

Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

**«Розробка Web-порталу з працевлаштування з
використанням PhpMyAdmin»**

Студента 4 курсу, 13 групи,
спеціальності
122 «Комп'ютерні науки»

Старовойтова
Олександра
Івановича

підпис студента

Науковий керівник
кандидат фізико-математичних наук,
доцент

Самойленко Ганна
Тимофіївна

підпис керівника

Гарант освітньої програми
кандидат технічних наук, доцент

Демідов Павло
Георгійович

підпис керівника

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____ **Затверджую**
Пурський О.І.
« » грудня 2020р.

Завдання на випускний кваліфікаційний проект студенту

Старовойтову Олександрю Івановичу
(прізвище, ім'я, по батькові)

- Тема випускної кваліфікаційної роботи (проекту)
«Розробка Web-порталу з працевлаштування з використанням PhpMyAdmin»
Затверджена наказом ректора від «04» грудня 2020 р. № 4111
- Строк здачі студентом закінченої роботи 31 травня 2021 року
- Цільова установка та вихідні дані до роботи
Мета роботи: обґрунтування та розробка Web-порталу, з урахуванням сучасних тенденцій побудови організаційних та функціональних інформаційних структур сервісу працевлаштування.
Об'єкт дослідження: процес проектування Web-порталу з працевлаштування .
Предмет дослідження: засоби створення Web-порталу з працевлаштування .
- Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.
2	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.
3	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.

6. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. Сучасні технології в розробці веб-порталу.

1.1 Концепція веб-порталу. Архітектура сучасного веб-порталу

1.2. Огляд існуючих засобів проектування

Висновки до розділу

РОЗДІЛ 2. Організація розробки веб-порталу працевлаштування

2.1 Функціональні вимоги до веб-порталу працевлаштування

2.2. Моделювання інформаційного забезпечення

Висновки до розділу

РОЗДІЛ 3. Розробка Web-порталу з працевлаштування

3.1. Розробка користувацького інтерфейсу

3.2. База даних та програмування сайту

3.3. Оцінка отриманих результатів проектування

Висновки до розділу

ВИСНОВКИ ТА РЕЗУЛЬТАТИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор .	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	01.10.2020	01.10.2020
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	15.12.2020	15.12.2020
3	<i>Вступ</i>	03.02.2021	
4	<i>РОЗДІЛ 1. Сучасні технології в розробці веб-порталу ..</i>	26.02.2021	
5	<i>РОЗДІЛ 2. Організація розробки веб-порталу працевлаштування</i>	06.04.2021	
6	<i>РОЗДІЛ 3 Розробка Web-порталу з працевлаштування.</i>	12.05.2021	
7	<i>Висновки</i>	15.05.2021	
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	20.05.2021	
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	26.05.2021	
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	27.05.2021	
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	31.05.2021	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	16.06.2021	

8. Дата видачі завдання «15» грудня 2020 р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Самойленко Г.Т.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Старовойтов О.І.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings visible.

_____ 30.05.2021 р.
(підпис, дата)

Випускна кваліфікаційна робота (проект) студента _____
(прізвище, ініціали)
може бути допущена до захисту в екзаменаційній комісії.

Завідувач кафедри _____ Пурський О.І.
(підпис, прізвище, ініціали)

5

Анотація

В ході виконання випускної кваліфікаційної роботи була спроектована і структурована концептуальна модель веб-порталу з працевлаштування. Далі були визначені функціональні вимоги до інтерфейсу та технічного оснащення сайту, на основі аналізу схожих продуктів на ринку.

Розробка веб-порталу складалася з чотирьох основних етапів: створення "скелета" ресурсу за допомогою мови розмітки HTML; розробка структурованої бази даних на основі СУБД «phpMyAdmin»; інтеграція веб-ресурсу з БД, силами мови програмування PHP; і, звичайно, фінальне оформлення створених сторінок за допомогою CSS і JavaScript.

Ключові слова: веб-портал, сайт, база даних, мова розмітки веб-сторінок, системи управління базами даних.

Abstract

During the final qualification work, a conceptual model of the web portal on employment was designed and structured. Next, the functional requirements for the interface and technical equipment of the site were determined, based on the analysis of similar products.

The development of the web portal consisted of four main stages: creating a "template" of the resource using the HTML markup language; development of a structured database based on the database "phpMyAdmin"; integration of a web resource with a DB, forces of PHP programming language; and, of course, the final design of the created pages using CSS and JavaScript.

Keywords: web portal, website, database, web page markup language, database management systems.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. Сучасні технології в розробці веб-порталу.....	10
1.1 Концепція веб-порталу. Архітектура сучасного веб-порталу.....	10
1.2. Огляд існуючих засобів проектування	17
Висновки до розділу 1.....	20
РОЗДІЛ 2. Організація розробки веб-порталу працевлаштування.....	21
2.1 Функціональні вимоги до веб-порталу працевлаштування.....	21
2.2. Моделювання інформаційного забезпечення.....	25
Висновки до розділу 2.....	27
РОЗДІЛ 3. Розробка Web-порталу з працевлаштування	29
3.1. Розробка користувацького інтерфейсу.....	29
3.2. База даних та програмування сайту.....	38
3.3. Оцінка отриманих результатів проектування.....	46
ВИСНОВКИ ТА РЕЗУЛЬТАТИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49

ВСТУП

Актуальність теми. На сьогоднішній день інтернет-технології торкнулися абсолютно всіх аспектів життя людства, надавши широкий спектр нових можливостей в усіх сферах діяльності. Різного роду послуги, навчання, пошук інформації - це лише мала частина того, що можуть надати доступні нам всім інтернет-технології.

Пошук роботи - одна з істотних сфер, представлених в глобальній мережі. Рано чи пізно, кожен з нас замислюється про пошук або зміну роботи. Звичайно, ми можемо шукати сайти конкретних компаній, сподіваючись, що знайдемо список всіх доступних вакансій і серед них буде бажана. Але, як показує світова практика, рядовий фахівець хоче розглянути безліч доступних варіантів роботи для його спеціалізації і порівняти їх.

Тут на допомогу приходять сайти з працевлаштування, які можуть містити пропозиції компаній різного масштабу і сфери діяльності.

Мета проекту: створення веб-порталу з працевлаштування, який дозволить переглядати безліч вакансій від різних підприємств. Для реалізації поставленої мети визначимо наступні етапи:

- Аналіз механізмів функціонування ресурсів по працевлаштуванню.
- Дослідження існуючих методів та архітектури для реалізації сучасних веб-ресурсів.
- Розробка функціональної моделі веб-порталу працевлаштування.
- Проектування інформаційно-логічної системи взаємодії модулів веб-ресурсу.
- Верстка та програмування сайту.
- Інтеграція веб-додатку з БД.
- Оформлення та оптимізація порталу.

Об'єкт дослідження є сучасні програмні засоби для проектування та розробки Веб-порталу для пошуку роботи та підбору кадрів в глобальній мережі «Інтернет».

Предметом дослідження є методи та способи, які використовуються для проектування та реалізації веб-ресурсів.

Інструменти для реалізації: Для проектування інтерфейсу порталу скористаємося онлайн-сервісом для розробки інтерфейсів і прототипування «Figma». За його допомогою створимо макет сайту, на який будемо орієнтуватися під час розробки.

Щоб коректно спроектувати структуру бази даних необхідно буде визначити ключові поля таблиць та їх типи, в залежності від виду інформації яка буде циркулювати в системі. Для розробки і адміністрування БД використаємо СУБД «phpMyAdmin».

Верстка сайту буде здійснюватися на мові розмітки тексту «HTML», а програмування і інтеграція з БД за допомогою мови програмування «PHP».

Візуальне оформлення та динамічність інтерфейсу порталу буде реалізована за рахунок «CSS» та «JavaScript». В якості середовища для розробки використаємо phpStorm - крос-платформний інтегрований фреймворк. Цей додаток дозволить сконцентрувати етапи верстки, програмування та інтеграції з БД в одному середовищі, що оптимізує розробку і значно скорочує час на пошук рішень для формування цілісності проекту.

Практичне значення. В результаті виконання проекту ВКР очікується отримати зручний веб-портал, який зможе надати швидкий доступ користувачам до актуальної інформації про вакансії різних компаній.

РОЗДІЛ 1.

СУЧАСНІ ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБ-ПОРТАЛУ.

1.1 Концепція веб-порталу. Архітектура сучасного веб-порталу.

Концепція веб-порталу.

Розробка будь-якого функціонального продукту ніколи не обходиться без обов'язкових початкових етапів, таких як проектування, визначення задач та потреб кінцевих користувачів, які повинні задовольнятися тощо. Одним з інструментів первинного проектування є концептуальна модель, або концепція продукту.

В нашому випадку об'єктом розробки є веб-портал, який також вважається певним «продуктом» та повинен мати свою концепцію. Що ж таке концепція сайту(веб-порталу)? В широкому сенсі – це перелік документації, яка була сформована в результаті аналізу сфери діяльності, цільової аудиторії та її територіальної приналежності, особливості регіону тощо. Таким чином, можна вважати, що концепція порталу – це деякий перелік фактів, на які нам потрібно опиратися при проектуванні та створенні сайту, а весь цикл реалізації даного проекту буде складатися з семи етапів, деякі з яких можна виконувати паралельно(рис. 1.1).

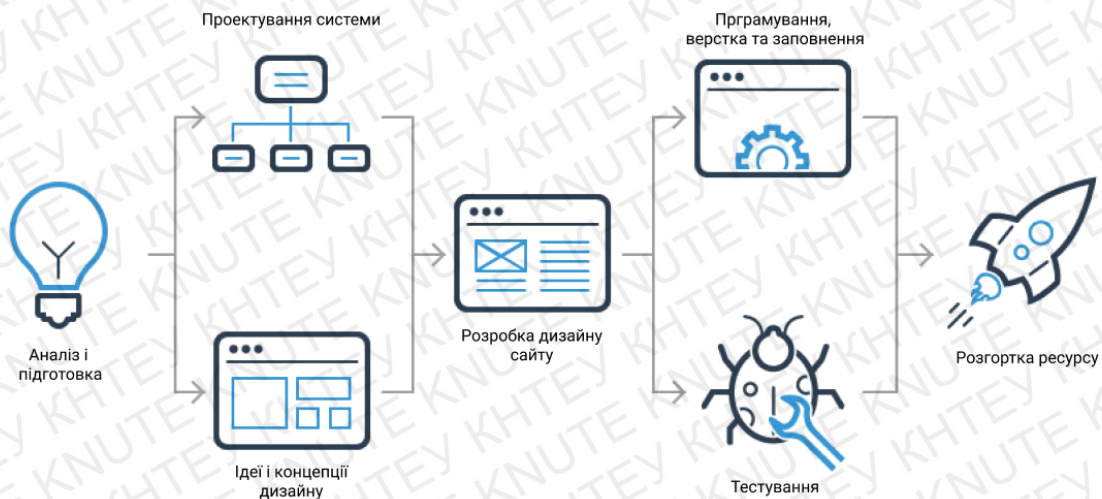


Рис. 1.1 – Загальний перелік етапів для реалізації веб-порталу
працевлаштування.

Будь-яка концептуальна модель починається з так званого «Брифу(з англ. Brief)» - це невеликий список пунктів, які зазвичай потребують аналізу та структуризації:

- Аналіз цільової аудиторії
- Пошук та аналіз продуктів, які схожі за кінцевою метою використання.
- Семантичне ядро сайту.
- Концепція дизайну.
- Матеріальне та змістовне забезпечення сайту.

