МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«РОЗРОБКА ВЕБ-САЙТУ ФРИЛАНС-БІРЖІ»**

Виконав: студент 2 курсу, групи 8.1223
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)
освітньої програми Комп'ютерні науки
(назва освітньої програми)
І. В. Шведов
(ініціали та прізвище)
Керівник завідувачка кафедри комп'ютерних наук, д.т.н., професор Шило Г.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент доцент кафедри програмної інженерії, к.ф.-м.н. Кривохата А.Г.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти магістр

Спеціальність 122 Комп'ютерні науки
(шифр і назва)

Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Шило Г.М.
(підпис)

" 5 " 03 2024 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Шведову Іллі Владиславовичу
(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка веб-сайту фриланс-біржі

керівник роботи Шило Галина Миколаївна, д.т.н., професор
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 1 » березня 2024 року № 385-с

2. Строк подання студентом роботи 27.11.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Розробка веб-сайту фриланс-біржі.

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

6. Консультанти розділів роботи

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

7. Дата видачі завдання 1.03.2024 р.

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	20.03.2024	
2.	Збір вихідних даних.	10.04.2024	
3.	Обробка методичних та теоретичних	26.04.2024	
	джерел.		
4.	Розробка першого та другого розділу.	14.05.2024	
5.	Розробка третього розділу.	12.06.2024	
6.	Оформлення та нормоконтроль	27.11.2024	
	кваліфікаційної роботи магістра.		
7.	Захист кваліфікаційної роботи.	12.12.2024	

Студент _____ І.В. Шведов
(підпис) (ініціали та прізвище)

Керівник роботи _____ Г.М. Шило
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г. Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка веб-сайту фриланс-біржі»: 62 с., 57 рис., 20 джерел.

APACHE, API, PHP, SQL, ХАМРР, ВЕБ-САЙТ, ФРИЛАНС-БІРЖА.

Об’єкт дослідження – існуючі аналоги веб-сайтів бірж та сервісів оголошень.

Мета роботи: розробити свій аналог веб-сайту фриланс-біржі.

Метод дослідження – аналітичний, порівняльний.

Фриланс-біржа – платформа для пошуку як тимчасової підробітки, так і постійного місця роботи найрізноманітнішими категоріями людей. Від вигулу домашніх улюбленців та домашньої випічки на продаж до офферів інтернаціональних корпорацій. У зв’язку з трендами на цифровізацію бізнесу, створення та споживання онлайн продукції, такі сайти стрімко знаходять свою цільову аудиторію. У кваліфікаційній роботі магістра розглянуто існуючі платформи фрилансу та на основі порівняння створено свій аналог фриланс-біржі.

SUMMARY

Master’s Qualifying Theses «Development of the Freelance Marketplace Website»: 62 pages, 57 figures, 20 references.

APACHE, API, FREELANCE MARKETPLACE, PHP, SQL, WEBSITE, XAMPP.

Object of the study – existing analogues of exchange websites and ad services.

Aim of the study: develop your own counterpart to a freelance marketplace website.

Method of research – analytical, comparative.

The freelance marketplace is a platform for finding both temporary part-time work and permanent employment by the most diverse categories of people. From walking pets and home baking for sale to offers from international corporations. In connection with the trends in digitalization of business, creation and consumption of online products, such sites are rapidly finding their target audience. The master’s qualification work examined the existing freelance platforms and based on the comparison, created its analogue of the freelance marketplace.

ЗМІСТ

Завдання на кваліфікаційну роботу……..2
Реферат 4
Summary 5
Вступ..... 7
1 Аналіз предметної області та вимоги..... 8
1.1 Постановка задачі 8
1.2 Цілі створення, призначення 8
1.3 Теоретичний аналіз аналогів..... 9
1.3.1 Kabanchik.ua 9
1.3.2 Freelancehunt.com..... 12
1.3.3 Freelance.ua 14
1.4 Вимоги..... 17
2 Опис засобів для розробки і проектування 18
2.1 Програмні засоби 18
2.2 Середовище для розробки..... 19
2.3 Налаштування Xampp..... 20
2.4 Діаграма прецендентів..... 25
2.5 Діаграма послідовності..... 27
2.6 Діаграма компонентів..... 28
2.7 База даних 29
3 Програмна реалізація..... 31
3.1 Розробка веб-сайту фриланс-біржі..... 31
3.2 Опис функціоналу та взаємодія з розробленим сайтом..... 50
3.2.1 В ролі замовника..... 50
3.2.2 В ролі фрилансера..... 55
Висновки 60
Перелік посилань..... 61

ВСТУП

Інтернет вже давно перестав бути засобом зв’язку спецслужб на випадок ядерної війни. Протягом близько 60 років свого існування він перетворився на доступну мережу для кожного, хто має хоч якийсь засіб зв’язку. Спілкування, навчання, розваги, новини, ідеї – все зараз можна знайти в інтернеті.

Засновник корпорації «Microsoft», Білл Гейтс на одній з своїх конференцій став автором пророцької крилатої фрази: «Якщо вашої компанії немає в інтернеті – вашого бізнесу не існує». І як показує час, бізнес або був вимушений з’явитися в мережі, або не витримати конкуренції тих, хто зміг адаптуватися до цієї вимоги нового часу [1, 2].

За роботодавцями підтягнулися і робітники. А під впливом останніх 5 років, віддалена робота взагалі стала більше нормою, ніж було нормою до цього фізичне відвідування робочого місця в умовному офісі.

Розвиток відчули й індивідуальні підприємці, отримавши змогу рекламувати чи шукати роботу не по оголошенням, наклеєним на умовній зупинці транспорту, а на спеціалізованих сайтах, як локальних так і інтернаціональних. Зараз, на таких платформах з легкістю можна знайти собі роботу, так і запропонувати, в залежності від свого фаху, можливостей і звісно крапельки удачі.

У кваліфікаційній роботі розглянуто основні вимоги до веб-сайтів фриланс-бірж та розроблений свій аналог фриланс-біржі з використанням SQL, PHP, HTML, JS і дотриманням цих вимог. Розроблений сайт має мати основний функціонал, необхідний користувачу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГИ

1.1 Постановка задачі

Пошук роботи онлайн вже давно увійшов в повсякденне життя, стрімко посунувши інші, «більш традиційні» методи з своїх позицій. Зараз близько кожен 3 найм відбувається або спочатку через онлайн співбесіду, як перший етап, або взагалі і сама робота не потребує фізичного відвідування офісу. Для зустрічі потенційного роботодавця і робітника необхідно організувати певну платформу, де робітник може вказати своє резюме або запропонувати послуги, а роботодавець запропонувати роботу або підібрати собі робітника на свій розсуд у зручному форматі.

1.2 Цілі створення, призначення

Експлуатаційне призначення – у результаті реалізації буде створено веб-сайт фриланс-біржі. З використанням такого стеку технологій як: HTML, PHP, SQL. Ця робота може стати або прикладом для розробників аналогічних сайтів, або стартовим проектом для коректив та доопрацювання вже в комерційний продукт.

Функціональне призначення – згідно поставленої задачі у результаті аналізу результатом кваліфікаційної роботи має бути розроблений веб-сайт фриланс-біржі.

Мета створення – розробити веб-сайт фриланс-біржі. У процесі розробки отримати нові навички з використання HTML, PHP, SQL.

1.3 Теоретичний аналіз аналогів

На сьогодні існує безліч як інтернаціональних так і локальних фриланс-бірж, сайтів оголошень і навіть маркетплейсів, що потроху створюють у себе категорію «Пошук роботи». Дійсно, у цих платформ доволі спільний функціонал, що дозволяє комбінувати наприклад ту ж саму біржу пошуку роботи з маркетплейсом безкоштовних оголошень.

Не будемо йти дуже далеко та розглянемо українські сайти, як Kabanchik.ua [3], Freelancehunt.com [4], Freelance.ua [5]. Порівняємо обрані сайти.

1.3.1 Kabanchik.ua

Ця платформа позиціонує себе, як онлайн сервіс пошуку приватних фахівців для вирішення побутових та бізнес завдань. Майданчик об’єднує замовників послуг, яким необхідно виконати яку-небудь роботу, та компетентних фахівців, які шукають підробіток або додатковий заробіток [3].

Сервіс був запущений у вересні 2012 року двома молодими підприємцями Романом Кірігетовим і Олександром Юр’євим. На даний час сервіс працює у всіх великих містах України, а також в Казахстані. З 2015 року проект був куплений Prom.ua і на сьогодні входить до складу групи компаній EVO [3].

На головній сторінці можна побачити сортування послуг, пошук за ключовою фразою, реєстрацію, вхід, зміну мови сайту, певні пояснення щодо роботи з сайтом для користувача, інформацію про розробників, цілі, контакти, допомогу.

Доступний як мобільний додаток, так і традиційно, за посиланням на сайт (див. рис. 1.1).

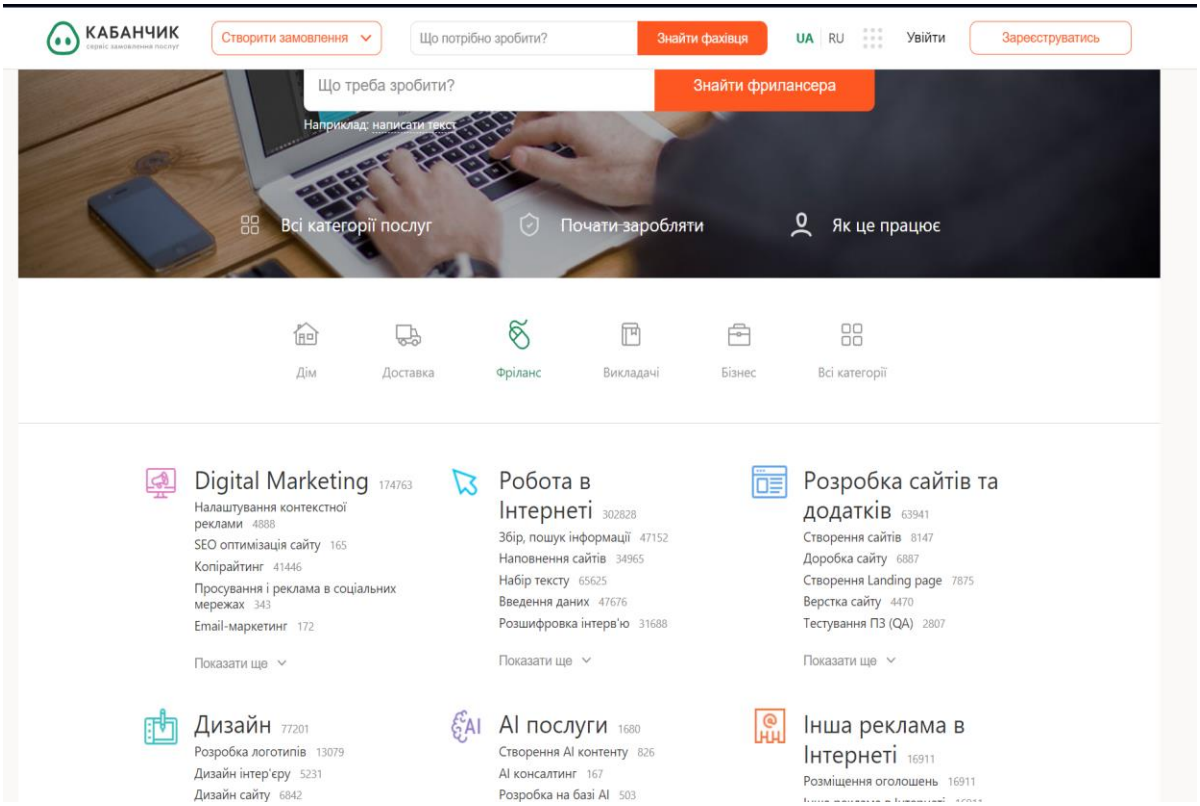


Рисунок 1.1 – Головна сторінка веб-версії платформи Kabanchik.ua [3]

На сторінці оголошень можна побачити підборку популярних виконавців, що вони вміють і пропонують, відгуки, статистику, аватарки, можливість створити замовлення, сортування (див. рис. 1.2).

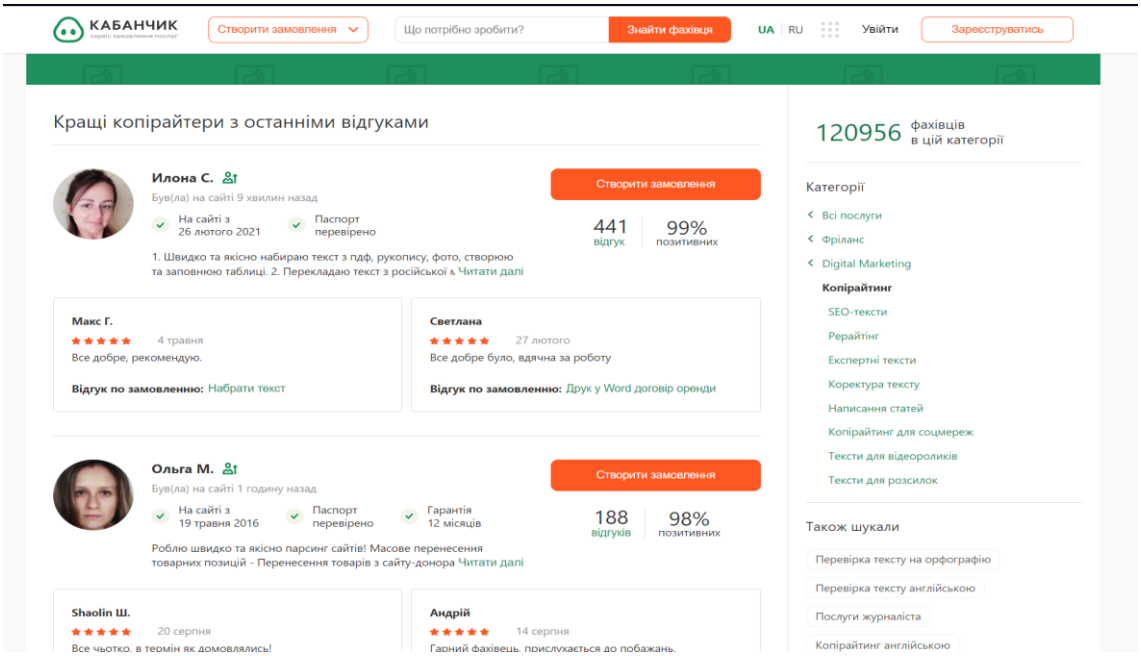


Рисунок 1.2 – Сторінка оголошень веб-версії платформи Kabanchik.ua [3]

На сторінці з детальним описом оголошення можна побачити аватарку користувача, локацію, відгуки, сам детальний опис сфери послуг що він пропонує, категорії роботи, запропонувати роботу, приблизний прайс (див. рис. 1.3).

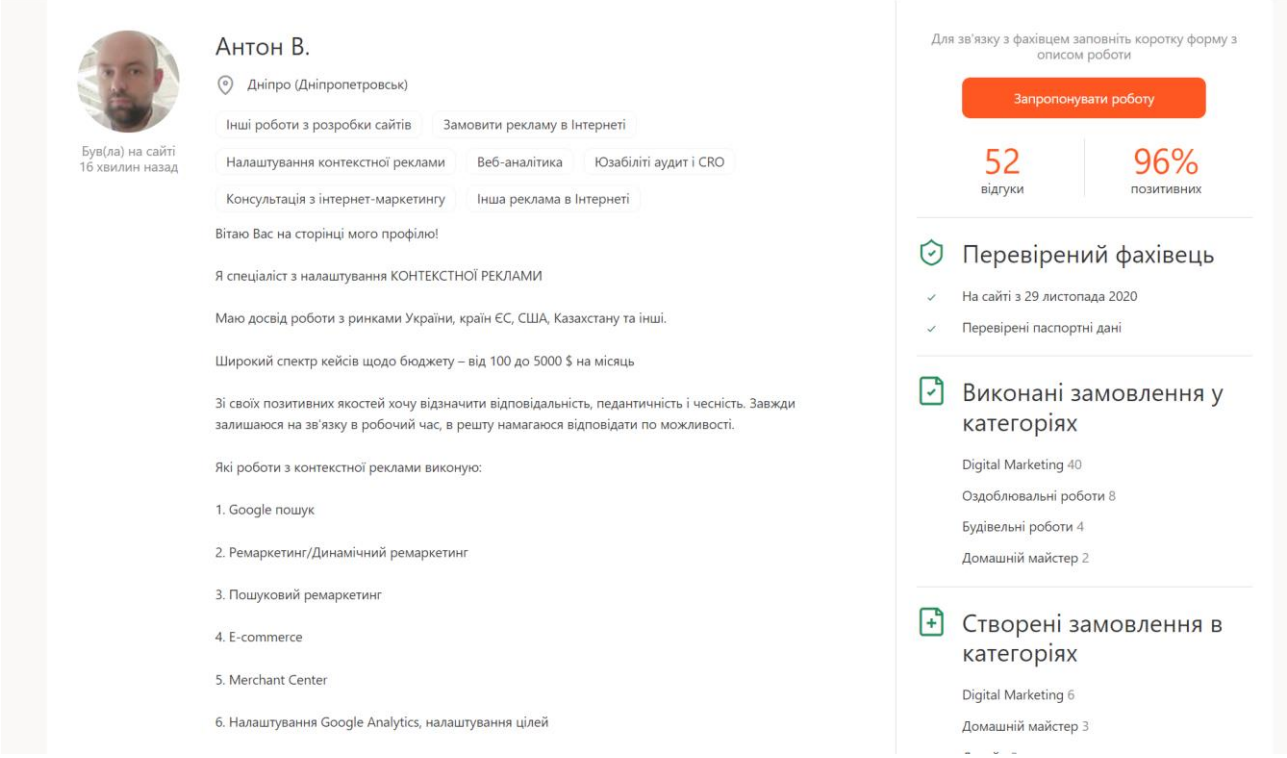


Рисунок 1.3 – Сторінка з детальним описом оголошення платформи Kabanchik.ua [3]

Щодо переваг сайту можна віднести просту реєстрацію, зрозумілий інтерфейс, можливість поділу на категорії, наявність партнерів, можливість створювати безкоштовні оголошення, роботу сайту більше 10 років а отже більш-менш стійкий онлайн.

