управління. *Наукові інновації та передові технології.* 2023. № 11(25). С. 54 - 60. URL: https://doi.org/10.52058/2786-5274-2023-11(25)-65-81 (дата звернення: 21 жовтня 2024 року).

4. Глевацька Н. М. Державна служба зайнятості: основні проблеми та напрями їх вирішення / Н. М. Глевацька, С. Е. Зелінський. *Конкурентоспроможна модель інноваційного розвитку економіки України :* матеріали V Міжнар. наук.-практ. конф., м. Кропивницький, 08 груд. 2022 р. – Кропивницький : ЦНТУ, 2022. С. 64–66.

5. Даніч М. О. Удосконалення діяльності Державної служби зайнятості з урахуванням світового досвіду. URL: https://er.nau.edu.ua/server/api/core/bitstreams/78754de2-7bf9-426d-a0f9-2363ba19ca98/content (дата звернення: 25 листопада 2024 року).

6. Деякі питання державного управління у сфері зайнятості населення : Постанова Каб. Міністрів України від 05.03.2014 № 90 : станом на 12 берез. 2020 р. URL: https://zakon.rada.gov.ua/laws/show/90-2014-п#Text (дата звернення: 25 жовтня 2024 року).

7. Інноваційні форми організації праці в умовах діджиталізації економіки України: теоретико-прикладний нарис : монографія / О. М. Ярошенко, О. Г.Середа, Л. Ю. Малюга, О. Є. Луценко, О. В. Таможанський; за заг. ред. проф. О. М. Ярошенка. Харків: Юрайт. 2023. 240 с.

8. «Електронний кабінет роботодавця» URL: https://www.dcz.gov.ua/cabinet (дата звернення: 23 листопада 2024 року).

9. Єрофєєнко Л. В. Проблеми трудового законодавства відносно внутрішньо переміщених осіб / Л. В. Єрофєєнко, А. О. Григорова. *Проблеми та перспективи розвитку сучасної науки в країнах Європи та Азії: збірник наукових праць:* XXIII міжнар. науково-практ. інтернет-конф. 31 січня 2020. Переяслав-Хмельницький : Державний педагогічний ун-т ім. Г. Сковороди., 2020. С.100-102. URL: http://repository.hneu.edu.ua/handle/123456789/22898

10.Збрицька, Т. П., Мінін, В. С. Проблеми молоді на ринку праці в сучасних умовах. *Вісник соціально-економічних досліджень:* зб. наук. праць. Одеса: Одеський національний економічний університет, 2021. No 2 (77). С. 55–65 DOI: https://doi.org/10.33987/vsed.2(77).2021.55-65

11.Зуб М. Я., Шуліка Д. А. Етапи становлення державної служби зайнятості як основного елементу інституційного наповнення інфраструктури ринку праці. *Вісник Хмельницького національного університету*. 2017. № 6. Том 3. С. 126 -133.

12.Ковач В.О. Використання цифрових сервісів для покращення надання адміністративних послуг населенню. *Інформаційно-комп'ютерні технології: стан, досягнення та перспективи розвитку:* тези доповідей наук.- практ. конф. Житомир: Житомирська політехніка. 2021. С.92-93.

13.Коноваленко І. В. Платформа .NET та мова програмування C# 8.0: навчальний посібник / І. В. Коноваленко, П. О. Марущак. Тернопіль : ФОП Паляниця В. А., 2020. 320 с.

14.Косяченко Г.Л., Тюлєнєв С.А. Оцінка ефективності діяльності базових центрів зайнятості. Ринок праці та зайнятість населення, 2017. №2 (51). С. 153-158.

15.Мартинець, В. В., Антоненко, А. Г., Коломієць, Д. А. Оцінка ефективності діяльності центру зайнятості. Вісник Сумського державного університету. Серія Економіка, 2021. № 3. С. 263-269. URL:https://doi.org/10.21272/1817-9215.2021.3-30 (дата звернення: 22 листопада 2024 року).

16.Панюкова Д. Адміністративно-правовий статус державної служби зайнятості України. Вісник Харківського національного університету імені В. Н. Каразіна. Серія «ПРАВО», 2020. Випуск 29. С. 213-220.

17.Персональний кабінет безробітного URL: https://www.dcz.gov.ua/cabinet (дата звернення: 23 листопада 2024 року).

18.Питання Міністерства цифрової трансформації: постанова Кабінету Міністрів України від 18.09.2018 № 856. URL: https://zakon.rada.gov.ua/laws/show/856-2019-%D0%BF. (дата звернення: 25 листопада 2024 року).

19.Про загальнообов'язкове державне соціальне страхування на випадок безробіття : Закон України від 02.03.2000 № 1533-III : станом на 19 груд. 2024 р. URL: https://zakon.rada.gov.ua/laws/show/1533-14#Text (дата звернення: 13 листопада 2024 року).

20.Про зайнятість населення : Закон України від 05.07.2012 № 5067-VI : станом на 1 січ. 2025 р. URL: https://zakon.rada.gov.ua/laws/show/5067-17#Text (дата звернення: 31 жовтня 2024 року).