Аналіз цільової аудиторії. Процес, під час якого оцінюють інтереси можливих споживачів, види та типи інформації які вони захочуть використовувати, користуючись порталом. Також аналізують регіональну приналежність аудиторії, мову, на якій суб'єкти, скоріш за все, захочуть обмінюватись інформацією. Така інформація дає змогу розробляти

функціональний продукт, одразу опираючись на особливості потенціальної аудиторії, що в підсумку може зіграти на користь.

Дослідження схожих продуктів. Аналіз конкурентів або можливих потенціальних партнерів – також дуже важливий етап при створенні концептуальної моделі. Досліджуючи вже активні продукти, можна з'ясувати що в першу чергу є головним для потенційного користувача, а що другорядним. Дослідивши декілька продуктів, що вже існують на ринку якийсь час – можна зробити висновок, що є обов'язковим для успішного старту вашого проекту.

Семантичне ядро порталу. Дуже важливий та трудомісткий етап розробки концептуальної моделі. Формування семантичного ядра стартує після аналізу потенціальної аудиторії та ринку. В широкому сенсі, семантичне ядро – це деякий набір слів або словосполучень, які максимально чітко та коротко характеризують діяльність або направленість сайту.

Семантичне ядро складається з комплексу пошукових запитів користувачів, які мають різну частоту згадування та рівень конкурентоспроможності на територіально обмеженій локації. Таким чином, етап наповнення семантичного ядра майбутнього сайту починається до початку створення сайту і триває паралельно його розробці, постійно оптимізуючись.

Концепція дизайну сайту. Створення концептуальної моделі пов'язано з розробкою структури і навігації на майбутньому веб-ресурсі.

Структура сайту, сформована на основі аналізі схожих продуктів, потенціальних користувачів та ключових запитів, представляє собою приблизну схему порталу. Вона стане основою для створення макету інтерфейсу та проектування ключових сторінок та блоків.

Також, на цьому етапі визначають основні функціональні вузли порталу, та закладають обов'язкові вимоги до зручності та гнучкості їх використання.

Змістовне наповнення сайту. Розробка концепції сайту передбачає розробку структури (переліку) матеріалів, призначених на для розміщення на порталі. На основі сегменту та цільової аудиторії визначають тематику, стиль подання, характер наповнення та місцезнаходження інформації на веб-ресурсі. Загалом, виділяють наступні основні блоки:

- Контактні дані.
- Інформаційно-текстові блоки.
- Графічні матеріали(фото-, відео-файли, банери).
- Колекції або галереї, представлені у вигляді таблиць або блоків.

В ході змістовного аналізу ми отримали дані, на основі яких створили ідеологічну основу проекту, тобто концептуальну модель сайту. Наступним кроком буде оцінка цієї моделі, де потрібно відповісти на ключові питання:

1. Наскільки концепція відповідає сучасним тенденціям?
2. Наскільки концепція відповідає потребам і вимогам потенційної аудиторії?
3. Наскільки приваблива дана концепція для потенційної аудиторії?

Концепція веб-порталу - це необхідний інструмент, який дозволяє скласти чітке бачення сайту, розуміння цільової аудиторії, її потреб. Концепція дозволяє зробити сайт, який буде повністю відповідати очікуванням клієнтів і ефективно працювати. Це дуже важливий етап розробки сайтів, без якого немає чіткого розуміння, куди рухатися і як реалізовувати проект.

Архітектура сучасного веб-порталу.

Під поняттям «архітектура веб-порталу» найчастіше розуміють структуру сторінок веб-ресурсу, певні шаблони та патерни, які визначають наповнення функціональних блоків. Головною метою проектування архітектури є зручність, гнучкість та функціональність інтерфейсу, який зміг би істотно полегшити пошук інформації.

Поняття «архітектура» охоплює повний спектр інформації, що циркулює на ресурсі: посилання, навігаційні панелі, пошук, заголовки, вказівники тощо. Окрім цього, архітектура веб-порталу повинна включати «back-end» структуру. Тобто окрім візуального представлення інформації на сайті, проектується і програмний «двигун» сайту. Веб-портали з якісно спроектованою архітектурою значно зручніші як для звичайного користувача, що є основною ціллю, так і для розробника.

По-перше, розглянемо якість та «привітливість» користувацького інтерфейсу. Якщо користувач на протязі 10 секунд не може знайти потрібну йому інформацію на сторінці – це свідчить про незручність інтерфейсу. Зараз швидка і зручна робота з сайтом є однією із основних причин використовувати його.

Щоб забезпечити надійну та швидку роботу веб-ресурсу можна використати основний шаблон проектування програмного забезпечення – «Functional design»(шаблон функціонального дизайну). Цей патерн гарантує, що кожен функціональний модуль додатку має одну конкретну задачу та виконує її з мінімальним числом побічних ефектів на інші частини програмних блоків. Такий підхід робить систему більш простою і надійнішою, а розробку блоків швидшою та дешевшою.

По-друге, розглянемо можливість розширювати ресурс, впровадити нову функціональність або оновити основні вузли. Зазвичай, при проектуванні архітектури необхідно заздалегідь закладати принцип модульності та низької залежності компонентів. Такі заходи дозволять у майбутньому замінити цілі функціональні блоки сайту, і це не матиме істотного негативного впливу на кінцевого споживача.

Реалізація «модульності» для будь-якого роду програмних продуктів може досягатися різними способами, але найпоширенішим вважають шаблон проектування MVC(Model, View, Controller або Модель, Представлення, Контролер). Даний патерн представляє собою певну схему розподілу даних інформаційної системи на три окремих блоки, які пов'язані загальною метою, але не залежать один від одного на низькому рівні.

Модель – функціональний блок, який працює з даними та інформацією, що циркулюють в системі. В програмуванні модель відповідає за роботу з базами даних, тобто містить набір запитів, проводить валідацію вхідних та вихідних даних тощо.

Представлення – говорить саме за себе. Блок, який відповідає за взаємодію користувача з системою. В широкому сенсі – він реагує на команди користувача та візуалізує дані, які були опрацьовані іншими блоками.

І, звичайно, контролер – функціональний блок, який контролює та спрямовує команди користувачів, введені дані, запити, тобто складає основу логічної моделі інформаційної системи.

Такий підхід до проектування програмного забезпечення дозволяє модифікувати кожен з трьох компонентів незалежно від інших.

Саме тому, обираючи той чи інший певний шаблон проектування системи варто орієнтуватися на сформовані вами, або замовником вимоги і наступні обов'язкові властивості системи:

Доступність. Спроможність ресурсу надати необхідну інформацію в будь-який момент часу.

Продуктивність. Архітектура повинна містити певні метрики, які дозволять кількісно оцінити ту чи іншу функціональність. Наприклад, пошук по сайту – повинен мати якісну оцінку швидкості роботи і, окрім цього, надавати необхідну інформацію.

Надійність. Архітектура будується таким чином, щоб забезпечувати максимально швидко і надійну видачу інформації.

Здатність до масштабування. При розробці архітектури веб-порталу необхідно мислити в перспективі. Необхідно на початку закласти можливість розширювати ресурс, не перероблюючи великі частини програмного коду і інтерфейсу.

Модульність. На сьогодні є одним із основних принципів при розробці сайтів. Полягає в тому, щоб заздалегідь підготувати програмні вузли коду для взаємної роботи з іншими ресурсами або додатками. Наприклад, архітектура «REST API» або «RESTful» може закладатися при створенні ресурсу, але ніяк не використовуватися певний час.

Керованість. Ще один пункт, який продовжує минулі два. Цей принцип вимагає забезпечення постійного оновлення і модифікації продукту, що розробляється. При цьому, його існуючі експлуатаційні якості не повинні погіршуватися.

Окрім нефункціональних вимог, які, скоріше, являють собою якісні характеристики ресурсу, існують базові рекомендації до створення «User Friendly» архітектури, або оптимізації вже існуючої. Наприклад:

- Правило «трьох кліків» - чим менше дій необхідно зробити, щоб добратися до необхідної сторінки – тим простіше її знайти.
- Використання ієрархічної структури URL при розміщенні сторінок в репозиторії.
- Використання фільтрів для відображення контенту.
- Контекстна залежність сторінок та перехресні посилання.
- Пагінація та безкінечні екрани прокрутки – дозволити користувачу самому обирати, як йому зручніше переглядати контент(переходити по сторінкам або постійно «скролити» вниз).
- Впровадження «хлібних крихт» - в широкому понятті це навігаційний ланцюжок по сайту, який показує шлях від кореня сайту до сторінки, на якій зараз знаходиться користувач.

1.2 Огляд існуючих засобів проектування

Сьогодні існує дуже багато різних способів і засобів для проектування і створення веб-ресурсів, але будь-який із способів можна віднести до однієї з трьох ключових категорій:

- Розробка в конструкторі сайтів, на основі запропонованих шаблонів.
- Проектування та розробка на основі CMS-систем.
- Самостійне проектування та розробка веб-ресурсу, з використанням мов програмування, СУБД та інших популярних інструментів і середовищ.
- Кожна з перерахованих категорій має свої переваги і недоліки.

Конструктори сайтів – це готове програмне рішення, яке дозволяє будувати веб-ресурс по модульному принципу. Такий підхід дозволяє спроектувати і створити сайт взагалі без знань програмування. Користувач отримує обмежену кількість шаблонів та модулів, які він може задіяти на своєму порталі. Вагомою перевагою такого підходу є простота використання,

адже ви просто обираєте необхідні блоки, а за верстку та програмний код відповідає конструктор. Також до переваг можна віднести низьку вартість.

Але недоліки, на мою думку, більш вагомі. Зазвичай такий підхід передбачає низький рівень оптимізації, адже містить дуже багато програмного коду, в порівнянні з самостійними рішеннями.

Другим істотним недоліком, на мою думку, є обмеження функціоналу – використовуючи конкретний конструктор сайтів, ви обмежуєтесь модулями, які представлені в каталозі. Якщо вам конче необхідно додати модуль, якого немає в переліку обраного конструктора – додати його, нажаль, буде майже неможливо. Окрім цього, нажаль, відсутня можливість формування «семантичного ядра» ресурсу, що не дозволить повноцінно провести SEO сайту.

CMS-системи - це комплекс програмних засобів для управління веб-контентом, що циркулює на ресурсі. В широкому розумінні - це базовий каркас і набір основних інструментів і надбудов, який дозволяє не тільки створити веб-портал, а й підтримувати його роботу, взаємодіяти з користувачами і оновлювати контент. Усі CMS-системи містять панель управління, зазвичай з примітивним інтерфейсом. Зазвичай, основною мовою програмування є PHP. Будь-яка CMS може дозволити створити навіть дуже складні рішення, такі як інтернет-магазини або великі корпоративні сайти з глибокої вкладеністю сторінок, але і у них є свої особливості.

До переваг CMS-систем віднести зручне управління контентом. Відсутність обмежень, в разі, якщо вам необхідно розширити функціонал ресурсу та велика кількість готових рішень та надбудов для різного роду задач(від динаміки інтерфейсу до систем SEO-оптимізації).

До недоліків CMS систем відносять потребу у базових знаннях верстки та програмування, на відміну від конструкторів сайтів. Також відмічають складність переносу розроблених продуктів та істотну невизначеність при

розробці великих ресурсів, адже складність розробки масштабних проектів за допомогою CMS-систем наближається до самостійної розробки.

Самостійна розробка веб-ресурсу вважається самим трудомістким підходом, але в той же час творчим і вільним, на відміну від двох минулих варіантів. Верстка, написання програмного коду, включення бази даних та оформлення потребує глибоких знань не тільки в мовах програмування, а й понятті архітектури, методологій, середовищ розробки тощо. Самостійна розробка займає більше часу, але дозволяє створювати продукти будь-якої складності. При цьому, в підсумку, ми отримуємо унікальний продукт, який буде відповідати конкретним вимогам та вирішувати конкретні задачі.