Відносно недоліків, я б назвав все ж різноманітність сектору послуг, у яких можна згубитися, а також платформа бере комісію з кожного замовлення, якщо воно оформлене через неї а не просто подивилися ваші контакти і подзвонили домовилися. Хоча з іншого боку тут дійсно можна знайти все.

1.3.2 Freelancehunt.com

Більш спеціалізована платформа для розміщення замовлення та пошуку виконавця онлайн роботи. Дещо схоже на HH, але на мій погляд менш інформативна головна сторінка (див рис. 1.4). Є лише веб-версія.

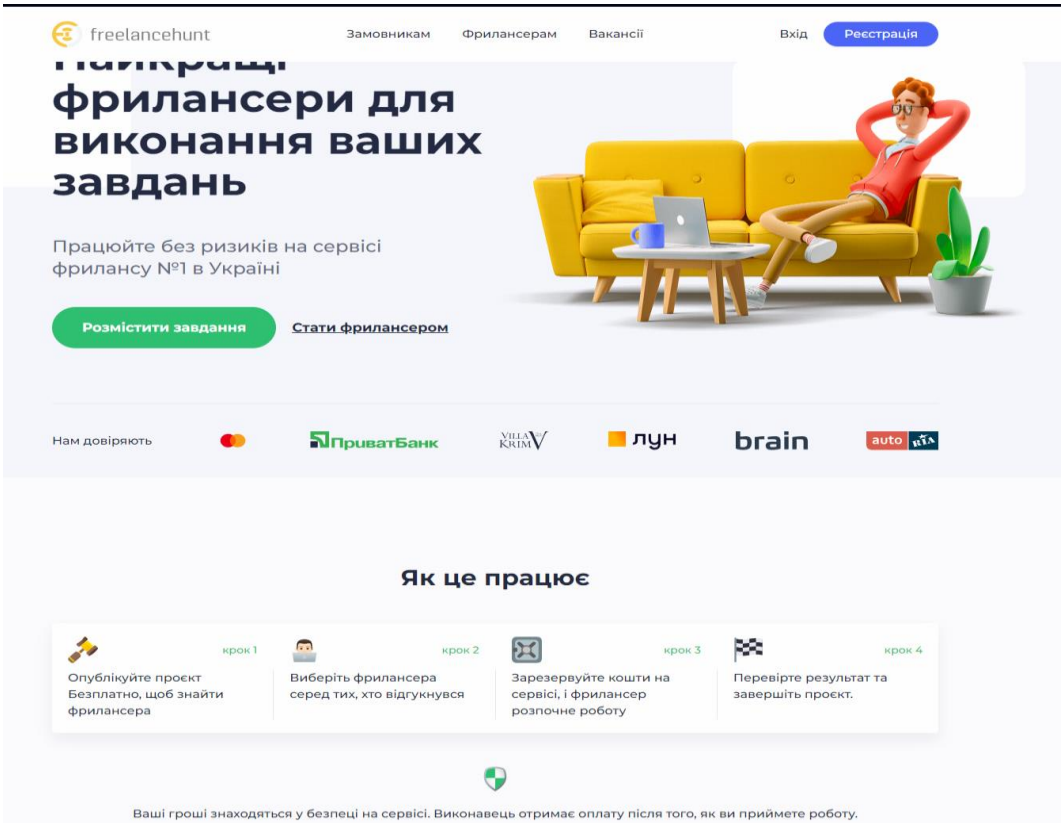


Рисунок 1.4 – Головна сторінка веб-версії платформи Freelancehunt.com [4]

Сьогодні Freelancehunt – біржа фрилансу топ-5 в Україні та активно розвивається у країнах Східної Європи. У 2021 році на сервісі було опубліковано 200 тис. завдань і зареєстровано мільйонного користувача [4].

На сторінці оголошень можна знайти підборку популярних(або преміум) виконавців, сортування, звідки вони, яку роботу виконують, рейтинг, приклади виконаної роботи (див. рис. 1.5).

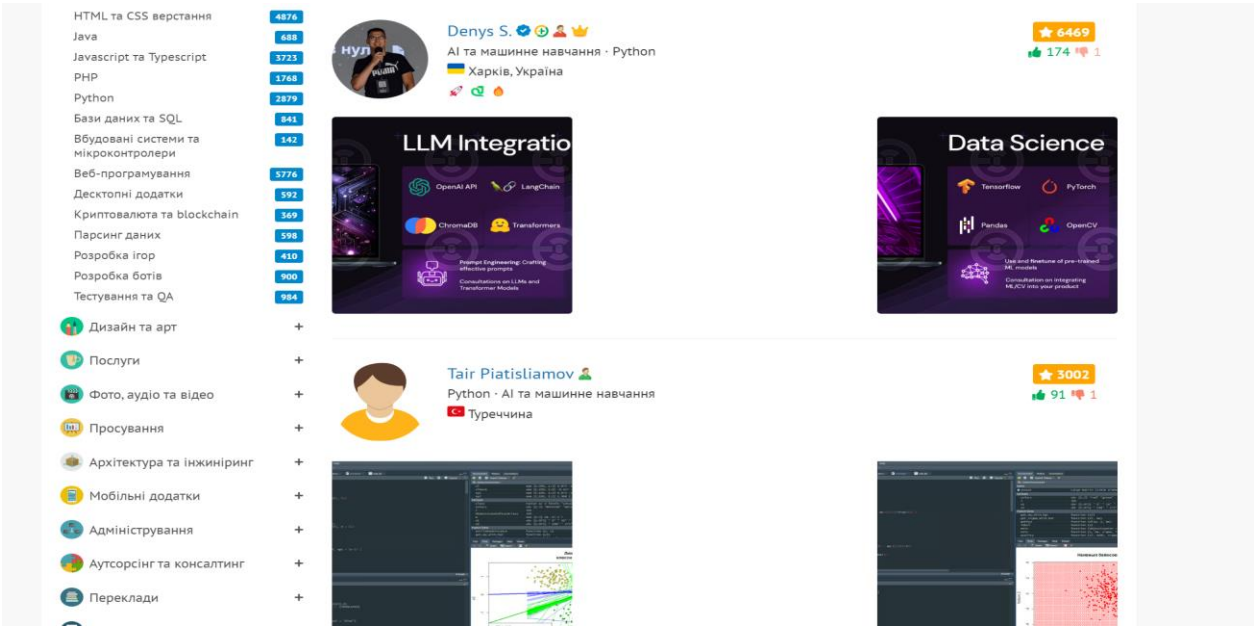


Рисунок 1.5 – Сторінка оголошень веб-версії платформи Freelancehunt.com [4]

Сторінка з детальним описом оголошення демонструє аватарку користувача, детальний опис самого оголошення, рейтинг, відгуки, мітки з категоріями пропонованого спектру послуг, перегляди цього оголошення, локацію користувача (див.рис.1.6).

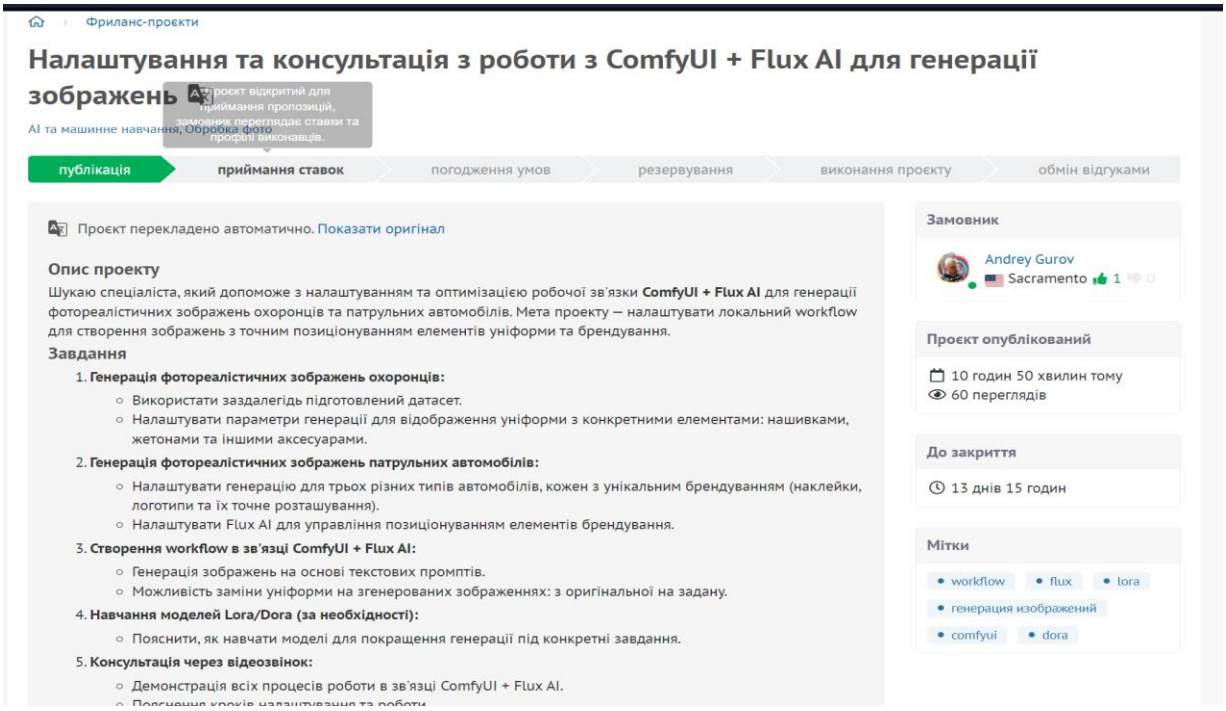


Рисунок 1.6 – Сторінка з детальним описом оголошення платформи Freelancehunt.com [4]

Щодо переваг, є реєстрація, авторизація, розміщення оголошень, сортування, відгуки, рейтинг, наявність партнерів, можливість розмістити резюме та у профілі розмістити приклади виконаних робіт.

Стосовно недоліків – наявність платного преміуму, комісію за виконане замовлення через сайт, відсутність мобільного застосунку, лише 4 роки роботи біржі.

1.3.3 Freelance.ua

Дуже схожа на Freelancehunt [4], але оформлена у більш спокійних тонах та напевне з більш зрозумілим дизайном (див. рис. 1.7), біржа Freelance.ua [5] працює вже 15 років на ринку. Мобільного додатку немає.

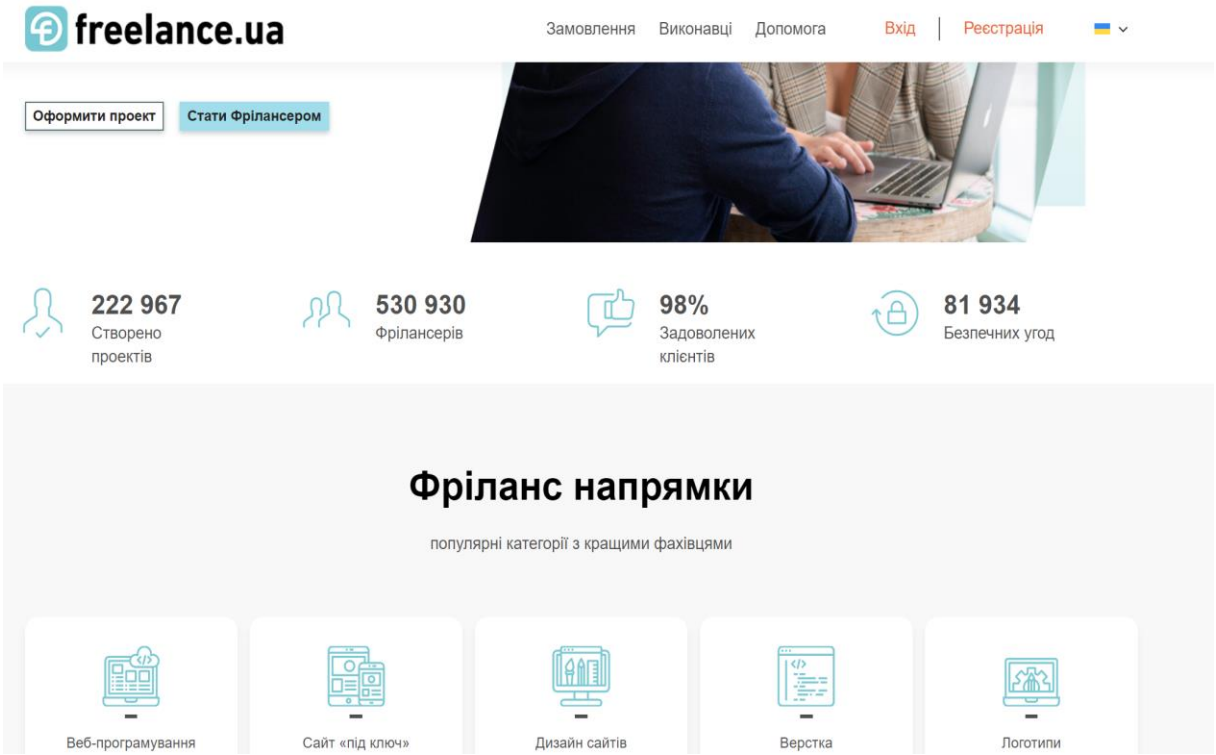


Рисунок 1.7 – Головна сторінка веб-версії платформи Freelance.ua [5]

Можна побачити на головній сторінці реєстрацію, вхід, поділ на категорії, інформацію перехід до виконаців або замовників, зміну мови сайту, трохи про статистику.

Сторінка оголошень містить інформацію про вільних на даний момент робітників(преміум користувачі вище у підборці), тип послуг що вони пропонують, приклади виконаної роботи, відгуки, опис, технології, локацію проживання, сортування (див. рис. 1.8).

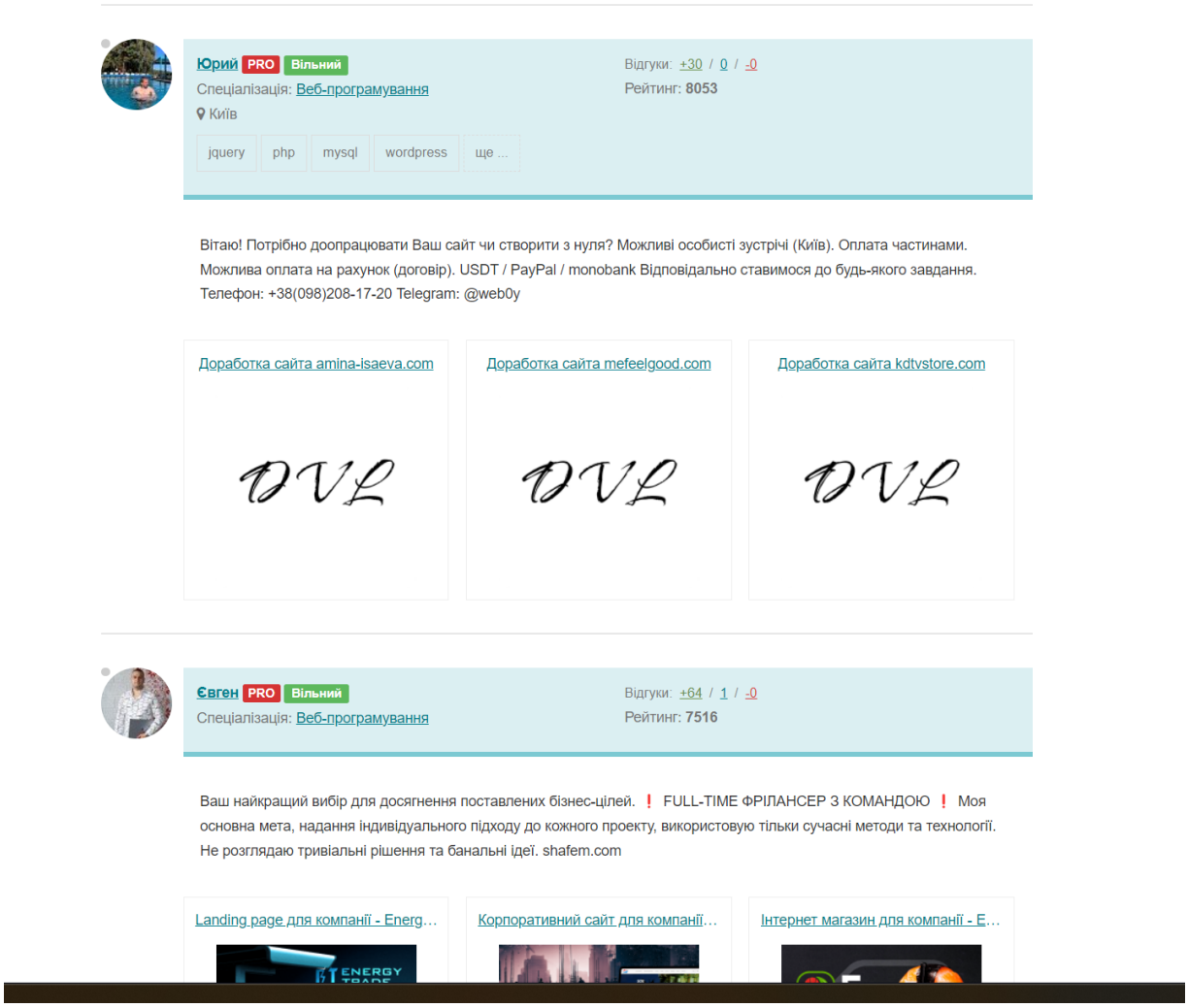


Рисунок 1.8 – Сторінка оголошень веб-версії платформи Freelance.ua [5]

На сторінці детального опису оголошення можна побачити аватарку користувача, детальний опис пропонованих послуг, портфоліо прикладів виконаних замовлень, рейтинг, мінімальний прайс, відгуки (див. рис. 1.9).