21.Про затвердження Державної стратегії регіонального розвитку на 2021-2027 роки : Постанова Каб. Міністрів України від 05.08.2020 № 695 : станом на 22 серп. 2024 р. URL: https://zakon.rada.gov.ua/laws/show/695-2020-п#Text (дата звернення: 30 жовтня 2024 року).

22.Про затвердження Положення про Державну службу зайнятості : Наказ М-ва розвитку економіки, торгівлі та сіл. госп-ва України від 16.12.2020 № 2663 : станом на 7 черв. 2023 р. URL: https://zakon.rada.gov.ua/laws/show/z1305-20#Text (дата звернення: 29 жовтня 2024 року).

23.Про затвердження Порядку реєстрації, перереєстрації безробітних та ведення обліку осіб, які шукають роботу : Постанова Каб. Міністрів України від 19.09.2018 № 792 : станом на 10 трав. 2023 р. URL: https://zakon.rada.gov.ua/laws/show/792-2018-п#Text (дата звернення: 28 жовтня 2024 року).

24. Степанова, Є. Д. Стратегія розвитку Державного центру зайнятості з урахування світового досвіду. URL: https://er.nau.edu.ua/server/api/core/bitstreams/8d147d62-0882-410a-a85b-2 bdfeda22573/content (дата звернення: 28 листопада 2024 року).

25.Улітко Р. І. Дослідження ефективності застосування та напрямів розвитку інформаційно- комунікативних технологій у публічному управлінні, на прикладі Зарічненського управління Вараської філії Рівненського обласного центру зайнятості» : робота на здобуття кваліфікаційного ступеня магістра : спец. 281 - публічне управління та адміністрування / наук. кер. М. Ф. Зяйлик. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2024. 81 с. URL: http://elartu.tntu.edu.ua/handle/lib/47163 (дата звернення: 25 січня 2025 року).

26.Цифрова трансформація економіки: мікро- та макроаспекти : колективна монографія / за заг. ред. Н.А. Мазур, д.е.н., проф.; Кам’янець-Подільськ. нац. ун-т ім. І. Огієнка. Чернівці : Чернівец. нац. ун-т ім. Ю. Федьковича, 2022. 440 с.

27.Чернега А. Л. Структура, основні цілі, функції та аналіз показників ефективності функціонування органів служби зайнятості на різних рівнях / А. Л. Чернега // Бионика интеллекта, 2016. – №1 (86). С. 125–129.

28.Як рекрутеру використовувати штучний інтелект. Команда HURMA. 31 серпня 2023. URL: https://hurma.work/blog/yak-rekruteru-vykorystovuvaty-shtuchnyj-intelek/ (дата звернення: 25 листопада 2024 року).

29.Drucker C. Die sechs Top Recruiting Trends 2025. URL: https://www.stepstone.at/e-recruiting/blog/recruiting-trends/ (дата звернення: 2 січня 2025 року).

30.E-Recruiting – Hintergründe und Trends. Beispiele für erste Erfahrungen bei der Siemens AG / A. de la Fontaine et al. Strategien der Personalentwicklung. Wiesbaden, 2002. P. 359–379. URL: https://doi.org/10.1007/978-3-663-11541-0_23 (дата звернення: 26 листопада 2024 року).

31. Coloris. URL: https://coloris.com.ua/ (дата звернення: дата звернення: 29 жовтня 2024 року).

32. Code.visualstudio. URL: https://code.visualstudio.com/ (дата звернення: 26 грудня 2025).

33. Angular.dev. URL: https://angular.dev/ дата звернення: 16 грудня 2024 року).

34. Angular Material. URL: https://material.angular.io/ (дата звернення:16 грудня 2024 року).

35.Rippling. URL: https://www.rippling.com/ (дата звернення:25 грудня 2024 року).

36. Державна служба зайнятості. URL: https://dRipplingcz.gov.ua/ (дата звернення: 2 січня 2025 року).

37. Платформа профорієнтації та розвитку кар’єрию URL: https://profi.dcz.gov.ua/ (дата звернення: 2 січня 2025 року.

38. І.В Машкіна, ВО Абрамов, ТІ Носенко, ЄВ Іваніченко.Навчально-методичний посібник до виконання і захисту кваліфікаційної роботи бакалавра спеціальностей 122 Комп’ютерні науки для

здобувачів вищої освіти денної форми навчання, Київський столичний університет імені Бориса Грінченка. Київ, 2024.- 43 с.

39. Теоретичні та практичні аспекти використання математичних методів та інформаційних технологій в освіті й науці:моногр. / за заг. ред. О. Литвин. — К.: Київ. ун-т ім. Б. Грінченка,2021. — 332 с.

40. Яскевич, Владислав Олександрович (2023) *JavaScriptбібліотека React Навчальний посібник для студентів спеціальності Комп'ютерні науки* Київський університет імені Бориса Грінченка, Україна, Київ. https://elibrary.kubg.edu.ua/id/eprint/48587/