До безпосередніх переваг відносять повну свободу при розробці, розгортці, підтримці та розширенні. Звичайно, індивідуальність ресурсу, що дає широкі можливості для його просування в пошукових системах.

Недоліки, на мою думку, також достатньо вагомі. Наявність глибоких знань в програмуванні, підходах розробки та архітектурі веб-додатків. Окрім цього, в порівнянні з двома минулими підходами, самостійна розробка вимагає набагато більше часу і, зазвичай, істотно дорожча.

Висновки до розділу

Сучасні тенденції методологій та архітектури веб-додатків диктують суворі правила до проектування та розробки ресурсів. Успішність сайту вже давно залежить не тільки від його змістовного наповнення, а й від комплексу вимог та факторів, які повинні бути проаналізовані та структуровані на етапі побудови концептуальної моделі. Структуру сайту також варто проектувати заздалегідь, опираючись на загальноприйняті функціональні та не функціональні вимоги до «гнучких» ресурсів.

Вибір методу розробки також важливе питання, на яке можна отримати відповідь, проаналізувавши можливості, вартість готових програмних рішень та розробки з нуля.

На мою думку, окрім описаних вимог, яких потрібно дотримуватися при проектуванні та розробці веб-додатку, потрібно пам'ятати про унікальність та необхідність продукту. Адже саме цей фактор може зіграти основну роль для потенційної аудиторії.

РОЗДІЛ 2.

ОРГАНІЗАЦІЯ РОЗРОБКИ ВЕБ-ПОРТАЛУ ПРАЦЕВЛАШТУВАННЯ

2.1 Функціональні вимоги до веб-порталу працевлаштування

Функціональні вимоги будь-якого продукту передбачають комплекс завдань, які необхідно реалізувати в ході розробки. В нашому випадку, функціональні вимоги можна поділити на дві категорії:

Перша – функціональні властивості ресурсу, з якими буде взаємодіяти користувач. Наприклад функції реєстрації, редагування профілю, створення та оновлення вакантних пропозицій тощо.

Друга – включає вимоги до реалізації користувацького інтерфейсу: інструменти пошуку, сортування, інформаційні сторінки, навігаційні блоки, їх оформлення та розміщення.

При цьому, варто пам'ятати, що обидві категорії тісно пов'язані між собою і не повинні виключати один одного, або конфліктувати.

Почнемо з функціональних властивостей, які необхідно реалізувати для коректного функціонування порталу. В нашому випадку правильним рішенням буде звернутися до концептуальної моделі Minimal Viable Product (мінімально життєздатний продукт), а саме «Однофункціональний MVP» . Такий підхід передбачає визначення найголовніших функціональних потреб потенційної аудиторії, яких буде достатньо для аналізу актуальності продукту та розуміння, над чим необхідно працювати далі. Після етапу розгортки стартує компанія по збору побажань та рекомендацій, які можуть надходити від кінцевих користувачів та зацікавлених суб'єктів. Саме тому варто зосередитися на базових потребах і вимогах аудиторії, які потрібно реалізувати в першу чергу, щоб представити продукт.

По-перше, користувачі хочуть мати змогу зареєструвати свою компанію для публікування вакантних пропозицій. Для реалізації такої вимоги нам

необхідно створити сторінку реєстрації, де користувач зможе вносити інформацію про своє підприємство. Далі ці дані необхідно зберігати. В цьому нам допоможе база даних, куди за допомогою SQL-запитів буде вноситися введена інформація.

По-друге, кожна компанія, яка зареєструвалась на порталі, має необхідність авторизуватись на порталі для подальшої роботи.

По-третє, окрім даних про компанію, необхідно мати змогу створювати вакантні пропозиції. Скоріш за все, вони захочуть модифікувати дані або видаляти пропозиції.

На відміну від користувачів-компаній, також необхідно створити умови для людей, які будуть заходити на портал та знайомитись із пропозиціями. Функціональні вимоги цієї групи користувачів потребують впровадження зручної системи навігації по порталі, пошуку по конкретним вакансіям та очевидне розташування необхідної інформації: контактних даних компанії, вимог при прийнятті, перелік необхідних якостей тощо.

Окрім таких очевидних речей, неможна забувати про оформлення інтерфейсу. Підібрати спокійні кольори та шрифти, які легко сприймати при читанні. Загострити увагу користувача на основних блоках, які він, скоріш за все, буде використовувати тощо.

Таким чином можемо виділити основні функціональні вимоги, які потрібно буде реалізувати в ході розробки порталу(табл. 2.1).

Таблиця 2.1

Функціональні вимоги	
Функціональні властивості	Вимоги до інтерфейсу
Зберігати дані, які вводять користувачі.	Зручне та швидке переміщення по сторінкам порталу за допомогою навігаційного меню.

Можливість реєстрації та авторизації користувачів-адміністраторів.	Фільтрація вакантних пропозицій по посаді і компанії.
Створення, редагування та видалення вакантних пропозицій.	Пошук по ключовим словам.
	Перегляд всіх або однієї вакантної пропозиції з повним переліком інформації.

На мою думку, визначений список вимог несе мінімальний, але достатній рівень функціональних можливостей для задоволення основних потреб потенційних користувачів веб-порталу.

Виходячи з визначених вимог, побудуємо простий макет головної сторінки сайту за допомогою додатку «draw.io»(рис. 2.1).

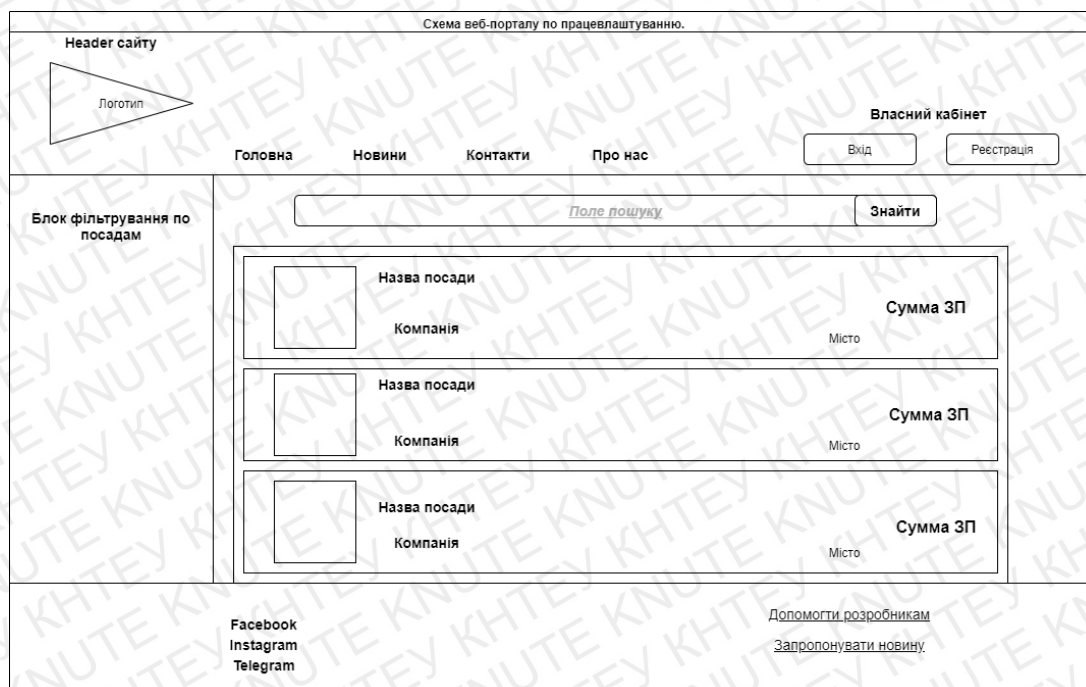


Рис. 2.1 – Макет головної сторінки сайту.

Далі визначимо інтерфейс функціоналу реєстрації та авторизації(рис. 2.2).

Схема веб-порталу по працевлаштуванню.

Header сайту


Головна
Новини
Контакти
Про нас

Власний кабінет
Вхід
Реєстрація

Блок фільтрування по посадам

Реєстрація

Логін
 Пароль
 Підтвердження пароля
Завантажити логотип
 Назва компанії
 Місто
 Адреса
 Контактний телефон
 E-mail
 Про компанію

Facebook
Instagram
Telegram

Допомогти розробникам
Запропонувати новину

Рис. 2.2 – Макет інтерфейсу реєстрації.

Сторінка авторизації буде мати схожу структуру та оформлення, для того, щоб дотримуватися загальної концепції(рис. 2.3).

Схема веб-порталу по працевлаштуванню.

Header сайту


Головна
Новини
Контакти
Про нас

Власний кабінет
Вхід
Реєстрація

Блок фільтрування по посадам

Вхід

Логін
 Пароль

[Забули пароль? З'яжіться з нами!](#)

Facebook
Instagram
Telegram

Допомогти розробникам
Запропонувати новину

Рис. 2.3 – Авторизація адміністратора.

В наступному кроці нам необхідно визначити яким чином буде відбуватися додавання вакантних пропозицій. Необхідно зауважити, що

функціонал створення вакансії буде доступним після авторизації адміністратора компанії. Додавання буде відбуватися на сторінці «власного кабінету»(рис. 2.4), також на цій сторінці можна буде переглядати, редагувати та видаляти попередні записи.

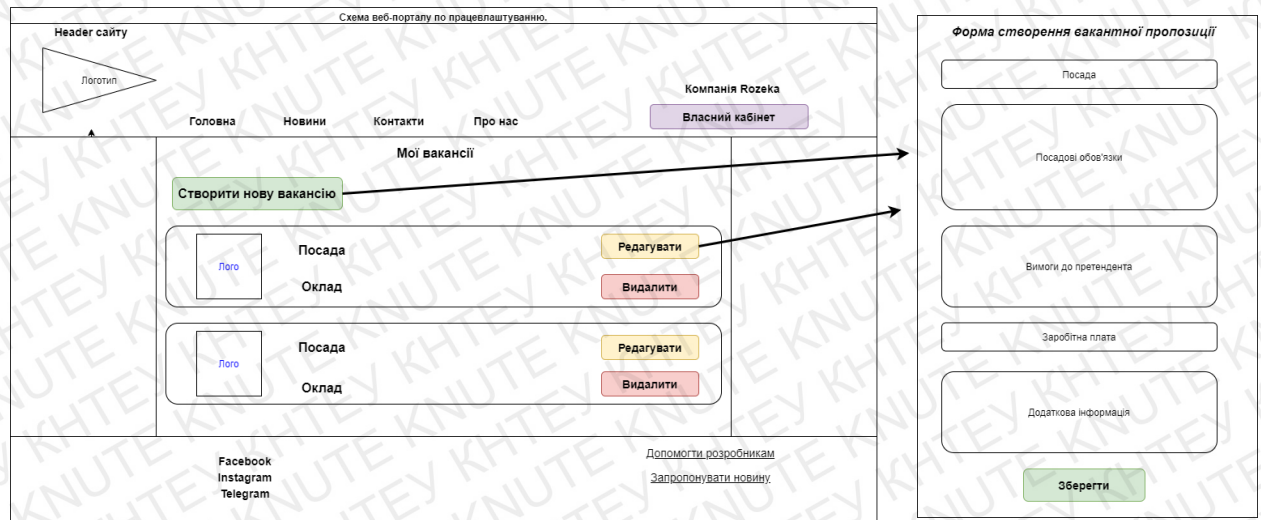


Рис. 2.4 – Створення та редагування вакантних пропозицій у власному кабінеті.

Функціонал видалення буде простішим за створення і редагування, адже для цього необхідно буде підтвердити видалення на окремій сторінці.

Також варто пам'ятати про суто інформаційні сторінки сайту, на яких будуть відображатись новини(оновлення) ресурсу, інформація про розробників, путівник по веб-порталу та контактні дані.