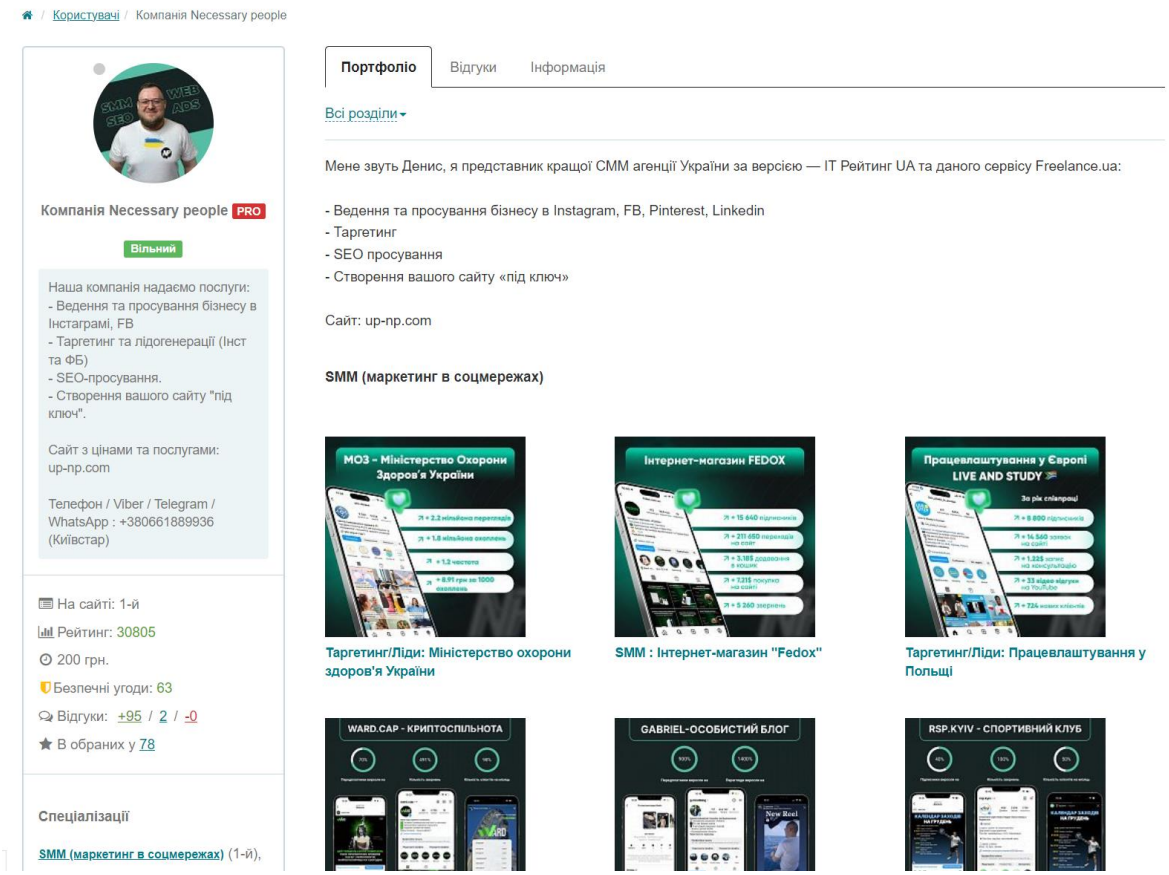


Рисунок 1.9 – Сторінка з детальним описом оголошення платформи Freelance.ua [5]

До переваг можна віднести реєстрацію, 15 років функціонування на ринку, безкоштовне розміщення оголошення, можливість розмістити резюме, приклади виконаної роботи, сортування, приємний дизайн, інформативність побудови структури сайту.

До недоліків віднесемо відсутність мобільного додатку та відсутність партнерів, звісно комісію платформи за замовлення, наявність платного преміуму.

1.4 Вимоги

Проаналізувавши аналоги вже існуючих рішень платформ фриланс-бірж, можна дійти висновку, що розроблений в результаті кваліфікаційної роботи сайт повинен містити:

- реєстрацію;
- авторизацію;
- сортування;
- вихід з сесії;
- пошук за ключовими словами;
- можливість створення безкоштовного оголошення/вакансії;
- поділ на замовника та фрилансера;
- динамічне наповнення сторінок;
- сторінка з підборкою вакансій для фрилансерів;
- сторінка з підборкою оголошень для замовників;
- сторінка с детальною інформацією про обране замовлення/оголошення;
- рейтинг виконавця/роботодавця;
- відсутність преміуму.

Серед технічних вимог до майбутньої платформи можна виділити:

- простоту реєстрації;
- простоту користування;
- захищеність бази даних.

2 ОПИС ЗАСОБІВ ДЛЯ РОЗРОБКИ І ПРОЕКТУВАННЯ

2.1 Програмні засоби

Згідно теми кваліфікаційної роботи, слід звернути увагу на інструменти мови програмування PHP та SQL для розробки бекенду сайту і налаштування бази даних, а також скоріш за все вистачить HTML з CSS та JS для корегування фронтенду. Тож, має сенс ознайомитись ближче з цими технологіями.

PHP – це скриптова мова програмування, спочатку створена для динамічної генерації HTML-сторінок на серверній стороні. Сьогодні PHP широко використовується у веб-розробці, і сервер обробляє її код, перетворюючи його на HTML, який передається клієнтському браузеру. Користувачі не бачать PHP-коду, оскільки їм надсилається вже готовий HTML. Це підвищує рівень безпеки, але зменшує інтерактивність сторінок. Водночас PHP може генерувати JavaScript-код для виконання на стороні клієнта, забезпечуючи більш динамічну взаємодію [6].

SQL (Structured Query Language) – це мова програмування, яка використовується для зберігання, обробки та управління даними у реляційних базах даних. Реляційна база організовує дані у вигляді таблиць, де рядки та стовпці представляють атрибути і зв’язки між значеннями. За допомогою SQL можна виконувати операції зберігання, оновлення, видалення та пошуку даних, а також оптимізувати продуктивність бази даних [7].

HTML (HyperText Markup Language) – це мова розмітки, яка використовується для створення структури вебсторінок. Вона складається з набору тегів, які визначають елементи сторінки та їхній зміст, забезпечуючи основну організацію веб-сайтів [8].

CSS – мова за допомогою якої оформлюються усі веб-сторінки. Для того, щоб підключити CSS стиль до певного HTML тегу чи сукупності тегів використовуються спеціальні HTML атрибути class чи id. HTML і CSS разом з

JavaScript утворюють основу сучасного веб-дизайну. З їх допомогою будь яка інформація, яку містить веб сторінка візуалізується, формується, стилізується та упорядковується [8].

JavaScript (JS) – динамічна, об’єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки [9].

2.2 Середовище для розробки

Для роботи з бекендом нам знадобляться такі програми як phpMyAdmin [10], базу даних оберемо MySQL [11]. Для комфортної роботи в локальному просторі з ними є зручне рішення – Xampp[12]. Щодо IDE – VS Code [13] повинен містити весь необхідний інструментарій для розробки конкретно цього проекту. Тож, є сенс ближче ознайомитись з обраним інструментарієм.

phpMyAdmin – це безкоштовний інструмент, створений на основі PHP, який дозволяє керувати MySQL та MariaDB через веб-інтерфейс. Він підтримує широкий спектр функцій, включаючи управління базами даних, таблицями, стовпцями, зв’язками, індексами, користувачами та дозволами. Крім того, phpMyAdmin забезпечує зручний графічний інтерфейс для виконання основних операцій, але також дозволяє запускати будь-які SQL-запити [10].

MySQL – напевне найрозповсюджена реляційна система керування базами даних. Понад 2000 незалежних постачальників програмного забезпечення, виробників оригінального обладнання та VAR покладаються на MySQL як вбудовану базу даних своїх продуктів, щоб зробити свої додатки, апаратне забезпечення та пристрої більш конкурентоспроможними, швидше вивести їх на ринок і знизити COG (собівартість проданих товарів) [11].

Xampp – просте і комфортне рішення для організації локального серверу з великим функціоналом. Ця збірка встановлюється одним інсталятором і вбудованим phpMyAdmin, MySQL, можливістю довстановлення Composer, Apache, і купи іншого функціоналу, що може знадобитися. У нашому випадку, вистачить лише правильно прописати порти, уникаючи налаштування і інсталяції кожного рішення окремо [10-12].

VS Code – простий та потужний текстовий редактор для програмування. VS Code підтримує майже всі основні мови програмування. Деякі з них поставляються в комплекті, як-от JavaScript, TypeScript, CSS і HTML, але розширення для інших можна знайти на VS Code Marketplace [13].

2.3 Налаштування Xampp

До початку розробки нам знадобиться встановити та правильно налаштувати Xampp [12]. Від його роботи залежить стабільність локального серверу. Не будемо гаяти часу.

Переходимо на офіційний сайт Xampp [12] та натискаємо на кнопку XAMPP for Windows (див. рис. 2.1). Відразу почнеться завантаження і після можна розпочати інсталювати збірку (див. рис. 2.2). У нашому випадку, не треба нічого міняти в налаштуваннях, інсталятор, просто натискаємо «далі» і чекаємо поки інсталятор зробить свою справу (див. рис. 2.3 – 2.4).



Рисунок 2.1 – Офіційний сайт Xampp [12] і посилання на інсталятор

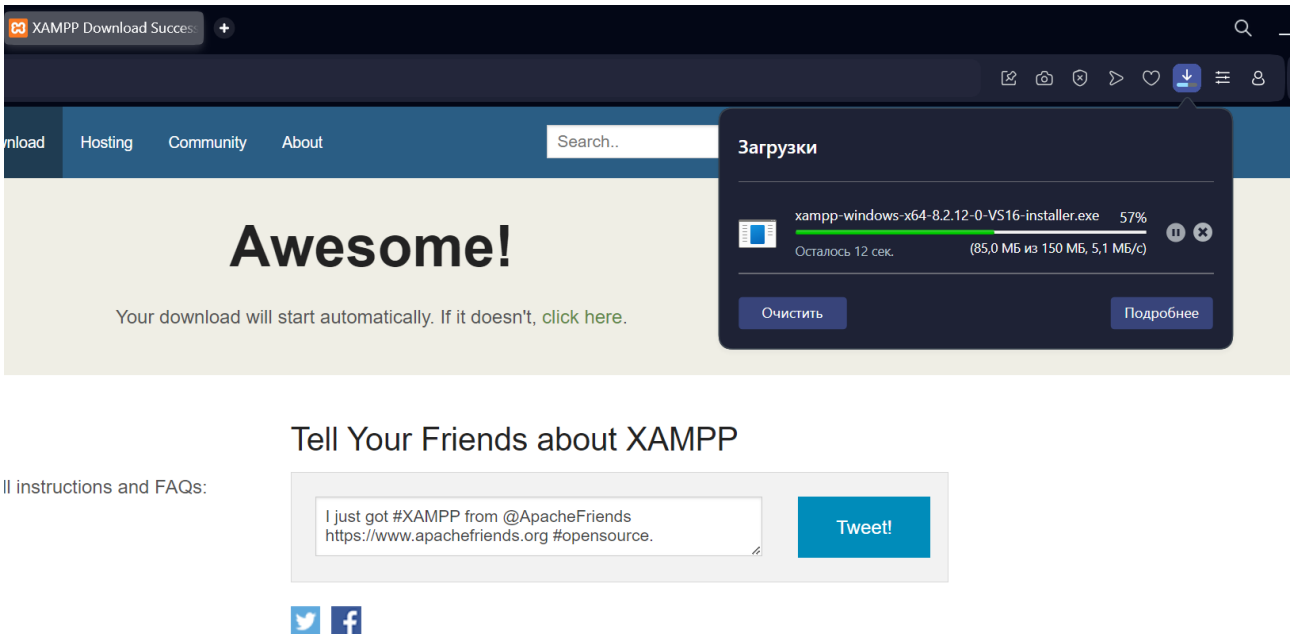


Рисунок 2.2 – Завантаження інсталятора

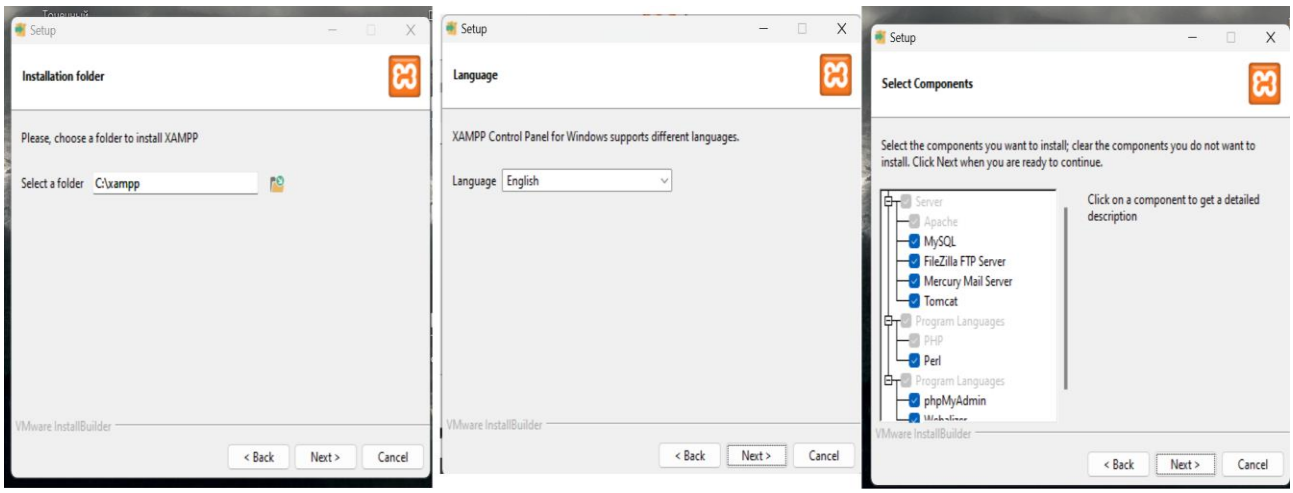


Рисунок 2.3 – Налаштування інсталятора

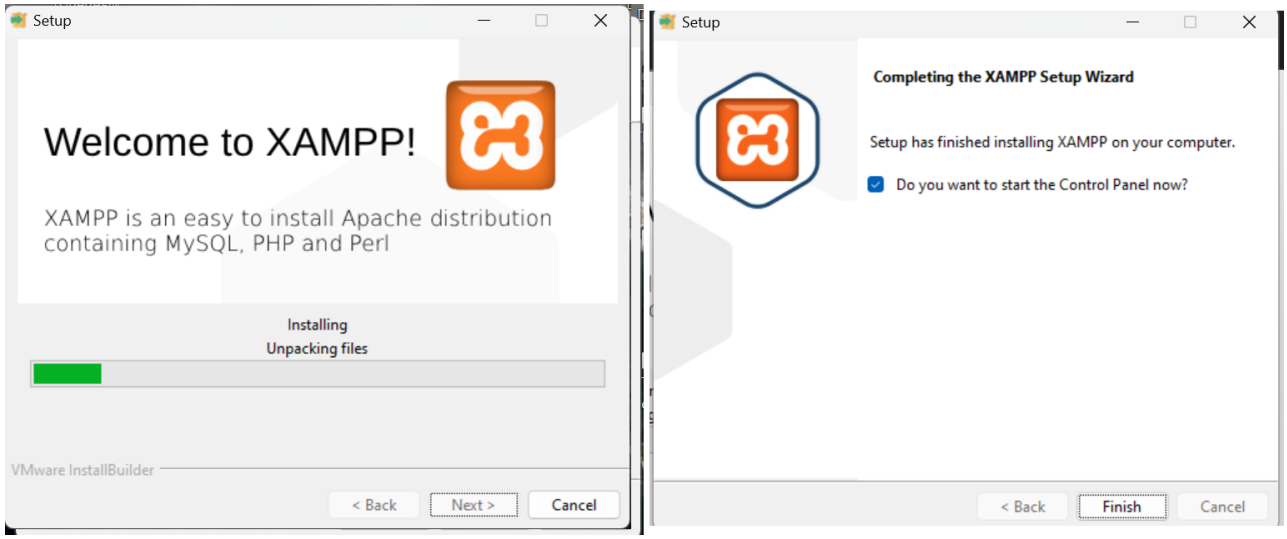


Рисунок 2.4 – Завершення налаштування і інсталяції

Після інсталяції відкриється панель управління програмою. Весь необхідний комплект для роботи з локальним сервером встановлений. Можна подивитись що робить кожна з кнопок, але нас будуть цікавити лише ті що навпроти Apache та MySQL [11]. Відразу необхідно налаштувати, щоб програма вмикалась лише від імені адміністратора, бо без цього працювати просто нічого не буде взагалі(див. рис. 2.5).