2.2. Моделювання інформаційного забезпечення

Перший етап моделювання інформаційного забезпечення порталу передбачає класифікацію даних та їх типів, що будуть попередньо прийматися системою, зберігатися та модифікуватися.

В широкому сенсі, при проектуванні інформаційної нам необхідно створити такі умови для циркуляції інформації на ресурсі, які зможуть забезпечити цілісність, доступність та контрольованість.

В нашому конкретному випадку, інформаційне забезпечення веб-порталу буде здійснюватися шляхом внесення інформації користувачем через інтерфейс, та введених даних до бази даних. Тому попередньо класифікуємо інформацію, що може надходити:

- Облікові дані та інформація про компанію.
 - Дані для входу у власний кабінет.
 - Контактні дані.
 - Додаткова інформація про підприємство.
- Вакантні пропозиції компаній.
 - Назва вакансії.
 - Обов'язки та вимоги до кандидата.
 - Додаткова інформація.

Знаючи вхідні дані, можемо перейти до проектування структури бази даних та її таблиць.

По-перше, розглянемо структуру таблиці 2.2, де буде зберігатися інформація про компанії:

Таблиця 2.2

Таблиця «Компанії»	
ID	Ідентифікатор для індексації компаній в системі.
Логін	Унікальне ім'я користувача в системі.
Пароль	Пароль для входу в систему.
Назва	Повна назва організації.
Логотип	Картинка логотипу компанії.
Телефон	Телефон для контакту.
E-mail	Поштова адреса.
Місто	Місто, в якому знаходиться компанія.
Адреса	Адреса.
Додаткова інформація	Будь-яка додаткова інформація про підприємство.

Вказані в таблиці поля будуть заповнюватися при реєстрації, та зберігатися в базі даних. Далі, при створенні вакантної пропозиції основна інформація з таблиці «Компанії» буде транслюватися в окремий запис, в залежності від того, який користувач її розмістив на ресурсі.

Такого результату можна буде досягти шляхом «зв'язування» таблиць за допомогою полів-індексів, що вказано на рисунку 2.5.

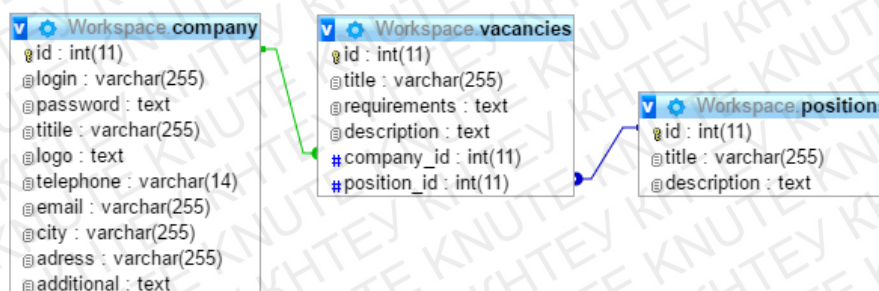


Рис. 2.5 – Структура таблиць бази даних.

За допомогою кольорових з'єднувальних ліній продемонстровано зв'язок таблиці «Вакансії» з таблицями «Компанії» та «Посада». Такий підхід дозволить уникнути створенню дублів посад на ресурсі, та позбавить необхідність кожний раз заповнювати інформацію про компанію при створенні вакансії.

Висновки до розділу 2

В даному розділі, в першу чергу, розглядається процес первинного проектування основних модулів веб-порталу, на основі визначених функціональних вимог.

В ході моделювання було визначено необхідні інтерфейсові модулі, за допомогою яких користувач зможе взаємодіяти з ресурсом. Окрім цього, враховуючи вимоги у додаванні та оновленні даних на ресурсі, спроектовано програмні вузли, які будуть відповідати за вищеперераховані функції.

Якщо розглядати веб-ресурс на високому рівні абстракції, його компоненти будуть взаємодіяти наступним чином(рис. 2.6).

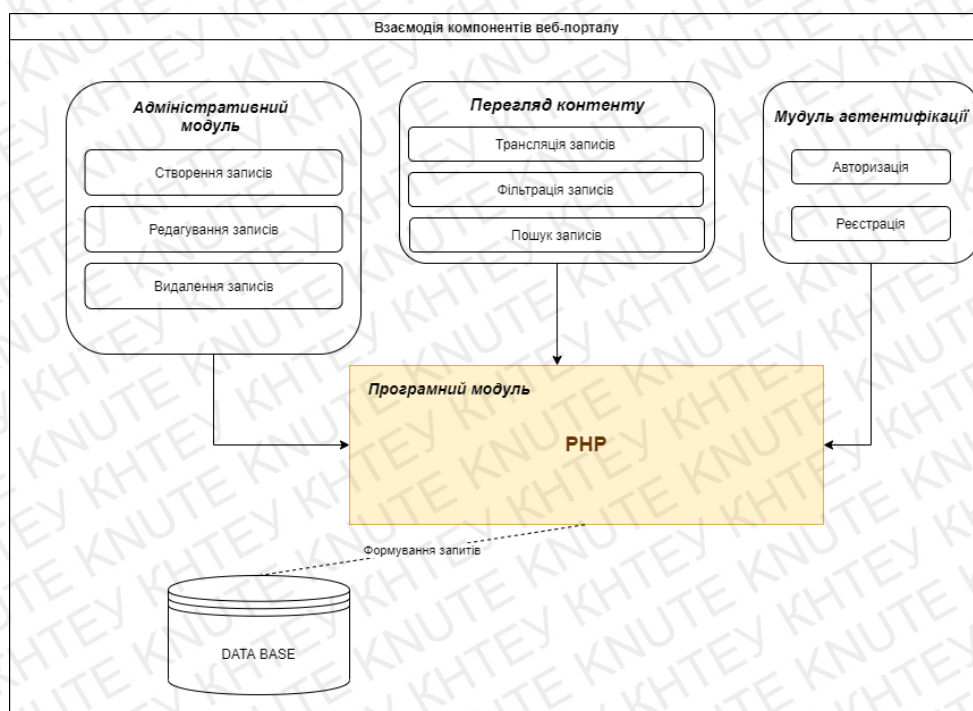


Рис. 2.6 – Взаємодія компонентів веб-порталу.

РОЗДІЛ 3.

РОЗРОБКА WEB-ПОРТАЛУ З ПРАЦЕВЛАШТУВАННЯ

3.1. Розробка користувацького інтерфейсу

Інтерфейс є невід’ємною частиною будь-якої інформаційної системи, яку потрібно розробляти з орієнтацією на кінцевого користувача. В моєму випадку є дві групи користувачів, які будуть взаємодіяти з веб-порталом:

Компанії можуть реєструватися в системі, розмішувати вакантні пропозиції. Саме тому необхідний простий і зручний інтерфейс реєстрації та авторизації у системі.

Також для актуалізації інформації на порталі, зареєстрованим користувачам(компаніям) буде дозволено редагувати власний обліковий запис, розміщені вакансії, а також видаляти їх.

Друга група – користувачі, які переглядають вакансії на ресурсі. Для перегляду інформації на сайті реєстрація необов’язкова. Але для перегляду вакансій також необхідно додати відповідний інтерфейс, наприклад:

- Фільтрація вакансій по компаніям – обравши фільтр, користувач переглядатиме пропозиції обраного підприємства.
- Пошук по ключовим словам – користувач може шукати вакантні пропозиції по назві, сумі заробітної плати, вимогам тощо.
- Розширений перегляд – окрема сторінка, де можна детально ознайомитися з кожною пропозицією. Адже на головній сторінці виводиться основна інформація про вакансії: назва, вимоги, місячний оклад.

Окрім спеціалізованого інтерфейсу для кожної групи користувачів, необхідно реалізувати зручний перехід по ключовим сторінкам, наприклад новини сайту, контакти, та зворотній зв’язок. Хоч такі сторінки несуть суто інформаційний характер, але для кінцевого користувача така інформація може знадобитися в будь-який момент. Тим більше, це істотний крок до

поліпшення інтерфейсу та інформаційного забезпечення ресурсу на основі відгуків користувачів.

На основі визначених вимог розробив інтерфейс головної сторінки сайту. Як видно на рисунку 3.1, в «шапці» розміщені меню з кнопками для переходу по ключовим сторінкам сайту. Головна сторінка, контакти, новини, вхід у власний кабінет та реєстрація. Під «шапкою» розміщений мотиваційний банер.



Рис. 3.1 – Головна сторінка сайту. Основне меню.

Далі бачимо поле пошуку (рис. 3.1) та кнопку «Пошук». Пошук можна проводити по всім вакансіям сайту.

Нижче розташований основний інформаційний блок, де виводяться вакантні пропозиції всіх компаній (рис. 3.2). Відображення вакансій на сторінці можна сортувати по зростанню та спаданню за допомогою відповідної кнопки.

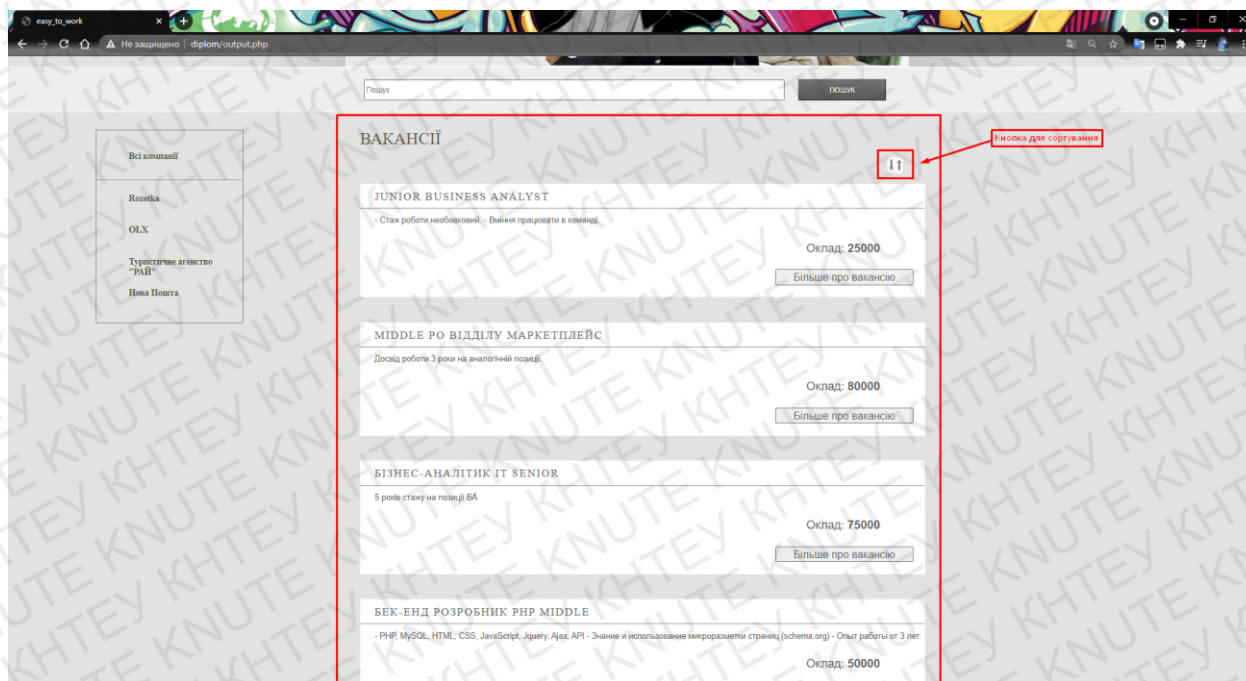


Рис. 3.2. – Блок для виводу вакансій.