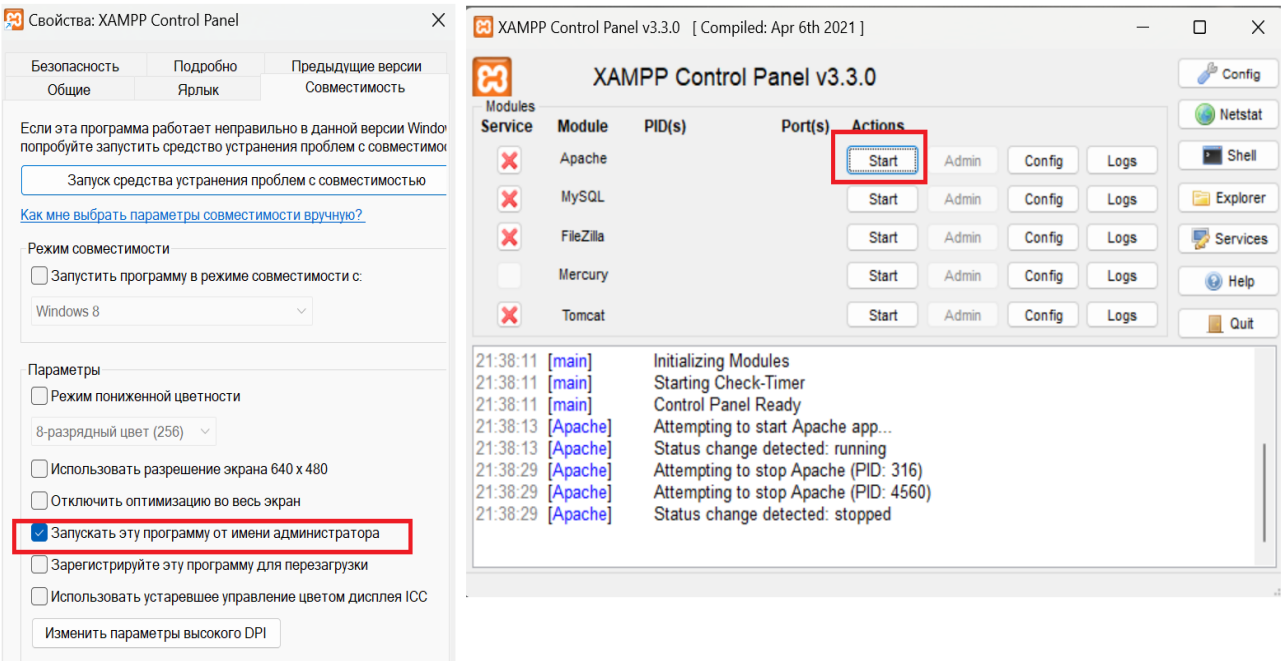


Рисунок 2.5 – Налаштування запуску від імені адміністратора та панель управління Xampp [12]

Після цього нам треба перейти у папку де встановлений Xampp [12] для подальшого налаштування. Нам потрібен файл httpd-vhosts.conf, в моєму випадку він знаходиться по шляху C:\xampp\apache\conf\extra. Відкриємо його за допомогою VS Code [13] та проведемо необхідні налаштування. Нам потрібно прописати VirtualHost для папки, де знаходитиметься наша директорія з проектом. Налаштування можна побачити на малюнку 2.6.

Після цього нам знадобиться файл hosts. Частіше за все на Windows його можна знайти за шляхом C:\Windows\System32\drivers\etc. Знайдемо цей файл і теж відкриємо його за допомогою текстового редактору для налаштування переадресації портів браузеру на папку з проектом за певним посиланням(у нашому випадку YO.loc). Детальні налаштування можна побачити на рис. 2.7.

Якщо все було зроблено вірно, то вмикаємо Xampp [12] та натискаємо на кнопку Start навпроти Apache. Після чого прописуємо в директорії проекту в файлі index.php що завгодно, що може бути скомпільовано і дати вивід на екран.

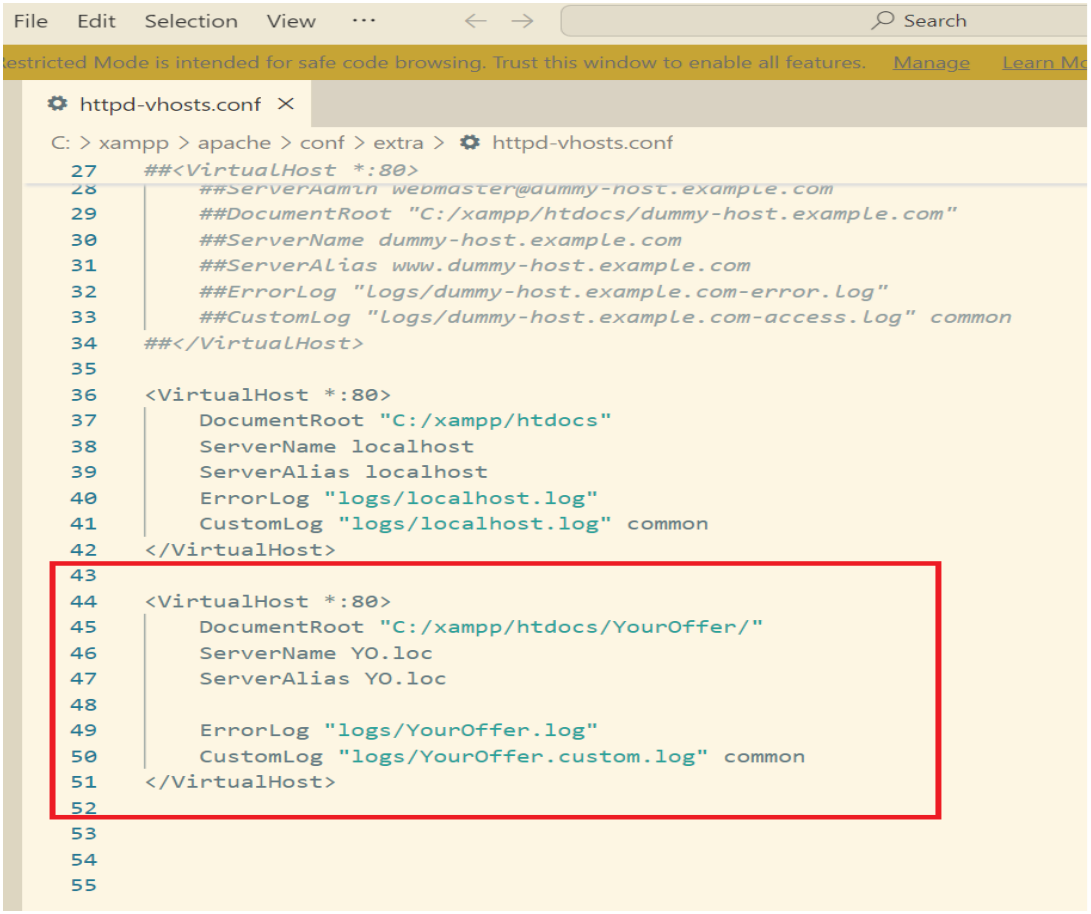


Рисунок 2.6 – Налаштування VirtualPort в файлі httpd-vhosts.conf

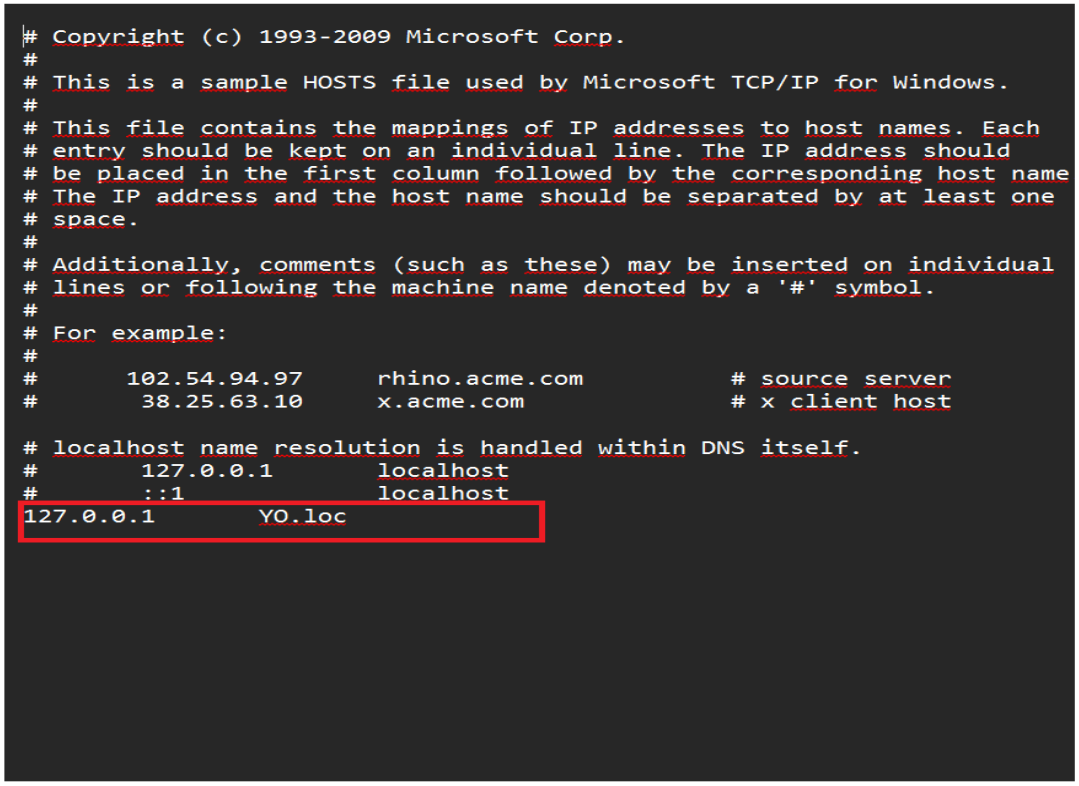


Рисунок 2.7 – Налаштування переадресації в файлі hosts

Якщо все було зроблено вірно, то після переходу в будь-якому браузері на нашому комп’ютері за посиланням, що було прописано в hosts, ми будемо переадресовані на сторінку що була скомпільована з файлу в нашій директорії (див. рис. 2.8). Після цього можна приступати до розробки.

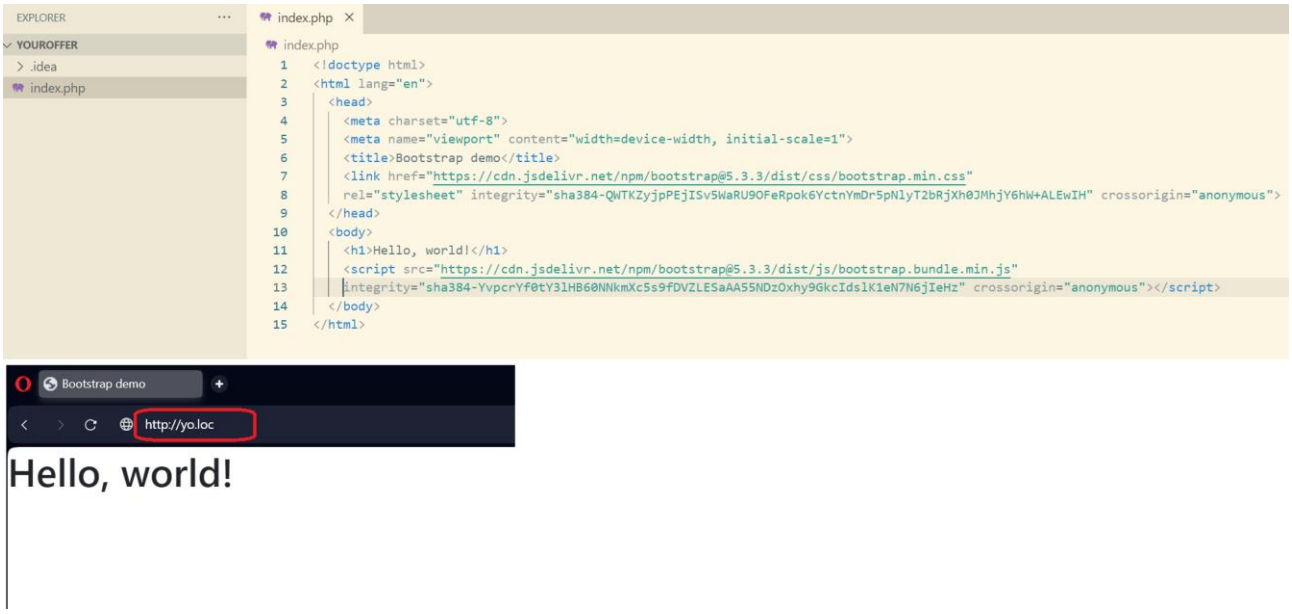


Рисунок 2.8 – Перехід за посиланням в браузері у скомпільований файл папки з проектом

2.4 Діаграма прецендентів

Діаграма прецендентів описує можливу взаємодію між акторами та прецендентами. На рисунку 2.9 побудована діаграма прецендентів для фриланс-біржі.

Діаграма прецедентів (або варіантів використання) показує, як користувачі (актори) взаємодіють із системою.

Складається з основних елементів:

– актори – ролі, які взаємодіють із системою (люди, програми, пристрої);

– прецеденти – описують дії або функції, доступні актору;

– зв’язки – лінії між акторами та прецедентами, що показують взаємодію.

Цей вид діаграми допомагає зрозуміти, хто і як використовує систему.

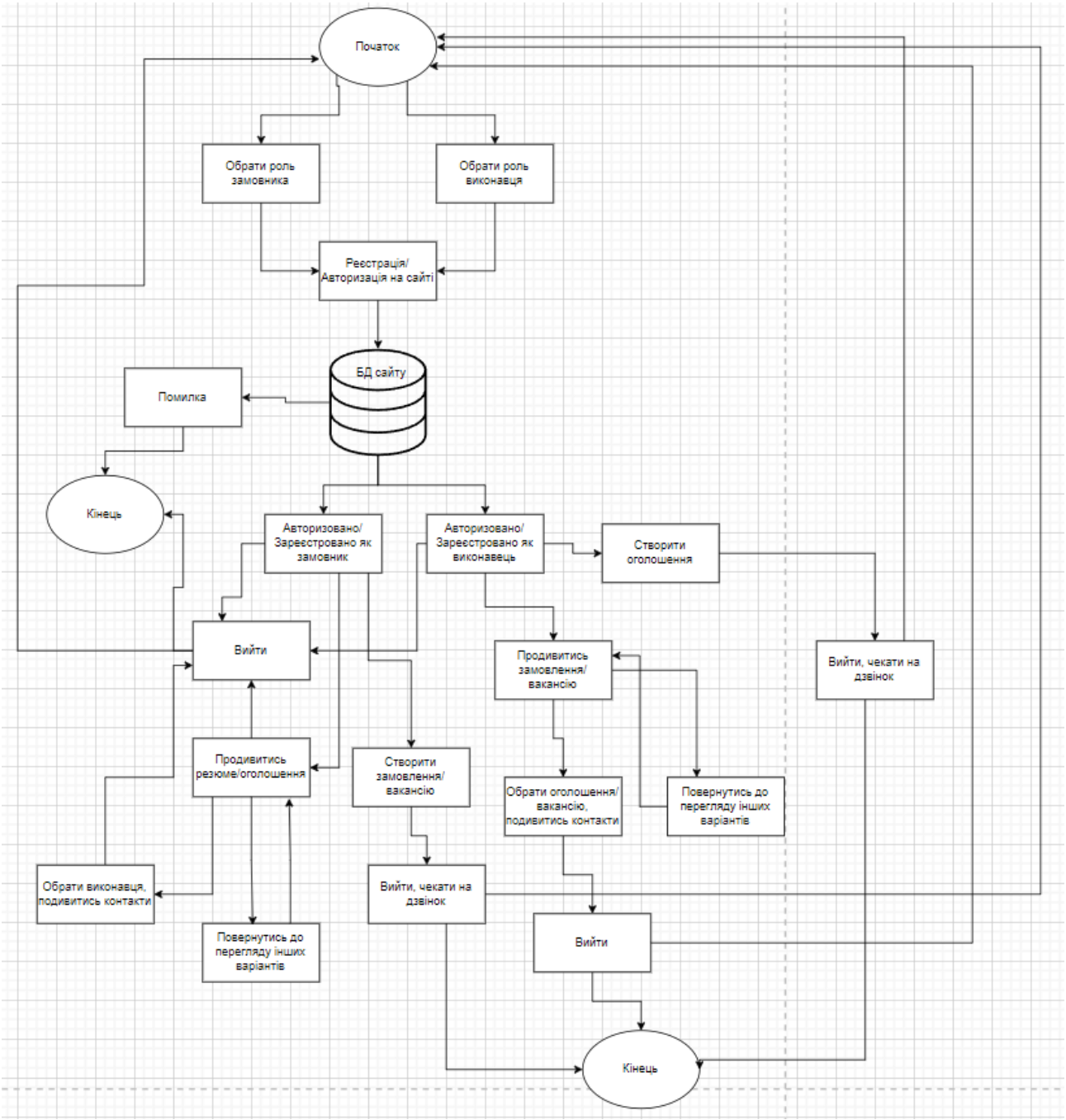


Рисунок 2.9 – Діаграма прецендентів

У нашому випадку є актори, що можуть внаслідок взаємодії з прецендентами фриланс-біржі обрати роль фрилансера або замовника,

продивитись вакансії/оголошення, створити його, видалити, вийти. Стандартний функціонал CRUD.

2.5 Діаграма послідовності

Діаграма послідовності – це UML-діаграма, яка показує взаємодію об’єктів або користувачів у хронологічному порядку. Вона ілюструє, які повідомлення передаються між ними для виконання певного процесу.

На рисунку 2.10 зображена діаграма послідовності взаємодії з фриланс-біржею для актора в ролі замовника.

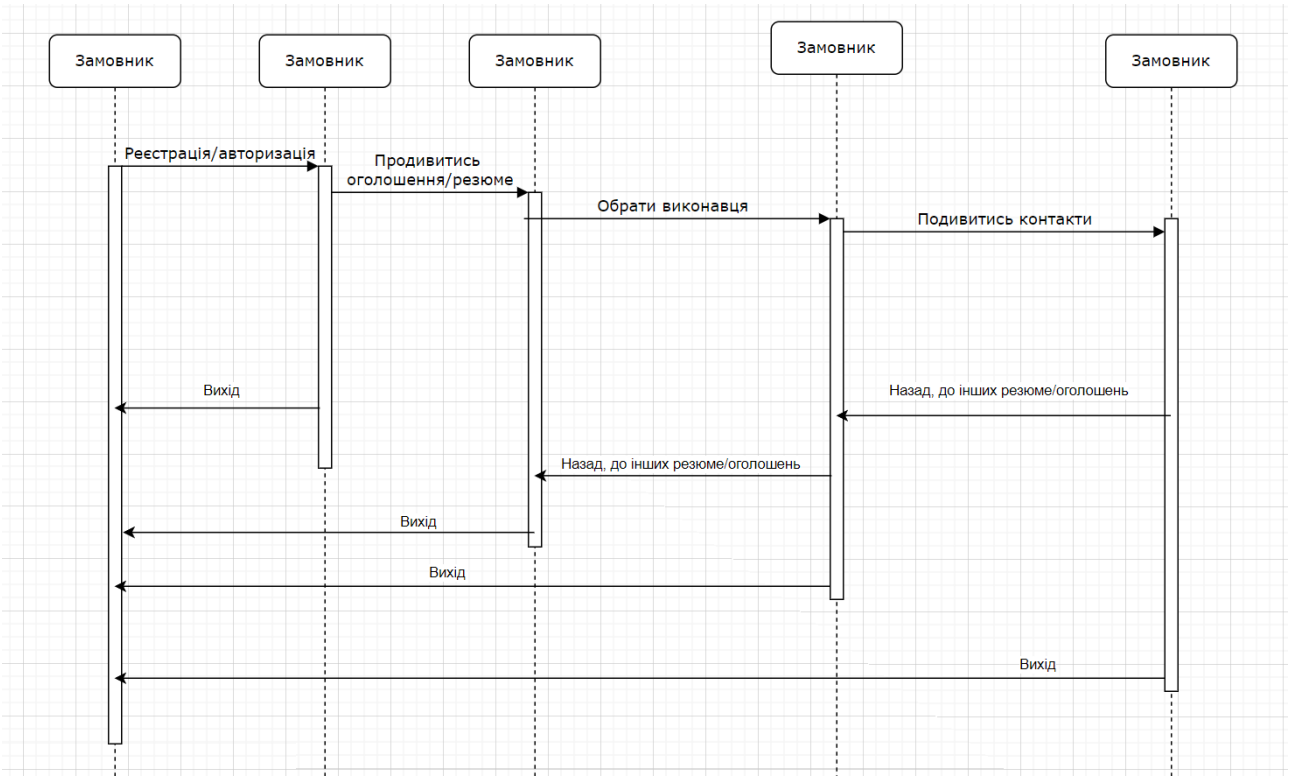


Рисунок 2.10 – Діаграма послідовності

У нашому випадку замовник(актор) потрапляє на сайт, повинен пройти реєстрацію(або авторизацію якщо вже зараєстрований). Після він може відразувийти з сесії, або продивитись оголошення(резюме) виконавців, або вийти з сесії, далі продивитись детальний опис виконавця та послуг що він

пропонує, після чого або отримати його контакти або вийти назад до всіх оголошень, або вийти з сесії.

2.6 Діаграма компонентів

Діаграма компонентів – це UML-діаграма, яка показує, з яких частин складається система та як ці частини взаємодіють між собою і з зовнішніми елементами. Ця діаграма використовується для опису архітектури системи.

На рисунку 2.11 зображена діаграма компонентів для веб-сайту фриланс-біржі.

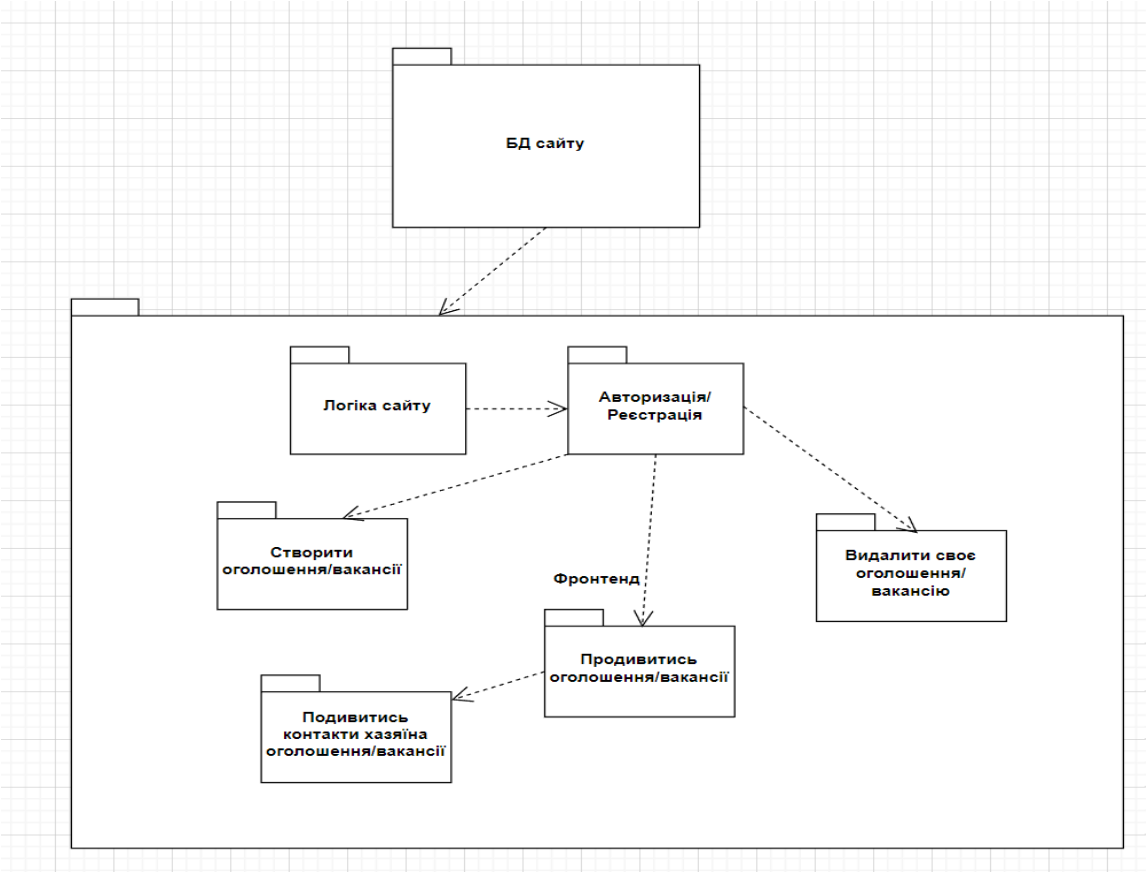


Рисунок 2.11 – Діаграма компонентів

Діаграма компонентів корисна для візуалізації архітектури системи, показуючи, з яких модулів вона складається і як вони взаємодіють. Вона

спрощує розробку, підтримку та масштабування системи, допомагає розподіляти завдання між командами, покращує повторне використання модулів і дає змогу аналізувати залежності та інтеграцію із зовнішніми системами.

2.7 База даних

Нам варто розробити план бази даних для фриланс-біржі завчасно, щоб простіше її потім реалізувати та заповнити даними (див. рис. 2.12).

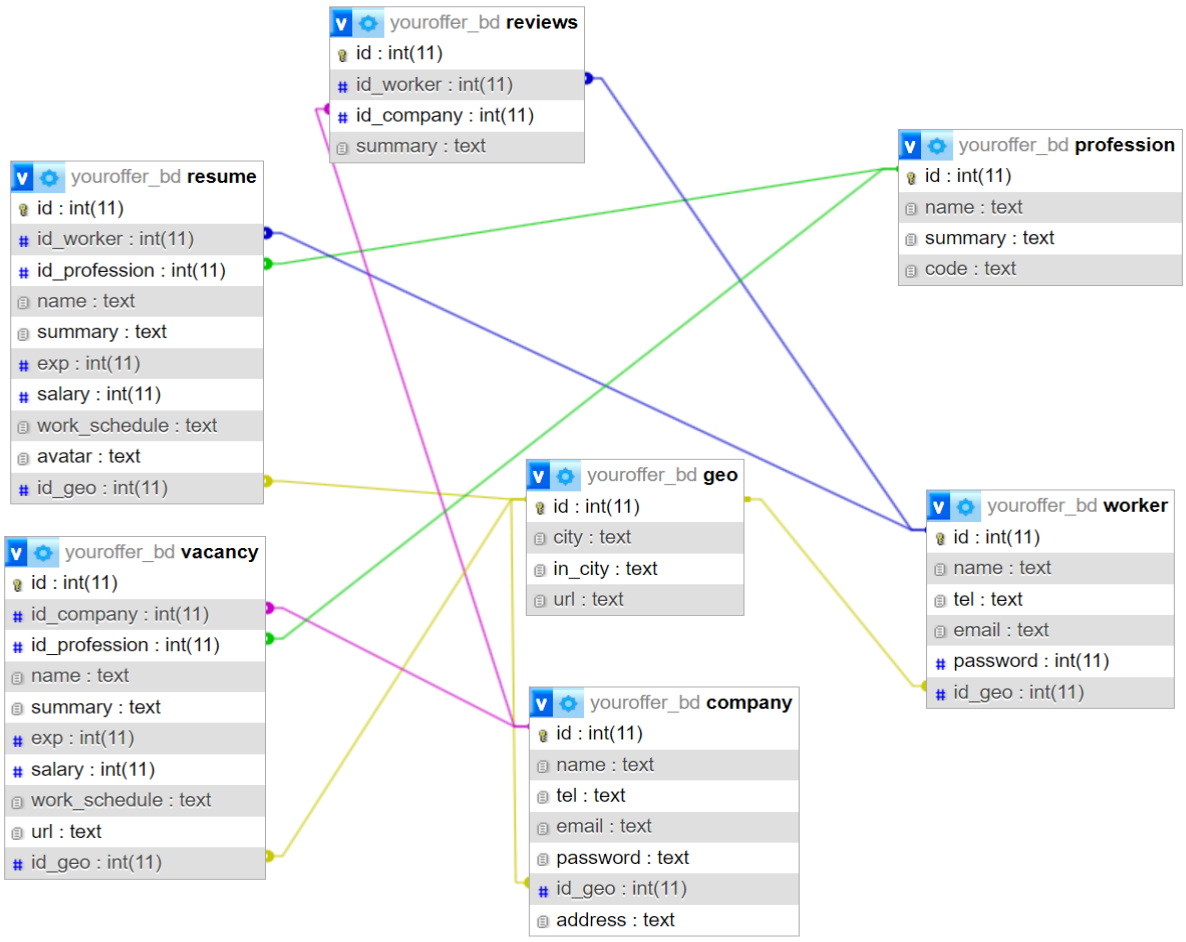


Рисунок 2.12 – База даних з побудованими зв’язками

Без створеної бази даних жоден сайт не буде мати майже всього функціоналу, який на нього був покладений або тільки планується. Саме в ній

зберігається інформація для відображення її у зручному вигляді користувачу на фронтенді, і саме її слід охороняти комплексним заходом фільтрації даних для унеможливлювання поломки або спроб взлому хакерами.

У нашому випадку було створено базу даних youroffer_bd , яка складається з 7 таблиць, логічно зв’язаних між собою, яка і буде зберігати всі необхідні дані для коректного функціоналу і відображення інформації на веб-сайті.

Якщо все зроблено правильно, можна приступати до самої розробки веб-сайту.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Розробка веб-сайту фриланс-біржі

Для початку, має сенс побудувати макет сторінок майбутнього сайту. Так буде більш зрозуміло шо нам потрібно зробити та який функціонал де розміщувати. Будемо використовувати Figma – онлайн-сервіс для розробки інтерфейсів й прототипування [14].

За допомогою інструментарію сервісу, приступимо до розробки макету інтерфейсу сторінки з оголошеннями (див. рис. 3.1) та сторінки з детальним описом оголошення (див. рис. 3.2).

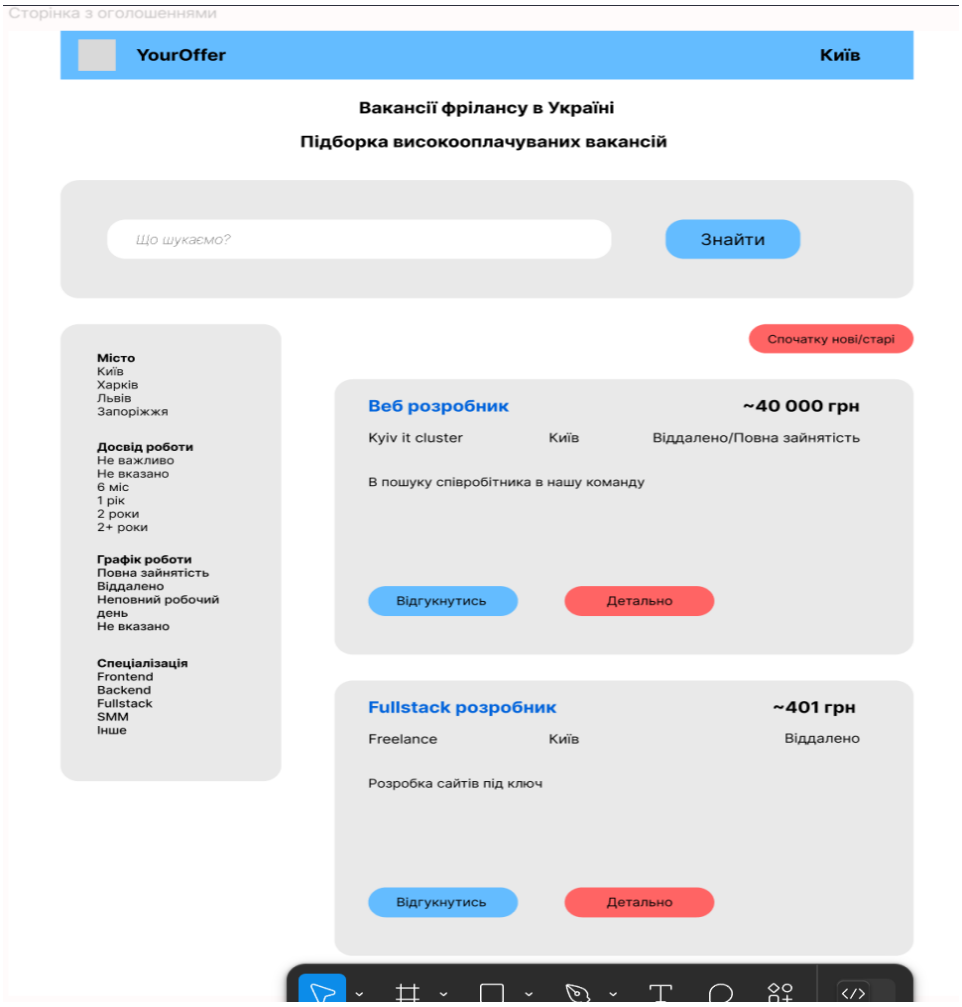


Рисунок 3.1 – Макет сторінки з оголошеннями

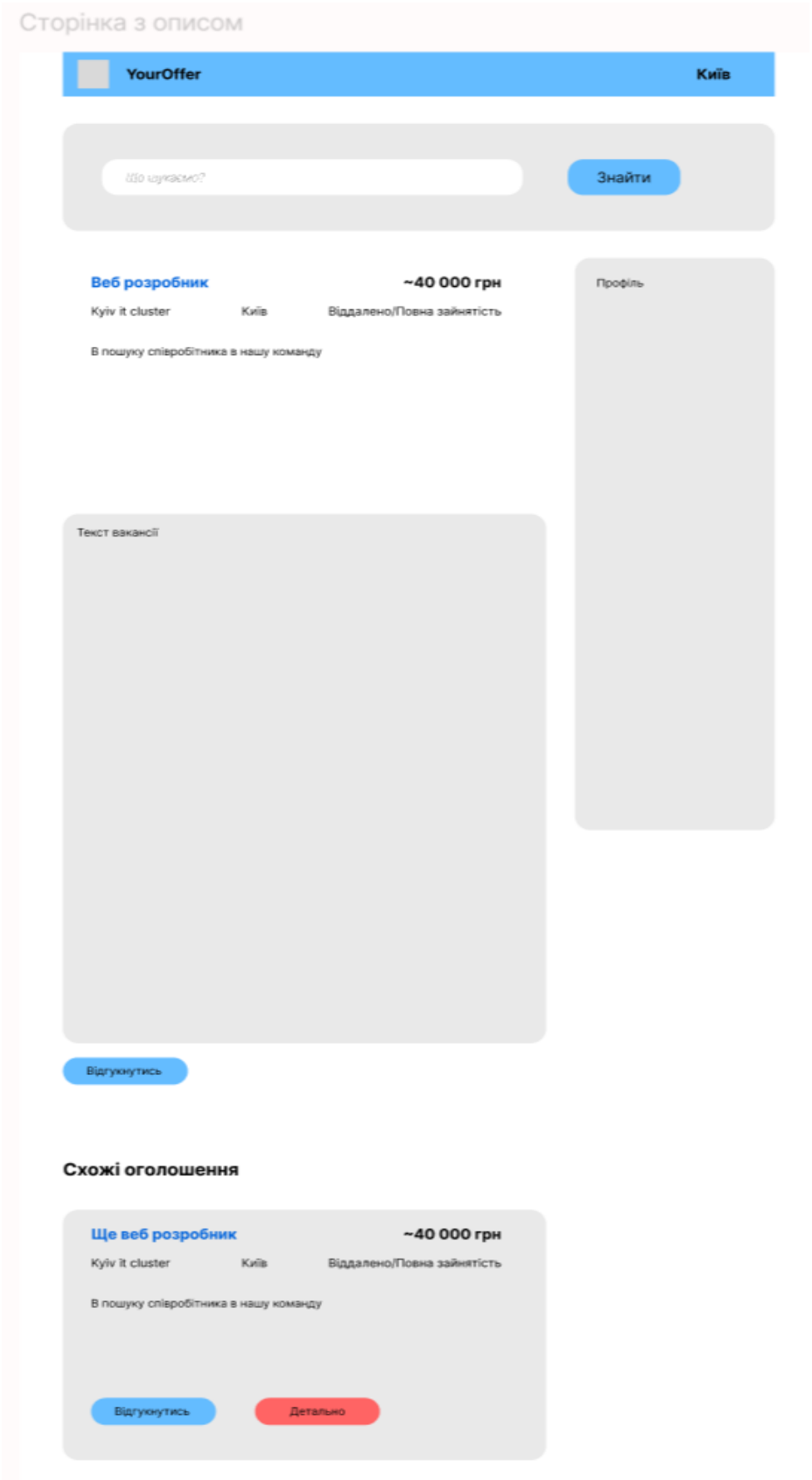


Рисунок 3.2 – Макет сторінки з детальним описом оголошення

Створивши приблизний макет, можна приступити до самої розробки.