Кожна вакансія представляє собою окремий блок з заголовком, вимогами та сумою окладу(рис №). Окрім інформаційних полів, кожна картка має функціональну кнопку для детального перегляду на окремій сторінці.



Рис. 3.3 – Функціональні кнопки для детального перегляду

Сторінка для детального перегляду включає всю інформацію про вакансію:

- Назву вакансії
- Сферу діяльності
- Вимоги та посадові обов'язки
- Місячна заробітна плата

Окрім цього, сторінка детального перегляду містить інформацію про компанію(рис. №):

- Контактні дані(телефон, електронна адреса)
- Фізична адреса
- Додаткову інформацію
- Логотип

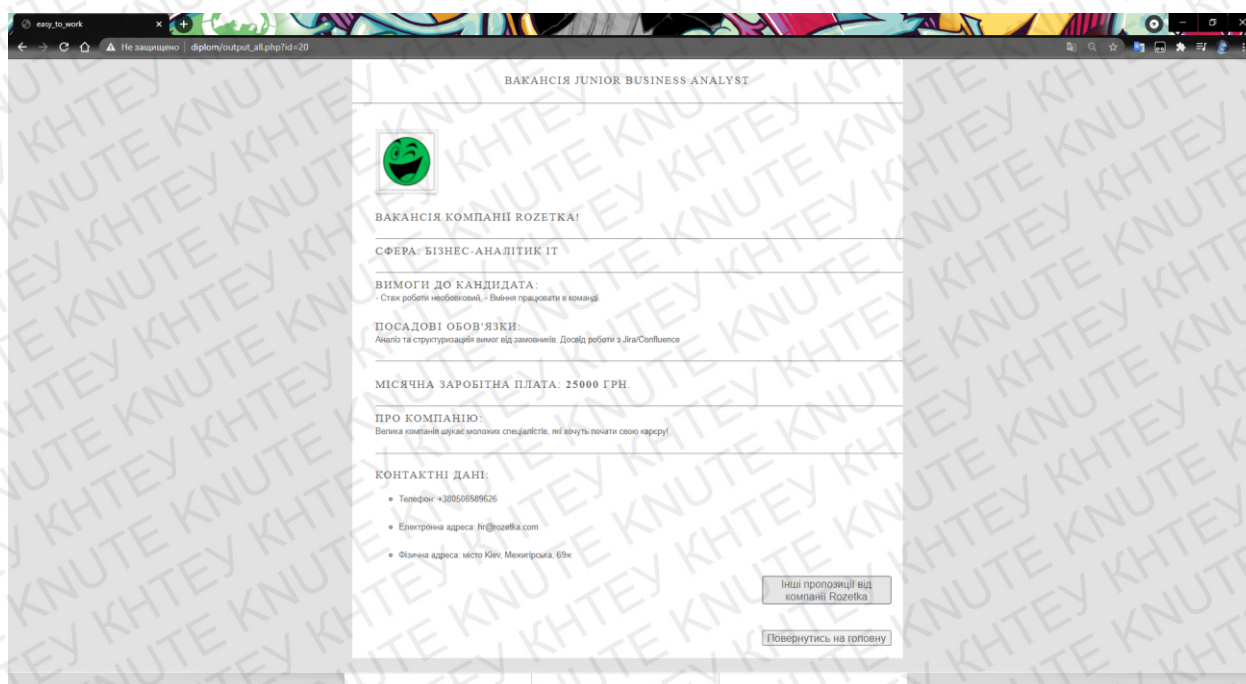


Рис. 3.4 – Детальна інформація про вакансію.

Внизу сторінки розміщені кнопки для швидкого повернення на головну сторінку та перехід до інших вакансій даного підприємства.

Також, для того щоб користувач міг сконцентруватися на перегляді вакансій конкретної компанії мною було додано можливість фільтрації(рис. №).

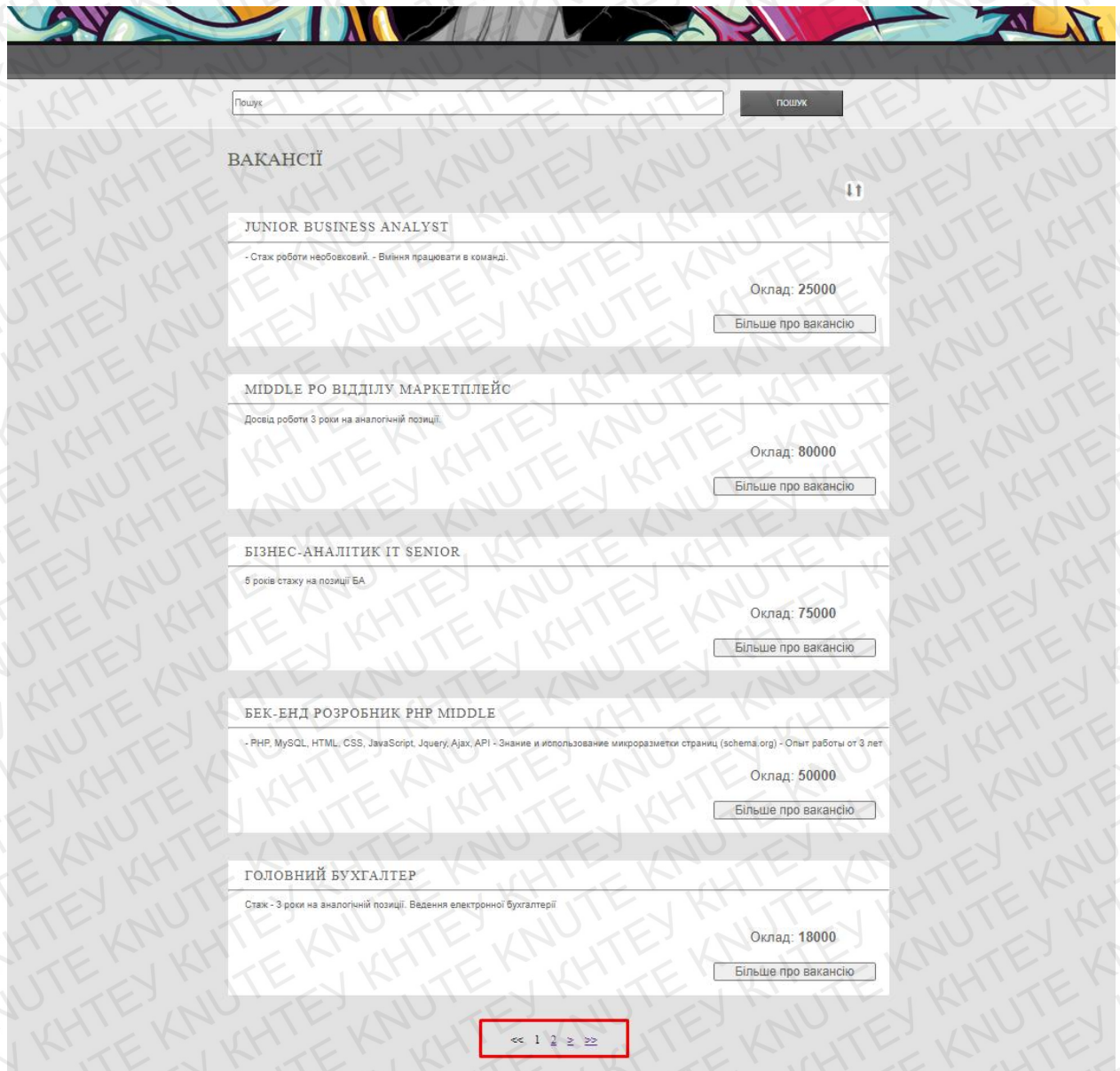


Рис. 3.5 – Блок переміщення по сторінкам(пагінація).

Натиснувши на назву компанії, на головній сторінці залишаться тільки ті вакансії, які розмістило обране підприємство.

В кінці сторінки розташований блок «пагінації». Користувач може переходити по сторінкам, при цьому є можливість перейти на останню існуючу сторінку, та повернутися на першу.

Далі перейдемо до інтерфейсу користувачів-компаній. Перша функціональна сторінка – реєстрація(рис. 3.6).

The screenshot shows the registration page of the ETW system. The header includes the ETW logo and the tagline 'Easy to work'. Navigation links are provided for 'ГОЛОВНА', 'ПРО НАС', 'НОВИНИ', 'КОНТАКТИ', 'УВІЙТИ', and 'РЕЄСТРАЦІЯ'. The main content area is titled 'РЕЄСТРАЦІЯ' and contains a form with the following fields:

- Введіть логін: (Login field)
- Введіть пароль: (Password field)
- Підтвердження пароля: (Confirm password field)
- Введіть назву компанії: (Company name field)
- Оберіть файл: (File selection button: 'Виберіть файл' / 'Файл не вибрано')
- Введіть контактний телефон: (Contact phone field)
- Введіть e-mail: (Email field)
- Введіть місто: (City field)
- Введіть адресу: (Address field)
- Додаткова інформація: (Additional information text area)

A 'ЗРЕЄСТРУВАТИСЬ' (Register) button is located at the bottom of the form.

Рис. 3.6 – Сторінка реєстрації облікового запису.

Перейшовши на сторінку реєстрації, користувач має заповнити всі поля, обрати логотип(у випадку, коли користувач не обере логотип – він заміниться на стандартний) та натиснути кнопку «Зареєструватись!».

Після реєстрації користувача автоматично потрапить на сторінку авторизації(рис. 3.7).

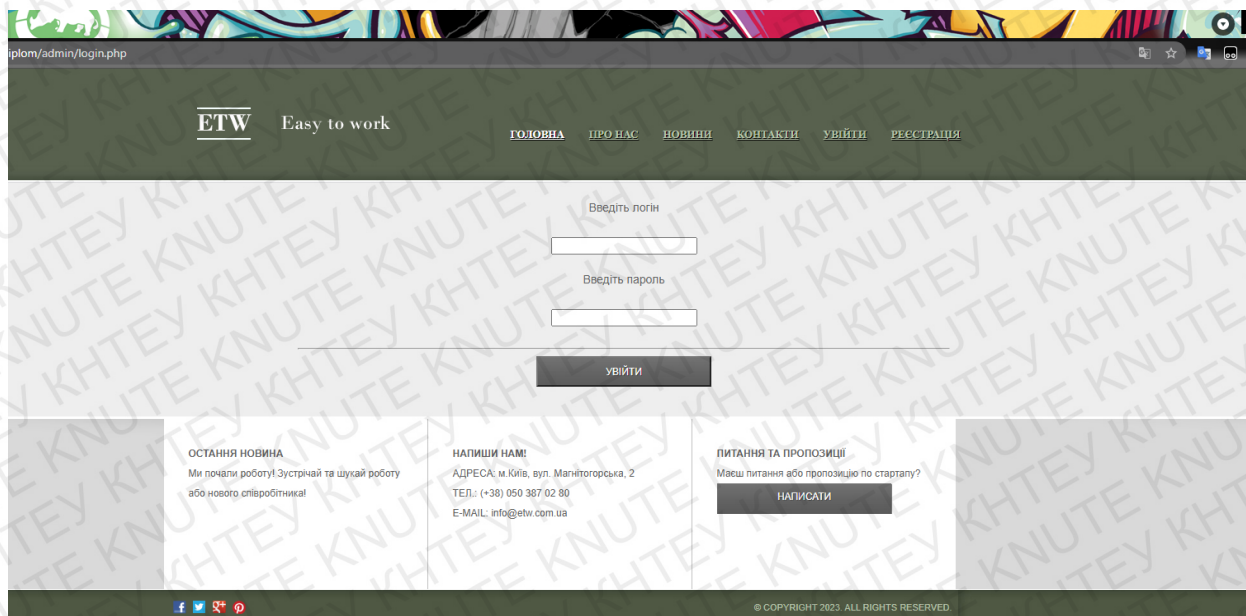


Рис. 3.7 – Сторінка авторизації.