Для початку пропишемо в routing.php необхідні посилання на сторінки, щоб до них можна було б пройти або через нього або взагалі ніяк. Саме через цей файл повинен йти трафік від користувача і бути переадресованим на іншу сторінки (див. рис. 3.3).

```php
<?php

session_start();
$link = new mysqli('localhost', 'root', '', 'youroffer_bd');

$url = $_SERVER['REQUEST_URI'];
$bd='/';

if($url=='/') {

    include('index.php');
    exit();
}

if($url=='/company/') {

    $bd='company';
    include('index.php');
    exit();
}

if($url=='/worker/') {

    $bd='worker';
    include('index.php');
    exit();
}

if($url=='/company/new_vacancy') {

    include('./company/new_vacancy.php');
    exit();
}

if($url=='/worker/new_advertisement') {

    include('./worker/new_advertisement.php');
    exit();
}

if($url=='/worker/new_advertisement') {

    include('./worker/new_advertisement.php');
    exit();
}

if($url=='/exit') {

    $_SESSION=array();
    session_destroy();
    header('Location:http://yo.loc');
    exit();

}

if(strripos($url,'vacancy') !=false) {

   include('./worker/vacancy.php');
   exit();

}

if(strripos($url,'advertisement') !=false) {

   include('./company/advertisement.php');
   exit();
}
?>
```

Рисунок 3.3 – Вміст файлу routing.php

Далі, поділимо html сторінку на декілька частин. На header.php і index.php. Щодо header.php – в ньому буде зберігатися логіка взаємодії з кнопками, локація обрана при реєстрації користувачем, назва сайту, а також шматок логіки реєстрації/авторизації. Користувач відразу повинен обрати, в якій ролі він збирається пройти авторизацію/реєстрацію, і без цього інші кнопки будуть неактивні. Також, деякі кнопки і локація будуть сховані в залежності від ролі користувача і наявності авторизованої сесії. До даних, що

вводить користувач при реєстрації/авторизації застосовується фільтрація. На рисунках 3.4–3.7 зображений вміст файлу header.php.

```php
<?php

if (isset($_POST['name']))
  if (isset($_POST['email']))
    if (isset($_POST['password']))

    {
      if(filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) { $email=$_POST['email'];} else exit();
      $password=$_POST['password'];
      $name=$_POST['name'];
      $id_geo=$_POST['id_geo'];
      $password=md5($password.strrev($email));
      $tel=$_POST['tel'];

      $sql="INSERT INTO `$bd`(`name`, `email`, `password`,`id_geo`, `tel`) VALUES ('$name', '$email', '$password', '$id_geo', '$tel')";
      $result = mysqli_query($link, $sql);
    }

if (isset($_POST['email']))
  if (isset($_POST['password'])){

    if(filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) { $email=$_POST['email'];} else exit();
    $password=$_POST['password'];

    $password=md5($password.strrev($email));

    $sql="SELECT * FROM `$bd` WHERE email='$email' AND password='$password'";
    $result = mysqli_query($link, $sql);
    while($row=mysqli_fetch_array($result)) {
      $_SESSION['bd']=$bd;
      $_SESSION['id']=$row['id'];
      $id_geo=$row['id_geo'];

    }
    $sql="SELECT * FROM `geo` WHERE `id`='$id_geo'";
    $result = mysqli_query($link, $sql);
    while($row=mysqli_fetch_array($result)) {
      $city=$row['city'];
      $_SESSION['city'] =$city;
    }
    }
  }

?>

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>YourOffer</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
  </head>
  <body>

  <div class = "container-fluid" style="background:#64BCFF">
    <div class="container">

      <div class="row">
        <div class="col-md-2 col-12" style="padding: 23px 0px 23px 0px;">
          <h1>YourOffer</h1>
        </div>
        <div class="col-md-2 col-12" style="padding: 20px;" >
          <?php if ($bd=='company') echo '<b>'; ?>
          <?php if ($bd=='company' || $bd=='/' ) { ?> <a href="/company/"><button type="button"
          style="border-radius:20px; margin-left:25px;" class="btn btn-info">Роботодавець</button></a><?php }?>
          <?php if ($bd=='company') echo '</b>'; ?>

        </div>
        <div class="col-md-2 col-12" style="padding: 20px;">
          <?php if ($bd=='worker') echo '<b>'; ?>
          <?php if ($bd=='worker' || $bd=='/' ) { ?> <a href="/worker/"><button type="button" style="border-radius:20px; margin-left:25px;"
          class="btn btn-info">Фрилансер</button></a> <?php }?>
          <?php if ($bd=='worker') echo '</b>'; ?>
        </div>
        <div class="col-md-2 col-12" style="padding: 20px;">
```

Рисунок 3.4 – Вміст файлу header.php

```php
<?php if(isset($_SESSION['id']) ==false) { ?>
<a href="#reg" data-bs-toggle="modal" data-bs-target="#reg"> <button type="button" style="border-radius:20px; margin-left:25px;"
class="btn btn-info">Реєстрація</button></a>
<?php }?>

<?php if ($bd=='company')
if (isset($_SESSION['id'])) { ?>
<a href="new_vacancy"> <button type="button" style="border-radius:20px; margin-left:25px;"
class="btn btn-info">Додати вакансію</button></a>
<?php } ?>

<?php if ($bd=='worker')
if (isset($_SESSION['id'])) { ?>
<a href="new_advertisement"> <button type="button" style="border-radius:20px; margin-left:25px;"
class="btn btn-info">Додати оголошення</button></a>
<?php } ?>

</div>
<div class="col-md-2 col-12" style="padding: 20px;">
<?php if(isset($_SESSION['id']) ==false) { ?>
<a href="#auth" data-bs-toggle="modal" data-bs-target="#auth"><button type="button" style="border-radius:20px; margin-left:25px;"
class="btn btn-info"> Авторизація</button></a>
<?php } ?>
<?php if(isset($_SESSION['id'])) {?> <a href="/exit"><button type="button" style="border-radius:20px;"
class="btn btn-info">Вийти</button></a> <?php } ?>
</div>
<div class="col-md-2 col-12" style="padding: 23px 0px 23px 0px;">
<?php if (isset($_SESSION['id'])) { ?>
<h1><?php echo $_SESSION['city'];?></h1>
<?php } ?>
<div class="r10"> </div>
</div>
</div>
</div>
</div>
<div class="r25"> </div>

<?php if ($bd=='worker') {?>
<div class="modal fade" id="reg" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-5" id="exampleModalLabel">Реєстрація фрілансером</h1>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
<form method="POST">

<div class="mb-3">
<label for="exampleFormControlInput1" class="form-label">Назва компанії/або Ім'я, Фамілія</label>
<input name="name" type="text" class="form-control" id="exampleFormControlInput1" placeholder="Тарас Шевченко">
</div>

<div class="mb-3">
<label for="exampleFormControlInput1" class="form-label">Email</label>
<input name="email" type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">
</div>

<div class="mb-3">
<label for="exampleFormControlInput1" class="form-label">Телефон</label>
<input name="tel" type="tel" class="form-control" id="exampleFormControlInput1" placeholder="+380961233212">
</div>

<div class="mb-3">
<label for="exampleFormControlInput1" class="form-label">Пароль</label>
<input name="password" type="password" class="form-control" id="exampleFormControlInput1" placeholder="1234">
</div>

<div class="r20"></div>

<select name="id_geo" class="form-select" aria-label="Default select example">
<option value="1">Київ</option>
<option value="2">Харків</option>
<option value="3">Дніпро</option>
<option value="4">Запоріжжя</option>
<option value="5">Львів</option>
</select>

<div class="r20"></div>

<input type="submit" value="Зареєструватися">
</form>
</div>
</div>
</div>
</div>
```

Рисунок 3.5 – Продовження вмісту файлу header.php

```
<div class="modal fade" id="auth" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">Ввійти фрілансером</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <form method="POST">

        <div class="mb-3">
          <label for="exampleFormControlInput1" class="form-label">Email</label>
          <input name="email" type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">
        </div>

        <div class="mb-3">
          <label for="exampleFormControlInput1" class="form-label">Пароль</label>
          <input name="password" type="password" class="form-control" id="exampleFormControlInput1" placeholder="1234">
        </div>

        <input type="submit" value="Вхід">
        </form>
      </div>
    </div>
  </div>
</div>
<?php };?>


<?php if ($bd=='company') {?>
<div class="modal fade" id="reg" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">Реєстрація роботодавцем</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <form method="POST">

        <div class="mb-3">
          <label for="exampleFormControlInput1" class="form-label">Назва компанії/або Ім'я, Фамілія</label>
          <input name="name" type="text" class="form-control" id="exampleFormControlInput1" placeholder="Kyiv IT Cluster">
        </div>

        <div class="mb-3">
          <label for="exampleFormControlInput1" class="form-label">Email</label>
          <input name="email" type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">
        </div>

        <div class="mb-3">
          <label for="exampleFormControlInput1" class="form-label">Телефон</label>
          <input name="tel" type="tel" class="form-control" id="exampleFormControlInput1" placeholder="+380961233212">
        </div>

        <div class="mb-3">
          <label for="exampleFormControlInput1" class="form-label">Пароль</label>
          <input name="password" type="password" class="form-control" id="exampleFormControlInput1" placeholder="1234">
        </div>

        <div class="r20"></div>

        <select name="id_geo" class="form-select" aria-label="Default select example">
        <option value="1">Київ</option>
        <option value="2">Харків</option>
        <option value="3">Дніпро</option>
        <option value="4">Запоріжжя</option>
        <option value="5">Львів</option>
        </select>

        <div class="r20"></div>

        <input type="submit" value="Зареєструватися">
        </form>
      </div>
    </div>
  </div>
</div>
```

Рисунок 3.6 – Продовження вмісту файлу header.php

```
243   </div>
244
245
246   <div class="modal fade" id="auth" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
247     <div class="modal-dialog">
248       <div class="modal-content">
249         <div class="modal-header">
250           <h1 class="modal-title fs-5" id="exampleModalLabel">Ввійти роботодавецем</h1>
251           <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
252         </div>
253         <div class="modal-body">
254           <form method="POST">
255
256           <div class="mb-3">
257             <label for="exampleFormControlInput1" class="form-label">Email</label>
258             <input name="email" type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">
259           </div>
260
261           <div class="mb-3">
262             <label for="exampleFormControlInput1" class="form-label">Пароль</label>
263             <input name="password" type="password" class="form-control" id="exampleFormControlInput1" placeholder="1234">
264           </div>
265
266           <input type="submit" value="Вхід">
267           </form>
268         </div>
269       </div>
270     </div>
271   </div>
272   <?php };?>
```

Рисунок 3.7 – Кінець вмісту файлу header.php

Далі, найоб’ємніший файл, index.php. У ньому відображена логіка відображення оголошень/вакансій, даних про авторів, вибору конкретного запису для переходу на сторінку з більш детальною інформацією, взаємодії користувача з сайтом в залежності від його ролі, сортування, пошук, підключення стилів. На малюнках 3.8 – 3.11 зображений вміст файлу index.php.

```
1   <?php include('header.php'); ?>
2
3   <div class="containder">
4     <center>
5     <?php   if ($bd=='company') {?>
6       <h2>Фрилансери в Україні</h2>
7       <h3>Підборка високоякісних резюме</h3>
8     <?php }
9     else if($bd=='worker' || $bd='/') {?>
10      <h2>Вакансії фрилансу в Україні</h2>
11      <h3>Підборка високоплачуваних вакансій</h3>
12    <?php }?>
13    </center>
14  </div>
15
16  <div class="r25"></div>
17
18  <div class="container">
19    <div class="row" style="border-radius:20px; background:#E9E9E9;">
20      <div class="col-md-2 col-12"></div>
21        <div class="col-md-8 col-12">
22          <div class="r50"> </div>
23            <form class="d-flex" role="search">
24              <input class="form-control me-2" style="border-radius:20px;" type="search" placeholder="Що шукаємо?" aria-label="Search">
25              <button type="button" style="border-radius:20px;" class="btn btn-info">Знайти</button>
26            </form>
27          <div class="r50"> </div>
28        </div>
29      <div class="col-md-2 col-12"></div>
30    </div>
31  </div>
```

Рисунок 3.8 – Вміст файлу index.php

```
<div class="r50"> </div>

<div class ="container">
  <div class="row">
    <div class="col-md-3 col-12">
      <div style="border-radius:20px; background:#E9E9E9; padding: 20px 20px 20px 20px;">

        <div class="r10"> </div>

        <h5>Місто</h5>

          <div class="r10"> </div>

          <select class="form-select" aria-label="Default select example">
            <option selected>Оберіть місто</option>
            <option value="1">Київ</option>
            <option value="2">Харків</option>
            <option value="3">Дніпро</option>
            <option value="4">Запоріжжя</option>
            <option value="5">Львів</option>
          </select>

          <div class="r30"> </div>

        <h5>Досвід роботи</h5>

          <div class="r10"> </div>

          <div class="form-check">
          <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault2" checked>
          <label class="form-check-label" for="flexRadioDefault2">
            <h6>Не важливо</h6>
          </label>
          </div>

          <div class="form-check">
          <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault1">
          <label class="form-check-label" for="flexRadioDefault1">
            <h6>1 рік</h6>
          </label>
          </div>

          <div class="form-check">
          <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault1">
          <label class="form-check-label" for="flexRadioDefault1">
            <h6>2 роки</h6>
          </label>
          </div>

          <div class="r30"> </div>

          <h5>Графік роботи</h5>

          <div class="r10"> </div>

          <div class="form-check">
          <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault2" checked>
          <label class="form-check-label" for="flexRadioDefault2">
            <h6>Не вказано</h6>
          </label>
          </div>


          <div class="form-check">
          <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault1">
          <label class="form-check-label" for="flexRadioDefault1">
            <h6>Повна зайнятість</h6>
          </label>
          </div>

          <div class="form-check">
          <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault1">
          <label class="form-check-label" for="flexRadioDefault1">
            <h6>Віддалено</h6>
          </label>
          </div>
```

Рисунок 3.9 – Продовження вмісту файлу index.php

```
                <div class="form-check">
                <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault1">
                <label class="form-check-label" for="flexRadioDefault1">
                  <h6>Неповний робочий день</h6>
                </label>
                </div>

                <div class="r30"> </div>

                <h5>Спеціалізація</h5>

                <div class="r10"> </div>

                <div class="form-check">
                <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault">
                <label class="form-check-label" for="flexCheckDefault">
                  <h6>Frontend</h6>
                </label>
                </div>

                <div class="form-check">
                <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault">
                <label class="form-check-label" for="flexCheckDefault">
                  <h6>Backend</h6>
                </label>
                </div>

                <div class="form-check">
                <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault">
                <label class="form-check-label" for="flexCheckDefault">
                  <h6>Fullstack</h6>
                </label>
                </div>

                <div class="form-check">
                <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault">
                <label class="form-check-label" for="flexCheckDefault">
                  <h6>SMM</h6>
                </label>
                </div>

                <div class="r10"> </div>

            </div>

        </div>
        <div class="col-md-9 col-12">

            <div class="r40"></div>

<?php   if ($bd=='worker' || $bd=='/' ) {
            $sql="SELECT * FROM `vacancy`";
            $result = mysqli_query($link, $sql);
            while($row=mysqli_fetch_array($result)) {
                $id=$row['id'];
                $id_company=$row['id_company'];
                $name=$row['name'];
                $summary=$row['summary'];
                $exp=$row['exp'];
                $salary=$row['salary'];
                $work_schedule=$row['work_schedule'];
                $id_geo=$row['id_geo'];

?>
            <div style="border-radius:20px; background:#E9E9E9; padding: 57px 57px 57px 57px">
                <div class="row">
                    <div class="col-9">
                        <a href="/worker/vacancy?id=<?php echo $id; ?>"><h4><?php echo $name; ?></h4></a>
                    </div>
                    <div class="col-3">
```

Рисунок 3.10 – Продовження вмісту файлу index.php

```
                <div class="col-3">
                  <h4><?php echo $salary; ?> грн</h4>
                </div>
              </div>

              <div class="r10"></div>

              <div class="row">
                <div class="col-4">
                  <p><?php echo $work_schedule; ?></p>
                </div>
                <div class="col-4">

                </div>
                <div class="col-4">

                </div>
              </div>

              <div class="r10"></div>

              <p><?php echo $summary; ?></p>

              <div class="r30"></div>

              <a href="/worker/vacancy?id=<?php echo $id; ?>"><button type="button" style="border-radius:20px;" class="btn btn-info">Детально</button></a>

              <div class="r30"></div>
            </div>

            <div class="r40"></div>
<?php }}?>
<?php   if ($bd=='company') {

            $sql="SELECT * FROM `resume`";
            $result = mysqli_query($link, $sql);
            while($row=mysqli_fetch_array($result)) {
            while($row=mysqli_fetch_array($result)) {
              $id=$row['id'];
              $id_worker=$row['id_worker'];
              $name=$row['name'];
              $summary=$row['summary'];
              $exp=$row['exp'];
              $salary=$row['salary'];
              $work_schedule=$row['work_schedule'];
              $id_geo=$row['id_geo'];
?>
            <div style="border-radius:20px; background:#E9E9E9; padding: 57px 57px 57px 57px">
              <div class="row">
                <div class="col-9">
                  <a href="/company/advertisement?id=<?php echo $id; ?>"><h4><?php echo $name; ?></h4></a>
                </div>
                <div class="col-3">
                  <h4><?php echo $salary; ?> грн</h4>
                </div>
              </div>

              <div class="r10"></div>

              <div class="row">
                <div class="col-4">
                  <p><?php echo $work_schedule; ?></p>
                </div>
                <div class="col-4">

                </div>
                <div class="col-4">

                </div>
              </div>

              <div class="r10"></div>

              <p><?php echo $summary; ?></p>

              <div class="r30"></div>

              <a href="/company/advertisement?id=<?php echo $id; ?>"><button type="button" style="border-radius:20px;" class="btn btn-info">Детально</button></a>

              <div class="r30"></div>
            </div>

            <div class="r40"></div>
<?php }}?>
          </div>
        </div>
      </div>
      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
      <?php include('style.php');?>
    </body>
  </html>
```

Рисунок 3.11 – Кінець вмісту файлу index.php

Далі, пропишемо необхідні стилі та доступи у файли styles.php та .htaccess. На рисунку 3.12 зображений вміст файлів styles.php та .htaccess.

style.php

```
<style>
    h1 {font-size: 24px;}
    h2 {font-size: 22px;}
    h3 {font-size: 20px;}

    .r5 {width: 100%; height: 5px;}
    .r10 {width: 100%; height: 10px;}
    .r15 {width: 100%; height: 15px;}
    .r20 {width: 100%; height: 20px;}
    .r25 {width: 100%; height: 25px;}
    .r30 {width: 100%; height: 30px;}
    .r35 {width: 100%; height: 35px;}
    .r40 {width: 100%; height: 40px;}
    .r45 {width: 100%; height: 45px;}
    .r50 {width: 100%; height: 50px;}

    .btn btn-info {"border-radius:20px;"}
    .btn btn-danger {"border-radius:20px; margin-left:25px;"}
</style>
```

.htaccess

```
RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ routing.php [L,QSA]
```

Рисунок 3.12 – Вміст файлів styles.php та .htaccess

Далі, створимо дві папки – company та worker. Це необхідно для переадресації в routing для того щоб користувач не з однією роллю не отримав функціонал, доступний лише іншій ролі. Користувач може зареєструватися і як виконавець і як замовник, але в один момент бути авторизованим лише за однією. Для зміни ролі треба вийти з сесії.