Зареєстрований користувач може авторизуватись у системі та потрапити у власний кабінет(рис. 3.8).

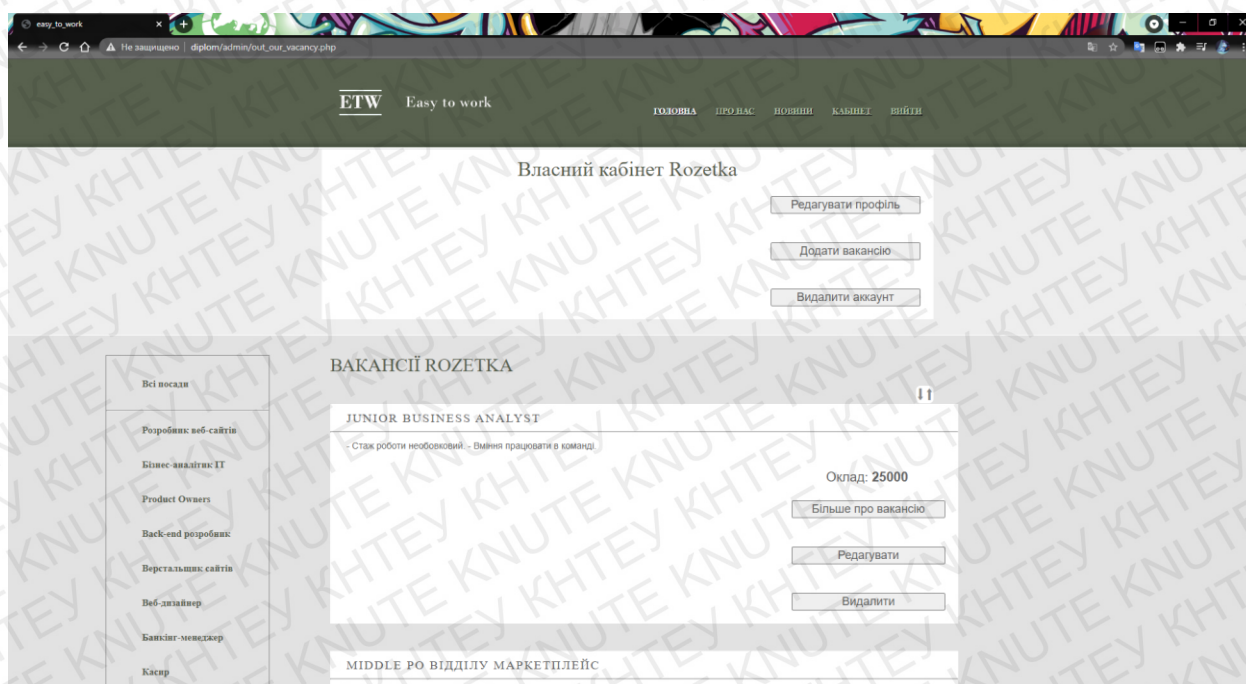


Рис. 3.8 – Інтерфейс власного кабінету компанії.

У власному кабінеті є функціональні кнопки редагування профілю, створення нової вакансії та видалення облікового запису.

Інтерфейс редагування профілю повторює інтерфейс реєстрації, але при переході на дану сторінку, всі поля заповнені актуальними на даний момент значеннями.

Процес додання вакансії також базується на вже існуючому інтерфейсі, який я використовував для реєстрації та авторизації(рис. 3.9). Такий підхід оптимізує загальну кількість стилізованих елементів, окрім цього всі сторінки виповнені в єдиному стилі.

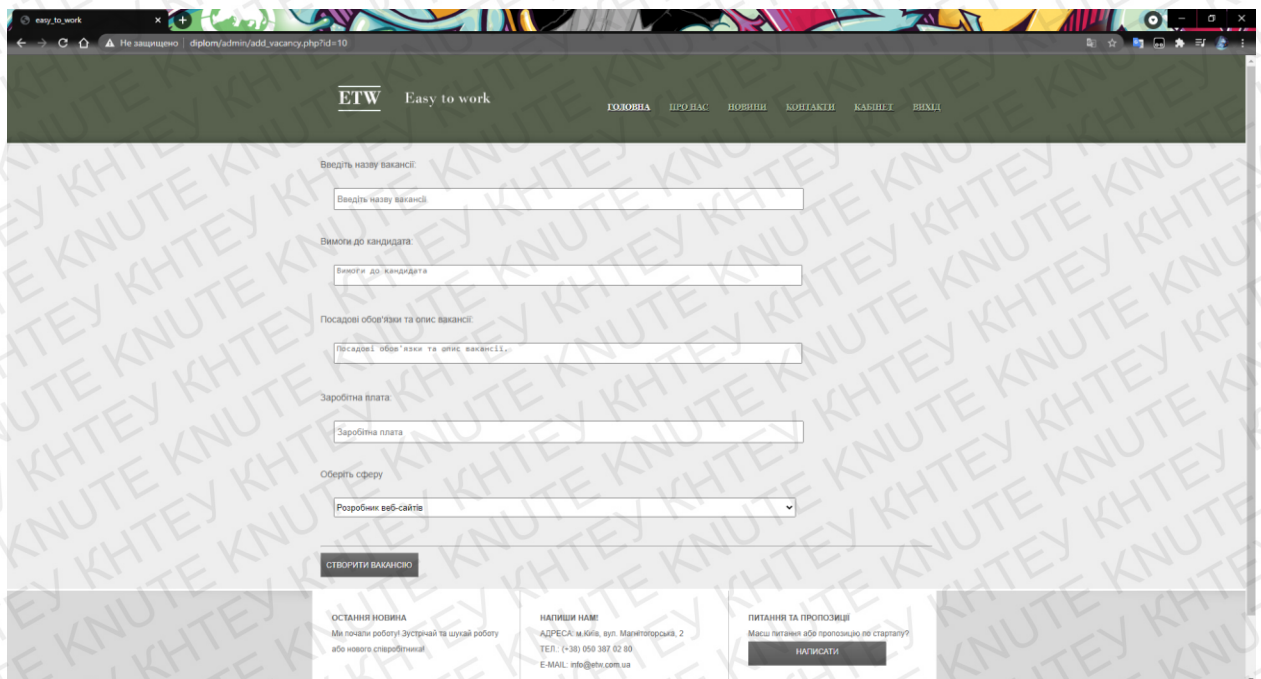
The image is a screenshot of a web browser displaying the 'Easy to work' website. The browser's address bar shows 'http://diprom/admin/add_vacancy.php?id=10'. The website has a dark green header with the 'ETW Easy to work' logo and navigation links: ГОЛОВНА, ПРО НАС, НОВИНИ, КОНТАКТИ, КАБІНЕТ, ВІЙДІ. The main content area is a light gray form for creating a job vacancy. It contains several input fields: 'Введіть назву вакансії', 'Введіть до кандидата', 'Посадові обов'язки та опис вакансії', 'Заробітна плата', and a dropdown menu for 'Оберіть сферу' with 'Розробник веб-сайтів' selected. A 'СТВОРИТИ ВАКАНСІЮ' button is at the bottom of the form. Below the form, there is a footer section with three columns: 'ОСТАННЯ НОВИНА' (last news), 'НАПІШИ НАМ!' (write to us) with contact details, and 'ПИТАННЯ ТА ПРОПОЗИЦІЇ' (questions and proposals) with a 'НАВІСІЛАТИ' button.

Рис. 3.9 – Сторінка для створення вакантної пропозиції.

У власному кабінеті кожна компанія-користувач може переглянути власні вакансії. Також існує функціонал фільтрування по сфері вакансії. Біля кожної вакансії виведені кнопки для маніпуляцій: розширений перегляд, редагування та видалення(рис. 3.10).

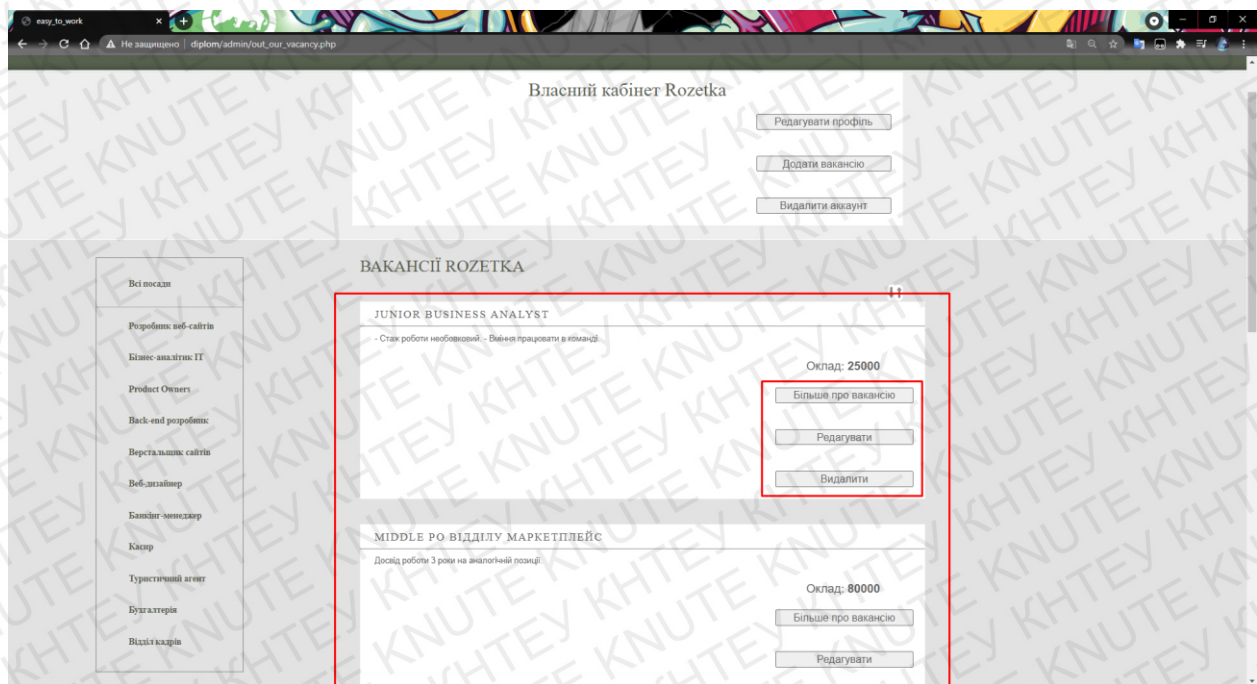


Рис. 3.10 – Список вакансій, які були створені з конкретного облікового запису.

Щоб детально переглянути інформацію про власну вакансію, необхідно натиснути на кнопку «Більше про вакансію» на відповідній картці. Відкриється сторінка з детальною інформацією про вакансію та функціональними кнопками для її видалення або редагування (рис. 3.11).

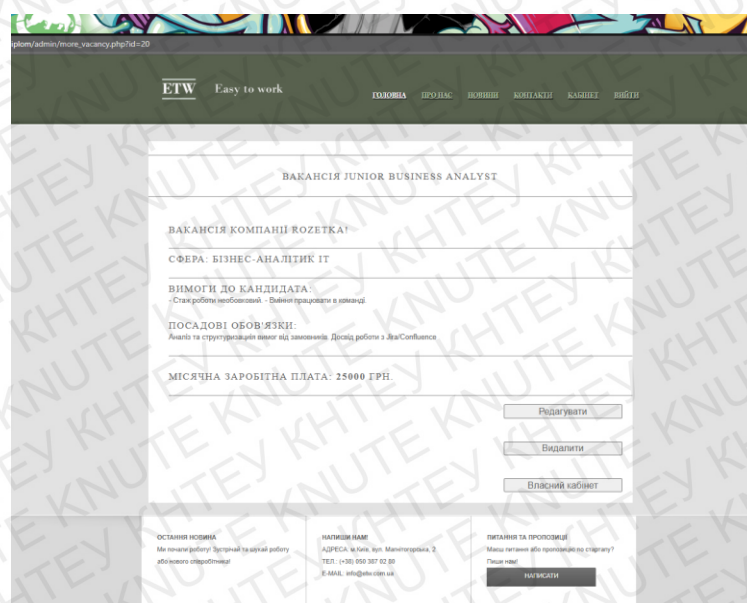


Рис. 3.11 – Детальний перегляд інформації про створену вакансію.