У папці company створимо файли – advertisement.php та new_vacancy.php. Ці сторінки будуть доступні лише користувачам з активною сесією в ролі замовника(кнопка «Роботодавець» в шапці).

В файлі advertisement.php прописана логіка відображення даних вже створених оголошень від фрилансерів з основною інформацією про них, можливістю перейти за посиланням на детальну сторінку з карткою оголошення та перегляду даних автора оголошення. Також зроблена перевірка на відповідність ролі користувача – лише роль виконавця для доступу, інших викине на основну сторінку сайту. На рисунках 3.13 – 3.15 зображений вміст файлу advertisement.php.

```php
<?php

if((isset($_SESSION['bd'])) & (isset($_SESSION['id'])) &($_SESSION['bd']=='company')){
  $bd='company';
}

else {header('Location:http://yo.loc'); exit();}


$id=$_GET['id'];
$sql="SELECT * FROM `resume` WHERE `id`='$id'";
$result = mysqli_query($link, $sql);
while($row=mysqli_fetch_array($result)) {

  $id_worker=$row['id_worker'];
  $id_profession=$row['id_profession'];
  $name=$row['name'];
  $summary=$row['summary'];
  $exp=$row['exp'];
  $salary=$row['salary'];
  $work_schedule=$row['work_schedule'];
  $id_geo=$row['id_geo'];
  //$url=$row['url'];
}

$sql="SELECT * FROM `worker` WHERE `id`='$id_worker'";
$result = mysqli_query($link, $sql);
while($row=mysqli_fetch_array($result)) {

  $name_worker=$row['name'];
  $tel=$row['tel'];
  $email=$row['email'];
}

$sql="SELECT * FROM `geo` WHERE `id`='$id_geo'";
$result = mysqli_query($link, $sql);
while($row=mysqli_fetch_array($result)) {

  $city=$row['city'];
}
include('header.php'); ?>

<div class="r25"></div>

<div class="container">
  <div class="row" style="border-radius:20px; background:#E9E9E9;">
    <div class="col-md-2 col-12"></div>
      <div class="col-md-8 col-12">
        <div class="r50"> </div>
          <form class="d-flex" role="search">
            <input class="form-control me-2" style="border-radius:20px;" type="search" placeholder="Що шукаємо?" aria-label="Search">
            <button type="button" style="border-radius:20px;" class="btn btn-info">Знайти</button>
          </form>
        <div class="r50"> </div>
      </div>
    <div class="col-md-2 col-12"></div>
  </div>
</div>

<div class="r50"> </div>

<div class ="container">
  <div class="row">

    <div class="col-md-9 col-12">

      <div style="border-radius:20px;">

        <div class="row">
          <div class="col-9">
            <h4><?php echo $name; ?></h4>
          </div>
          <div class="col-3">
            <h4><?php echo $salary; ?> грн</h4>
          </div>
        </div>

        <div class="r10"></div>

        <div class="row">
```

Рисунок 3.13 – Вміст файлу advertisement.php

```
<div class="row">
  <div class="col-4">
    <p><?php echo $name_worker; ?></p>
  </div>
  <div class="col-4">
    <p><?php echo $city; ?></p>
  </div>
  <div class="col-4">
    <p><?php echo $work_schedule; ?></p>
  </div>
</div>

<div class="r10"></div>

<p><?php echo $summary; ?></p>

<div class="r30"></div>

<a href="#info" data-bs-toggle="modal" data-bs-target="#info"><button type="button"
style="border-radius:20px;" class="btn btn-info">Відгукнутись</button></a>
<a href="/exit"><button type="button" style="border-radius:20px; margin-left:25px;" class="btn btn-danger">Назад</button></a>
<div class="r30"></div>

</div>

<div class="r50"></div>
<div class="r50"></div>

<h2>Схожі оголошення</h2>

<div class="r50"></div>

<div style="border-radius:20px; background:#E9E9E9; padding: 57px 57px 57px 57px">

<div class="row">
  <div class="col-9">
    <a href=""><h4>Веб розробник</h4></a>
  </div>
  <div class="col-3">
    <h4>~40 000 грн</h4>
  </div>
</div>

<div class="r10"></div>

<div class="row">
  <div class="col-4">
    <p>Віддалено/Повна зайнятість</p>
  </div>
  <div class="col-4">

  </div>
  <div class="col-4">

  </div>
</div>

<div class="r10"></div>

<p>В пошуку співробітника у нашу команду.</p>

<div class="r30"></div>

<button type="button" style="border-radius:20px;" class="btn btn-info">Відгукнутись</button>

<div class="r30"></div>

</div>

</div>

<div class="modal fade" id="info" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">
  <div class="modal-content">
    <div class="modal-header">
      <h1 class="modal-title fs-5" id="exampleModalLabel">Інформація про фрилансера</h1>
      <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
    </div>
    <div class="modal-body">
      <form method="POST">
```

Рисунок 3.14 – Продовження вмісту файлу advertisement.php

```
<form method="POST">

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Назва компанії/або Ім'я, Фамілія</label>
  <h6><?php echo $name_worker; ?></h6>
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Email</label>
  <h6><?php echo $email; ?></h6>
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Телефон</label>
  <h6><?php echo $tel; ?></h6>
</div>

<div class="r20"></div>

<input type="submit" value="OK">
</form>
</div>
</div>
</div>
</div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
<?php include('style.php');?>
</body>
</html>
```

Рисунок 3.15 – Кінець вмісту файлу advertisement.php

Перейдемо до файлу new_vacancy.php. Доступ має лише роль замовника. Це сторінка для створення нової вакансії, яка буде видна фрілансерам. На рисунках 3.16 – 3.17 зображений вміст файлу new_vacancy.php.

```
<?php

if((isset($_SESSION['bd'])) & (isset($_SESSION['id'])) &($_SESSION['bd']=='company')){
  $bd='company';

  include('header.php');


}
else {header('Location:http://yo.loc'); exit();}

?>

<?php

$link = new mysqli('localhost', 'root', '', 'youroffer_bd');

if(isset($_POST['name'])) {

  $name=$_POST['name'];
  $summary=$_POST['summary'];
  $exp=$_POST['exp'];
  $salary=$_POST['salary'];
  $work_schedule=$_POST['work_schedule'];
  $id_profession=$_POST['id_profession'];
  $id_geo=$_POST['id_geo'];

  $id_company=$_SESSION['id'];

  $sql="INSERT INTO `vacancy`(`id_company`, `id_profession`, `name`, `summary`, `exp`, `salary`, `work_schedule`, `id_geo`
  VALUES ('$id_company', '$id_profession', '$name', '$summary', '$exp', '$salary', '$work_schedule', '$id_geo')";
  $result = mysqli_query($link, $sql);
}
?>
```

Рисунок 3.16 – Вміст файлу new_vacancy.php

```
<div class="containder">
  <center>
    <h2>Додати нову вакансію</h2>
  </center>
</div>

<div class="r25"></div>



<div class ="container">
  <form method="POST">

    <div class="r10"></div>

    <h6>Спеціалізація</h6>

    <div class="r10"></div>

    <select name="id_profession" class="form-select" aria-label="Default select example">
      <option value="1">Frontend</option>
      <option value="2">Backend</option>
      <option value="3">Fullstack</option>
      <option value="3">SMM</option>
    </select>

    <div class="r10"></div>

    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">Назва</label>
      <input name="name" type="text" class="form-control" id="exampleFormControlInput1" placeholder="Fullstack dev">
    </div>

    <div class="r10"></div>

    <div class="mb-3">
      <label for="exampleFormControlTextarea1" class="form-label">Опис</label>
      <textarea name="summary" class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
    </div>

    <div class="r10"></div>

    <select name="exp" class="form-select" aria-label="Default select example">
    <option selected>Досвід роботи</option>
    <option value="0">Не важливо</option>
    <option value="1">1-2 роки</option>
    <option value="4">3+ роки</option>
    </select>

    <div class="r10"></div>

    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">Оплата(грн)</label>
      <input name="salary" type="text" class="form-control" id="exampleFormControlInput1" placeholder="40 000">
    </div>

    <div class="r10"></div>

    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">Графік роботи</label>
      <input name="work_schedule" type="text" class="form-control" id="exampleFormControlInput1" placeholder="Повний робочий день">
    </div>

    <div class="r10"></div>

    <h6>Місто</h6>

    <div class="r10"></div>

    <select name="id_geo" class="form-select" aria-label="Default select example">
    <option value="1">Київ</option>
      <option value="2">Харків</option>
      <option value="3">Дніпро</option>
      <option value="4">Запоріжжя</option>
      <option value="5">Львів</option>
    </select>

    <div class="r20"></div>

    <input type="submit" class= "btn btn-danger" value="Додати"></input>

  </form>


    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
    <?php include('style.php');?>
</div>
</body>
</html>
```

Рисунок 3.17 – Кінець вмісту файлу new_vacancy.php

Перейдемо до папки worker. До файлів в ній будуть мати доступ лише користувачі з активною роллю фрілансер(виконавець). Створимо два файли – new_advertisement.php та vacancy.php.

У файлі new_advertisement.php пропишемо логіку створення нового оголошення від фрилансера. На рисунку 3.18 – 3.19 зображено вміст файлу new_advertisement.php.

```php
<?php

if((isset($_SESSION['bd'])) & (isset($_SESSION['id'])) &($_SESSION['bd']=='worker')){
  $bd='worker';
  include('header.php');
}
else {header('Location:http://yo.loc'); exit();}

$link = new mysqli('localhost', 'root', '', 'youroffer_bd');

if(isset($_POST['name']))  {

  $name=$_POST['name'];
  $summary=$_POST['summary'];
  $exp=$_POST['exp'];
  $salary=$_POST['salary'];
  $work_schedule=$_POST['work_schedule'];
  $id_profession=$_POST['id_profession'];
  $id_geo=$_POST['id_geo'];

  $id_worker=$_SESSION['id'];

  $sql="INSERT INTO `resume`(`id_worker`, `id_profession`, `name`, `summary`, `exp`, `salary`, `work_schedule`, `id_geo`)
  VALUES ('$id_worker', '$id_profession', '$name', '$summary', '$exp', '$salary', '$work_schedule', '$id_geo')";
  $result = mysqli_query($link, $sql);
}
?>


<div class="containder">
  <center>
    <h2>Додати нове оголошення</h2>
  </center>
</div>

<div class="r25"></div>

<div class ="container">
  <form method="POST">

    <div class="r10"></div>

    <h6>Спеціалізація</h6>

    <div class="r10"></div>

    <select name="id_profession" class="form-select" aria-label="Default select example">
      <option value="1">Frontend</option>
      <option value="2">Backend</option>
      <option value="3">Fullstack</option>
      <option value="4">SMM</option>
    </select>

    <div class="r10"></div>

    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">Назва</label>
      <input name="name" type="text" class="form-control" id="exampleFormControlInput1" placeholder="Fullstack dev">
    </div>

    <div class="r10"></div>

    <div class="mb-3">
      <label for="exampleFormControlTextarea1" class="form-label">Опис</label>
      <textarea name="summary" class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
    </div>

    <div class="r10"></div>

    <select name="exp" class="form-select" aria-label="Default select example">
    <option selected>Досвід роботи</option>
    <option value="0">Не важливо</option>
    <option value="1">1-2 роки</option>
    <option value="4">3+ роки</option>
    </select>

    <div class="r10"></div>
```

Рисунок 3.18 – Вміст файлу new_advertisement.php

```
    <div class="r10"></div>

    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">Оплата(грн)</label>
      <input name="salary" type="text" class="form-control" id="exampleFormControlInput1" placeholder="40 000">
    </div>

    <div class="r10"></div>

    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">Графік роботи</label>
      <input name="work_schedule" type="text" class="form-control" id="exampleFormControlInput1" placeholder="Повний робочий день">
    </div>

    <div class="r10"></div>

    <h6>Місто</h6>

    <div class="r10"></div>

    <select name="id_geo" class="form-select" aria-label="Default select example">
      <option value="1">Київ</option>
      <option value="2">Харків</option>
      <option value="3">Дніпро</option>
      <option value="4">Запоріжжя</option>
      <option value="5">Львів</option>
    </select>

    <div class="r20"></div>

    <input type="submit" class= "btn btn-danger" value="Додати"></input>

  </form>


    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
    <?php include('style.php');?>
</div>
```

Рисунок 3.19 – Кінець вмісту файлу new_advertisement.php

Перейдемо до файлу vacancy.php. Доступ до нього мають лише користувачі з активною сесією фрилансера. Пропишемо в ньому логіку відображення вакансій від замовників, основною інформацією про них, можливістю перейти за посиланням на детальну сторінку з карткою вакансії та перегляду даних автора вакансії. На малюнку 3.20 – 3.22 зображений вміст файлу vacancy.php.

```
<?php

if((isset($_SESSION['bd'])) & (isset($_SESSION['id'])) &($_SESSION['bd']=='worker')){
  $bd='worker';
}

else {header('Location:http://yo.loc'); exit();}

$id=$_GET['id'];
$sql="SELECT * FROM `vacancy` WHERE `id`='$id'";
$result = mysqli_query($link, $sql);
while($row=mysqli_fetch_array($result)) {
```

Рисунок 3.20 – Вміст файлу vacancy.php

```php
while($row=mysqli_fetch_array($result)) {

  $id_company=$row['id_company'];
  $id_profession=$row['id_profession'];
  $name=$row['name'];
  $summary=$row['summary'];
  $exp=$row['exp'];
  $salary=$row['salary'];
  $work_schedule=$row['work_schedule'];
  $id_geo=$row['id_geo'];
  //$url=$row['url'];
}
$sql="SELECT * FROM `company` WHERE `id`='$id_company'";
$result = mysqli_query($link, $sql);
while($row=mysqli_fetch_array($result)) {

  $name_company=$row['name'];
  $tel=$row['tel'];
  $email=$row['email'];
}

$sql="SELECT * FROM `geo` WHERE `id`='$id_geo'";
$result = mysqli_query($link, $sql);
while($row=mysqli_fetch_array($result)) {

  $city=$row['city'];
}

include('header.php'); ?>

<div class="r25"></div>

<div class="container">
  <div class="row" style="border-radius:20px; background:#E9E9E9;">
    <div class="col-md-2 col-12"></div>
      <div class="col-md-8 col-12">
        <div class="r50"> </div>
          <form class="d-flex" role="search">
            <input class="form-control me-2" style="border-radius:20px;" type="search" placeholder="Що шукаємо?" aria-label="Search">
            <button type="button" style="border-radius:20px;" class="btn btn-info">Знайти</button>
          </form>
        <div class="r50"> </div>
      </div>
    <div class="col-md-2 col-12"></div>
  </div>
</div>

<div class="r50"> </div>

<div class ="container">
  <div class="row">

    <div class="col-md-9 col-12">

      <div style="border-radius:20px;">

        <div class="row">
          <div class="col-9">
            <h4><?php echo $name; ?></h4>
          </div>
          <div class="col-3">
            <h4><?php echo $salary; ?> грн</h4>
          </div>
        </div>

        <div class="r10"></div>

        <div class="row">
          <div class="col-4">
            <p><?php echo $name_company; ?></p>
          </div>
          <div class="col-4">
            <p><?php echo $city; ?></p>
          </div>
          <div class="col-4">
            <p><?php echo $work_schedule; ?></p>
          </div>
        </div>

        <div class="r10"></div>

        <p><?php echo $summary; ?></p>

        <div class="r30"></div>

        <a href="#info" data-bs-toggle="modal" data-bs-target="#info"><button type="button" style="border-radius:20px;" class="btn btn-info">Відгукнутись</button
        <a href="/exit"><button type="button" style="border-radius:20px; margin-left:25px;" class="btn btn-danger">Назад</button></a>
        <div class="r30"></div>

      </div>



      <div class="r50"></div>
      <div class="r50"></div>

      <h2>Схожі вакансії</h2>

      <div class="r50"></div>
```

Рисунок 3.21 – Продовження вмісту файлу vacancy.php

```
114         <div style="border-radius:20px; background:#E9E9E9; padding: 57px 57px 57px 57px">
115
116           <div class="row">
117             <div class="col-9">
118               <a href=""><h4>Веб розробник</h4></a>
119             </div>
120             <div class="col-3">
121               <h4>~40 000 грн</h4>
122             </div>
123           </div>
124
125           <div class="r10"></div>
126
127           <div class="row">
128             <div class="col-4">
129               <p>Віддалено/Повна зайнятість</p>
130             </div>
131             <div class="col-4">
132
133             </div>
134             <div class="col-4">
135
136             </div>
137           </div>
138
139           <div class="r10"></div>
140
141           <p>В пошуку співробітника у нашу команду.</p>
142
143           <div class="r30"></div>
144
145           <button type="button" style="border-radius:20px;" class="btn btn-info">Відгукнутись</button>
146
147           <div class="r30"></div>
148
149         </div>
150       </div>
151       <div class="modal fade" id="info" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
152     <div class="modal-dialog">
153       <div class="modal-content">
154         <div class="modal-header">
155           <h1 class="modal-title fs-5" id="exampleModalLabel">Інформація про компанію</h1>
156           <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
157         </div>
158         <div class="modal-body">
159           <form method="POST">
160
161           <div class="mb-3">
162             <label for="exampleFormControlInput1" class="form-label">Назва компанії/або Ім'я, Фамілія</label>
163             <h6><?php echo $name_company; ?></h6>
164           </div>
165
166           <div class="mb-3">
167             <label for="exampleFormControlInput1" class="form-label">Email</label>
168             <h6><?php echo $email; ?></h6>
169           </div>
170
171           <div class="mb-3">
172             <label for="exampleFormControlInput1" class="form-label">Телефон</label>
173             <h6><?php echo $tel; ?></h6>
174           </div>
175
176           <div class="r20"></div>
177
178           <input type="submit" value="OK">
179           </form>
180         </div>
181       </div>
182     </div>
183   </div>
184
185     </div>
186   </div>
187
188     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
189     integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
190     <?php include('style.php');?>
191   </body>
192 </html>
```

Рисунок 3.22 – Кінець вмісту файлу vacancy.php

3.2 Опис функціоналу та взаємодія з розробленим сайтом

У будь-якому випадку нас переадресовує без активної сесії на головну сторінку сайту. Розроблений сайт вимагає з самого початку, щоб ми обрали роль або замовника або фрилансера, тож розглянемо взаємодію з ним з двох сторін. Без реєстрації/авторизації в одній з ролей нас просто нікуди не пропустить далі.