3.2. База даних та програмування сайту

При створенні концептуальної моделі веб-порталу, мною було обрано схему «MVC», та паттерн(шаблон проектування) «Функціональний інтерфейс».

Для зберігання інформації, яка надходить від користувачів, нам необхідно створити базу даних. Мною було вирішено використати додаток з відкритим кодом, написаним на мові PHP – phpMyAdmin. Дана утиліта представляє собою простий веб-інтерфейс для адміністрування СУБД MySQL.

Першим кроком є створення самої бази даних. Для цього необхідно авторизуватися в локальному середовищі «phpMyAdmin» та перейти до сторінки адміністрування БД. Щоб створити нову базу даних, достатньо натиснути на кнопку «Створити БД», після чого вказати її ім'я та тип – в нашому випадку це реляційна база даних, тому оберемо тип «порівняння», як вказано на рисунку 3.12.

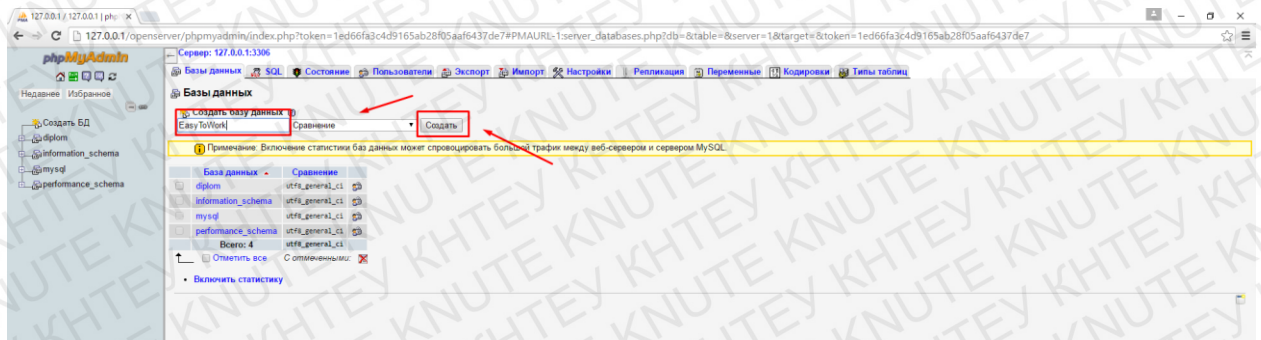


Рис. 3.12 – Створення бази даних в веб-додатку «phpMyAdmin».

Після ініціалізації БД можна переходити до створення таблиць. На мою думку, для початку роботи веб-порталу працевлаштування буде достатньо 3 пов'язані таблиці.

- Таблиця «Вакансії»
- Таблиця «Посада»

– Таблиця «Компанії»

Для кожної таблиці існує власний ідентифікатор, по якому вона індексується. Такий ідентифікатор унікальний в межах всієї таблиці, та є «ключем», по якому можна звертатися до кожного конкретного запису в таблиці.

Структуру таблиці «Вакансії» можна переглянути на рис. 3.13. На зображенні можна побачити, що поле «id» має специфічну позначку – ключ, тобто це поле є ідентифікатором.

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно
☐	1	id 	int(11)			Нет	Нет	AUTO_INCREMENT
☐	2	title	varchar(255)	utf8_general_ci		Нет	Нет	
☐	3	requirements	text	utf8_general_ci		Нет	Нет	
☐	4	description	text	utf8_general_ci		Нет	Нет	
☐	5	payment	int(50)			Нет	Нет	
☐	6	company_id	int(11)			Нет	Нет	
☐	7	position_id	int(11)			Нет	Нет	

Рис. 3.13 - Таблиця «Вакансії».

Необхідно відмітити, що в даній таблиці поля «company_id» і «position_id» - являють собою «іноземний ключ» (foreign key), за допомогою якого реалізовані зв'язки з іншими таблицями.

Щодо наступних двох таблиць – їх створення повністю повторює вже продемонстрований алгоритм, який передбачає визначення кількості полів, їх типів та властивостей. Фінальний результат ініціалізації БД для створення веб-порталу продемонстровано на рисунку 3.14.



Рис. 3.14 – Структура бази даних веб-ресурсу працевлаштування.

Наступним масштабним кроком до реалізації веб-порталу є ініціалізація проекту в середовищі програмування «JetBrains phpStorm». Для початку ми створюємо репозиторій, в якому будуть розташовані всі функціональні та інформаційні блоки майбутнього ресурсу, файл зі стилями і сховище для файлів, які будуть завантажувати користувачі.

Далі налаштовуємо з'єднання із базою даних. Після успішного з'єднання середовище програмування відобразить всі доступні для проекту таблиці, які ми будемо використовувати під час написання програмного коду веб-ресурсу(рис. 3.15).

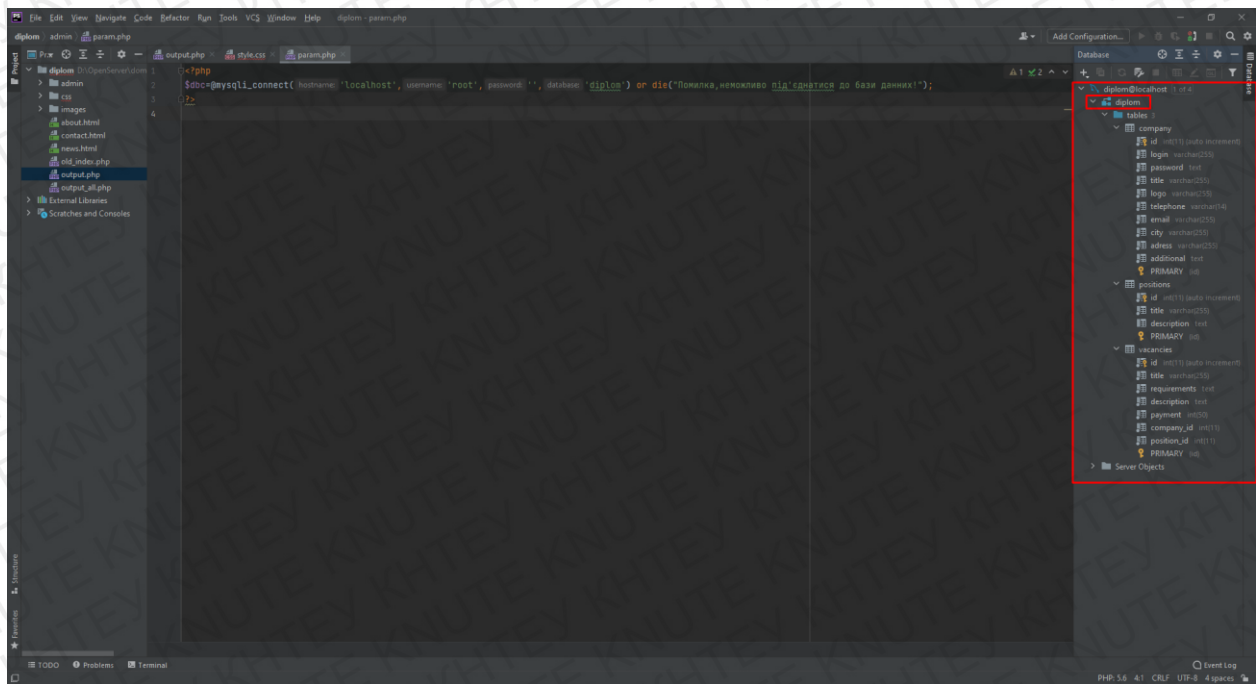


Рис. 3.15 – Інтеграція проекту з базою даних.

Почнемо із сторінки реєстрації, адже нам, насамперед, необхідно мати можливість потрапити до власного кабінету, щоб почати створювати вакансії.

На сторінці реєстрації ми виводимо форму, яка методом «POST» може передавати значення, які ввів користувач. Якщо користувач натисне кнопку, спрацює обробник події, який передасть всі введені дані в вигляді об'єкту. Далі ми можемо сформулювати запит «Insert» в базу даних, за допомогою якого запишемо введені дані в таблицю «Компанії».

Необхідно зауважити, що перед виконанням запиту нам необхідно перевірити унікальність введеного логіну. Таку перевірку реалізуємо за допомогою умовного оператора «if», який буде перевіряти результат відповіді СУБД на запит «SELECT».

Окрім цього, нам необхідно забезпечити захист пароля користувача. Тому, під час формування запиту, який записує дані з форми в БД, використаємо вбудовану функцію хешування «SHA1» для шифрування паролю.

Далі, необхідно реалізувати функціонал авторизації на сайті. Користувач вводить логін та пароль для входу в обліковий запис на сторінці. Після того, як він натисне кнопку «Авторизуватись» - програмний код відправить запит на пошук користувача з таким логіном, і якщо такий користувач існує в системі – порівняє «хеш» введеного пароля з тим, що знаходиться в базі даних(рис. 3.16). Якщо паролі ідентичні – користувач увійде у власний кабінет.

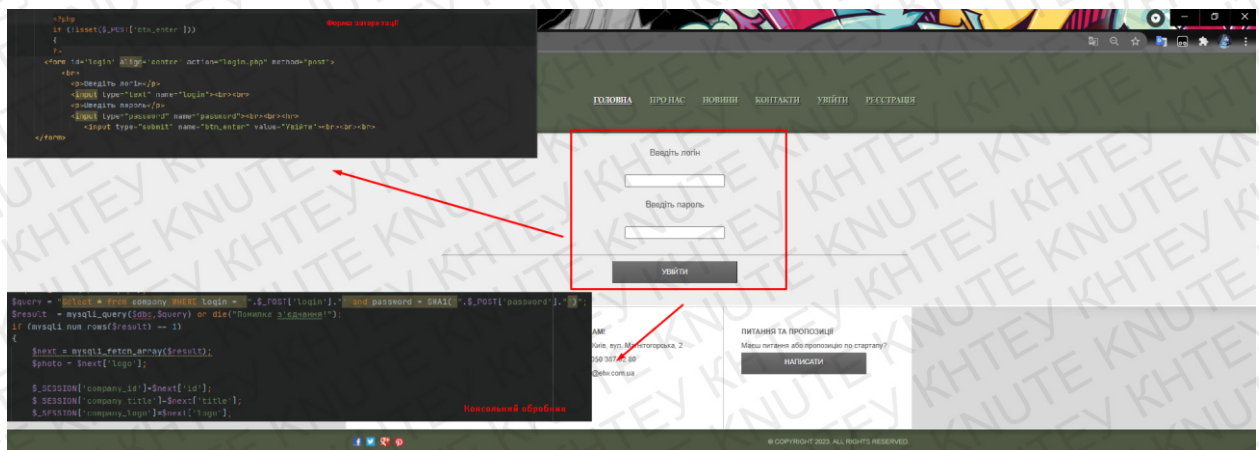


Рис. 3.16 – Створення та програмування інтерфейсу авторизації.

Якщо авторизація пройде успішно, консольний додаток розпочне «сесію», яка дозволить перемішуватися по сторінками сайту без повторної авторизації. При виході із сайту, сесія буде видалена.

Тепер перейдемо до програмування інтерфейсу власного кабінету. Насамперед, основна задача – реалізувати можливість створення вакансій. Для цього скористаємося формою, з необхідними полями, звідки зможемо отримувати введену користувачем інформацію та вносити її в базу даних за допомогою запиту «INSERT INTO».

При створенні вакансій, мені вдалося автоматично надсилати ідентифікатор компанії в таблицю «Вакансій», адже «сесія» може зберігати деяку кількість значень в масиві. Заповнення поля «position_id» відбувається

за допомогою випадаючого списку «select»(рис. 3.17). Цей список заповнюється значеннями із таблиці «positions», за допомогою запиту «SELECT». При виборі необхідного значення, формою буде передаватися ідентифікатор, для запису в БД.

The screenshot shows a web form for posting a job on the ETW website. The form has the following fields:

- Введіть назву вакансії: (Text input)
- Вимоги до кандидата: (Text input)
- Посадові обов'язки та опис вакансії: (Text input)
- Заробітна плата: (Text input)
- Оберіть сферу: (Dropdown menu)

The dropdown menu for 'Оберіть сферу' is open, showing a list of job categories. The selected category is 'Розробник веб-сайтів'. Other categories include:

- Бізнес-аналітик IT
- Product Owners
- Back-end розробник
- Верстальщик сайтів
- Веб-дизайнер
- Банкінг-менеджер
- Касир
- Туристичний агент
- Бухгалтерія
- Відділ кадрів

The website header includes the ETW logo and the text 'Easy to work'. Navigation links are: ГОЛОВНА, ПРО НАС, НОВИНИ, КОНТАКТИ, КАБІNET, ВИХІД.

The footer includes contact information: ТЕЛ.: (+38) 050 387 02 80, E-MAIL: info@etw.com.ua, and a button 'НАПИСАТИ'. Social media icons for Facebook, Twitter, and YouTube are also present.

Рис. 3.17 – Випадаючий список «select».

Наступний крок – редагування вакансій та облікових записів. Загалом, інтерфейс редагування має ідентичну структуру зі створенням. Але є декілька важливих відмінностей:

По-перше, під час редагування, ми заповнюємо всі поля із бази даних, за допомогою запиту «SELECT» (рис. 3.18). Таким чином, користувач наглядно

бачить те, що збирається редагувати. При цьому, поля в яких не було змін – перезапишуться в БД у тому ж вигляді.

По-друге, тип запити змінюється на «UPDATE» - цей запит перезаписує значення у БД.

Рис. 3.18 – Автоматичне заповнення полів на сторінці редагування.

Далі необхідно реалізувати можливість видалення вакансій з таблиці. Також, я вирішив надати можливість видаляти обліковий запис.

Натиснувши на кнопку «Видалити», відкриється сторінка з підтвердженням. Якщо він підтвердить видалення, система відправить запит типу «DELETE», який видалить відповідний запит в таблиці.

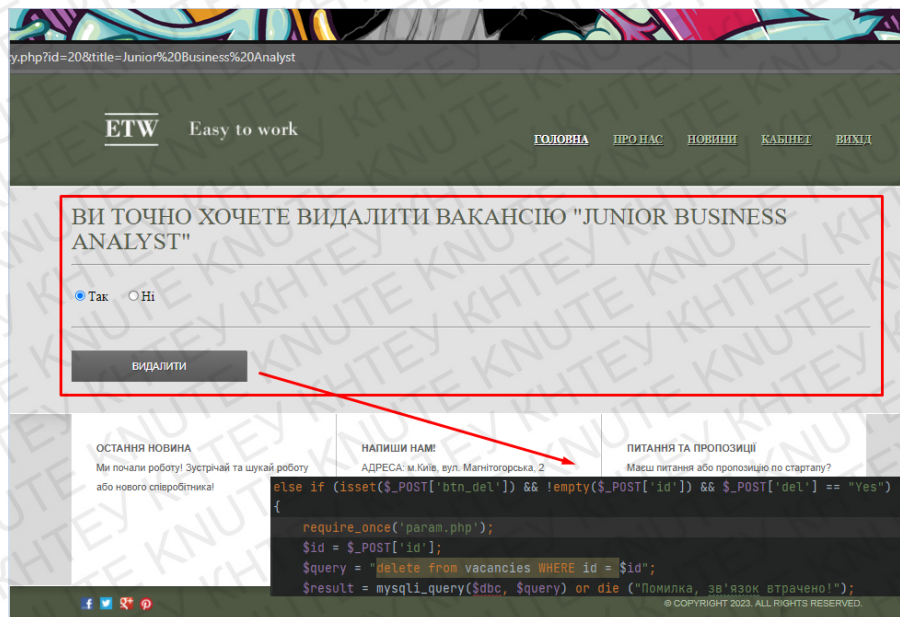


Рис. 3.19 - Видалення вакансій.

У випадку, коли користувач хоче видалити свій обліковий запис – нам також необхідно видаляти всі вакансії, які були створені цією компанією.

Далі перейдемо до програмування інтерфейсу перегляду вакансій. В моєму випадку – це фільтрування, пошук та сортування.

Пошук було реалізовано за допомогою форми, в якій розміщене текстове поле та кнопка. Введений пошуковий запис додається до «SELECT» запиту в базу даних. Пошук не чутливий до регістру, окрім цього шукає збіг після кожного символу в рядку.

Фільтрація була реалізована за допомогою гіперпосилань, які передають методом «GET» ідентифікатор компанії. Отримані дані додаються до запиту в БД, після чого сторінка оновлюється, а вивід вакансій обмежений однією конкретною компанією.

Сортування – вже існуючий функціонал в СУБД. В запит «SELECT» можна включити сортування по спаданню або зростанню («ASC» і «DESC»). Таким чином, нам необхідно запрограмувати одну кнопку, яка при натисканні буде змінювати запит.

На даному етапі завершується процес програмування веб-порталу

працевлаштування. Основні вимоги до функціонування інтерфейсу виконані, в відповідності до функціональних потреб потенціальних користувачів.

3.3. Оцінка отриманих результатів проектування

Процес реалізації веб-ресурсу об'єднав процеси проектування інтерфейсу та структури бази даних, створення інтерфейсу та дизайн функціональних блоків, і, насамперед, написання програмного коду консольного обробника. В результаті ми отримали готовий робочий продукт, який відповідає паттерну «Functional design».

Для користувачів-здобувачів, було реалізовано простий та зручний інтерфейс, який дозволяє переглядати вакантні пропозиції різних компаній.

Для даної групи користувачів мною було впроваджено:

- Сортування за спаданням або зростанням.
- Фільтрація по зареєстрованим компаніям.
- Пошук вакансій по сайту

Для другої групи користувачів – компаній, було створено можливість зареєструватися на порталі, що передбачає наявність облікового запису. Кожен користувач з цієї групи має доступ до власного кабінету, де він має доступ до всіх своїх вакансій, може редагувати, видаляти і, насамперед, створювати їх. Окрім цього, він може вносити зміни в свій обліковий запис, або навіть видалити його.

На цьому етапі створення веб-порталу працевлаштування можна вважати виконаним в повному обсязі.

ВИСНОВКИ ТА РЕЗУЛЬТАТИ

Слід відзначити, що тема пошуку роботи та «ханті́нгу» співробітників залишається актуальною, незважаючи на плин часу та ступінь інформатизації суспільства. Але, звичайно, сучасні технології можуть значно полегшити та прискорити процеси пошуку роботи та співробітників.

Для реалізації проекту було проведено змістовний аналіз сучасних тенденцій, методологій та архітектури, на що варто істотно опиратися при розробці веб-додатків. Визначено основні не функціональні вимоги до веб-ресурсу, які повинні забезпечити надійність та безпосередню доступність ресурсу для користувача в будь-який момент часу.

Проведено змістовне порівняння сучасних засобів для проектування та створення веб-застосовань, на основі яких визначено максимально доцільний спосіб, який зміг би повністю відповідати поставленим функціональним та не функціональним вимогам.

Другий розділ присвячений формалізації функціональних вимог до функціонального наповнення веб-порталу, на основі аналізу схожих програмних веб-застосунків, потенційної аудиторії та концепції MVP (Minimum viable product). В ході моделювання було визначено необхідні інтерфейсові модулі для кожної групи користувачів, на які можна спиратися при верстці, програмуванні функціональних блоків та реалізації дизайну веб-порталу.

В третьому розділі було змістовно описано процес верстки сайту та застосування стилів до всіх груп елементів. Також було описано процес програмування основних функціональних модулів, які направлені на організацію простої та зручної взаємодії користувачів з веб-ресурсом, а саме:

- Модуль реєстрації.
- Модуль авторизації.

- Редагування розміщених вакансій та облікових записів.
- Створення вакантних пропозицій.
- Видалення неактуальних вакансій.
- Регістронезалежний пошук вакансій на сайті.
- Фільтрація та сортування вакансій за певними ознаками.

В ході виконання випускного кваліфікаційного проекту були формалізовані теоретичні відомості про сучасні тенденції розробки Web-ресурсів, на основі яких спроектовано та програмно реалізовано сайт працевлаштування.

Таким чином, ми отримали модульний веб-ресурс, що володіє мінімальними, але достатніми для задоволення перших споживачів функціями, і який в майбутньому може модернізуватися великим переліком модулів та бібліотек, якщо цього буде потребувати кінцевий користувач.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dan Olsen - The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback, 336 сторінок, видавництво: Wiley, рік випуску 2015.
2. Programming PHP: Creating Dynamic Web Pages, 544 сторінки, видавництво: O'Reilly Media, рік випуску 2020.
3. Робін Ніксон - Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5, 768 сторінок, видавництво: Питер, рік випуску 2019.
4. В.І. Гайдаржи, І.В. Ізварін - Бази даних в інформаційних системах, 418 сторінок, видавництво: Університет "Україна", рік випуску 2018.
5. Джордж Шлосснейгл - Профессиональное программирование на PHP, 624 сторінки, видавництво: Вильямс, рік випуску: 2006, тираж 3000.
6. Опалєв М., Вакуленко О. Закономірності формування сучасного тренду веб-дизайну – «плаский» дизайн // Вісник Львівської національної академії мистецтв. Львів: Вид-во ЛНАМ, 2016.
7. Лаура Томсон та Люк Веллінг - Разработка WEB-приложений на PHP и MySQL, 672 сторінки, видавництво DiaSoft, рік випуску 2003.
8. Берон Шварц, Петр Зайцев, Вадим Ткаченко - MySQL по максимуму. 3-е издание, 864 сторінки, видавництво: Питер, рік випуску 2018.
9. Kevin McArthur - PHP: Patterns, Frameworks, Testing and More, Print length 373 pages, Publisher Apress, Publication date April 1, 2008.
10. Эрик А. Мейер - CSS - каскадные таблицы стилей. Подробное руководство, 576 сторінок, видавництво: Символ-Плюс, рік випуску 2008.
11. Документація по середовищу програмування phpStorm - <https://www.jetbrains.com/ru-ru/clion/features/code-documentation.html>

12. Документація по налагодженню середовища phpStorm - <https://www.jetbrains.com/ru-ru/phpstorm/documentation/debugging/>
13. Документація PHP 4 і PHP 5 - <https://www.php.net/docs.php>
14. Користувачка документація по phpMyAdmin - <https://phpmyadmin.ru/doc/>
15. Інструкція для початку роботи з веб-додатком Figma - <https://help.figma.com/hc/en-us>
16. Довідник по HTML - <http://htmlbook.ru/html>
17. Документація по утиліті Open Server - <https://ospanel.io/docs/>
18. Довідник з MySQL - <http://www.mysql.ru/docs/man/>