3.2.1 В ролі замовника

Нас зустрічає головна сторінка сайту. Без реєстрації/авторизації більшість функцій і переадресацій на інші сторінки недоступні. Активні лише кнопки в шапці – Замовник, а після Реєстрація і Авторизація. Натиснемо спочатку на Замовник(кнопка Фрилансер зникне) а потім на реєстрацію або авторизацію (див. рис. 3.23).

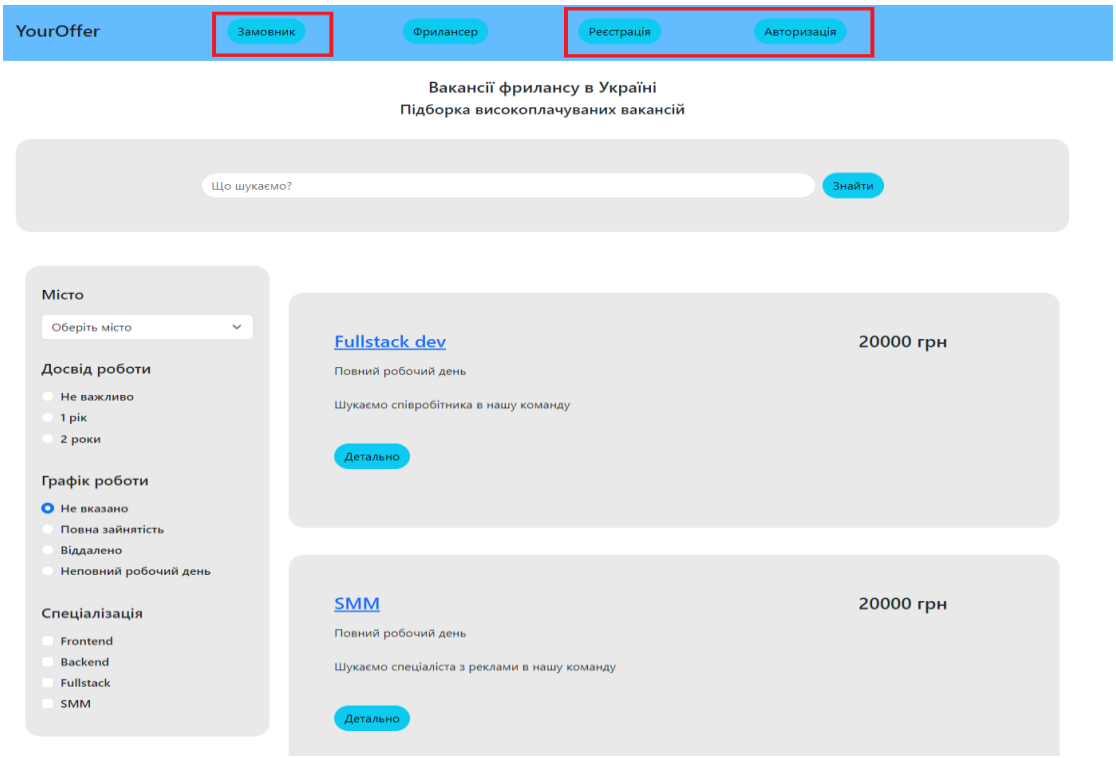


Рисунок 3.23 – Неавторизована головна сторінка в ролі замовника

Далі нас зустріне форма реєстрації/авторизації яка потребує інформації від користувача для початку активної сесії (див. рис. 3.24). Введені дані перевіряються на валідність нативними інструментами PHP. Якщо дані не валідні то сесія не почнеться.

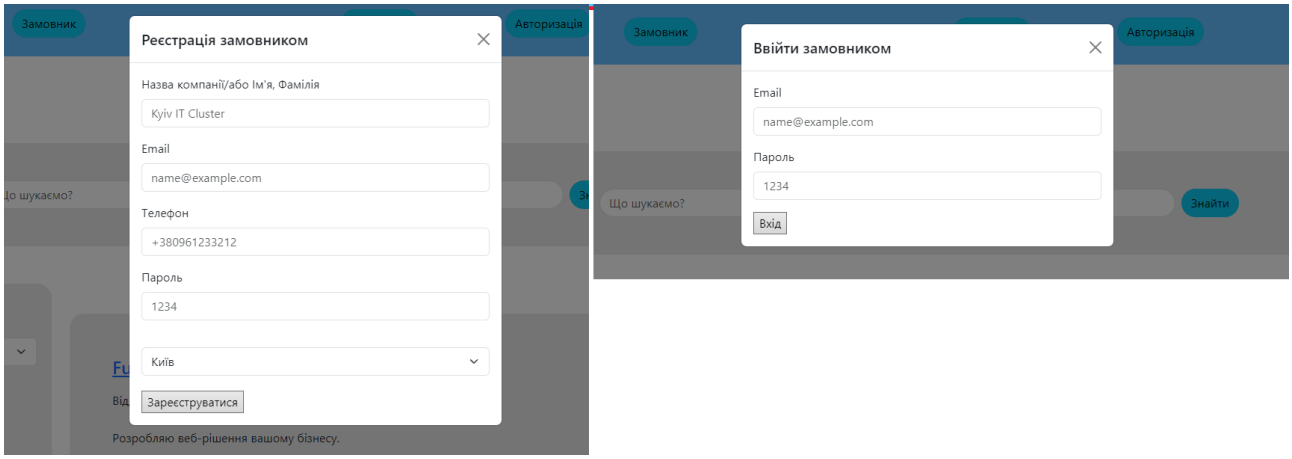


Рисунок 3.24 – Сторінка реєстрації/авторизації замовником

Після того як сесія була активована і користувач отримав роль замовника, з'являться нові кнопки в шапці(Додати вакансію, Вийти), локація, яку обрав при реєстрації користувач, зміниться вітальний текст а також на сторінці з'явиться підборка оголошень від фрилансерів, яка вже буде для нас активна.Також, стане активним пошук за ключовими словами серед оголошень та фільтрація за критеріями (див. рис. 3.25).

Кнопка Вийти в шапці завершить активну сесію, і надасть нам можливість обрати іншу роль і іншу активну сесію.

Поглянемо на оголошення від фрилансерів. Ми бачимо картки оголошення з мінімальною інформацією про автора. Можемо натиснути на кнопку Детально під кожним з них або на саму назву (див. рис. 3.26). Це перекине нас на сторінку з детальною інформацією про конкретно це оголошення.

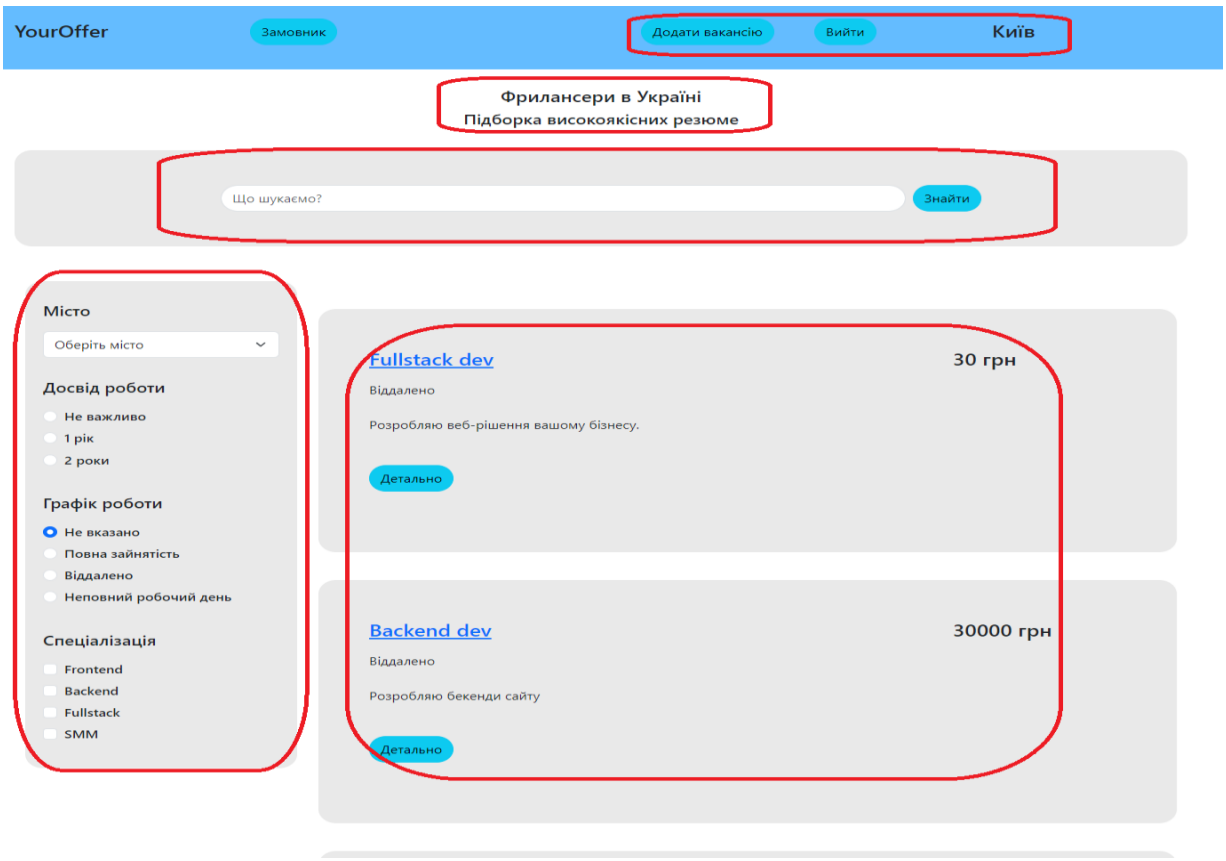


Рисунок 3.25 – Сторінка з активною сесією замовника

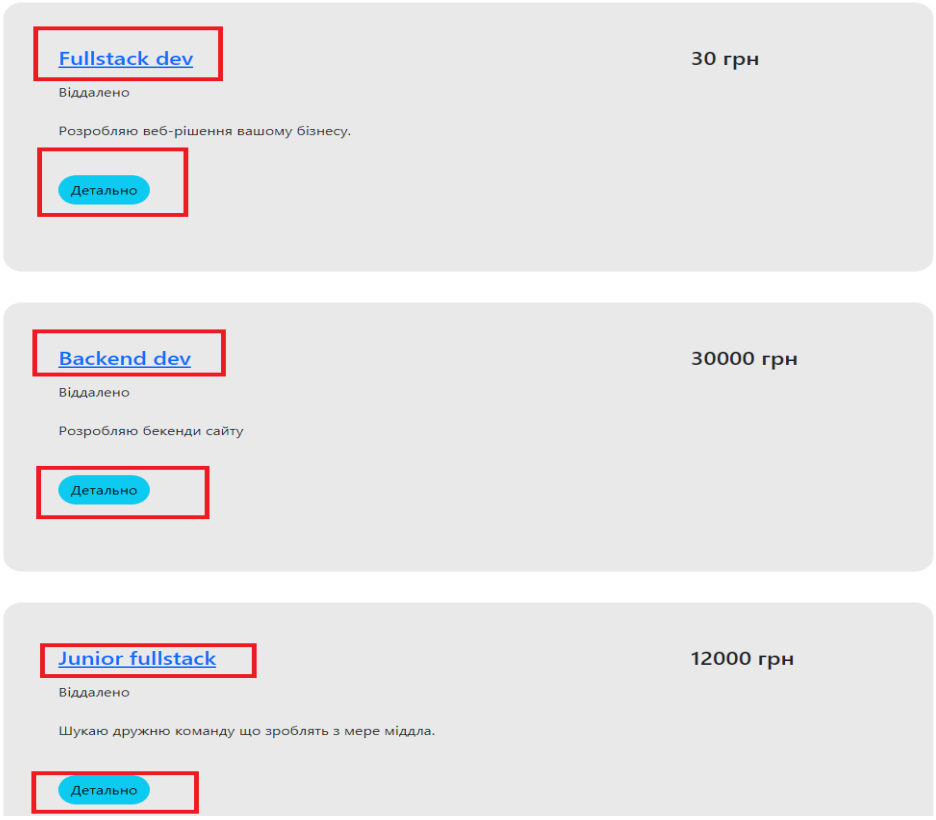


Рисунок 3.26 – Оголошення фрилансерів

Після натискання нас перекине на сторінку конкретно цього оголошення. До них мають доступ лише користувачі з активною сесією замовник. Можемо побачити більш детальну інформацію про оголошення, схожі оголошення, кнопку Назад, що поверне нас на головну сторінку та кнопку Відгукнутись (див. рис. 3.27). Кнопка Детально на схожому оголошенні перекине нас на сторінку з тим оголошенням.

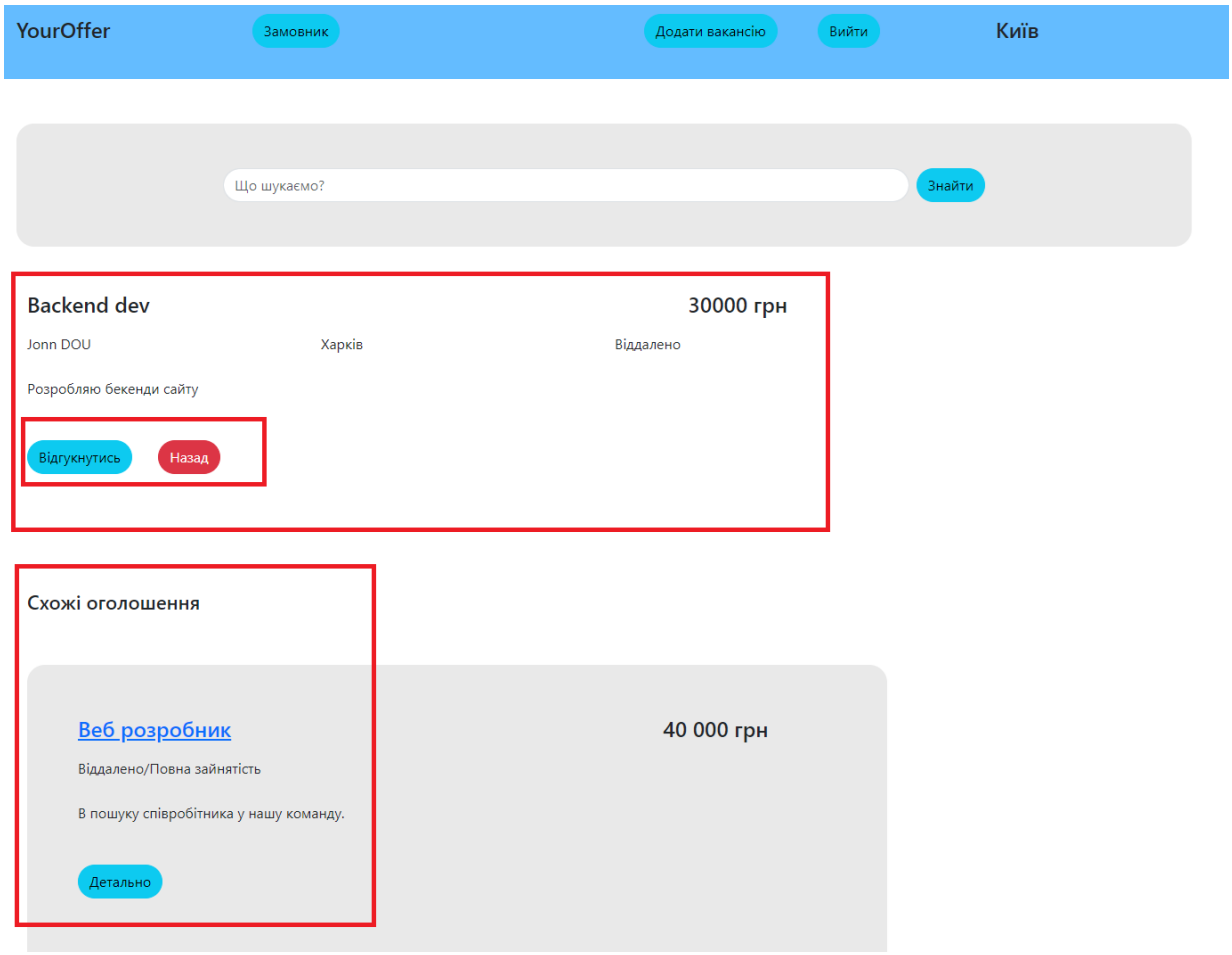


Рисунок 3.27 – Детальна сторінка оголошення

Після натискання на кнопку Відгукнутись відкриється форма з контактами вказаними автором оголошення (див. рис. 3.28). Після цього можемо натиснути ОК та на кнопку Назад для повернення на головну сторінку або Детально на схожому оголошенні для переходу на нього, або на Додати вакансію для створення нової вакансії для фрилансерів.

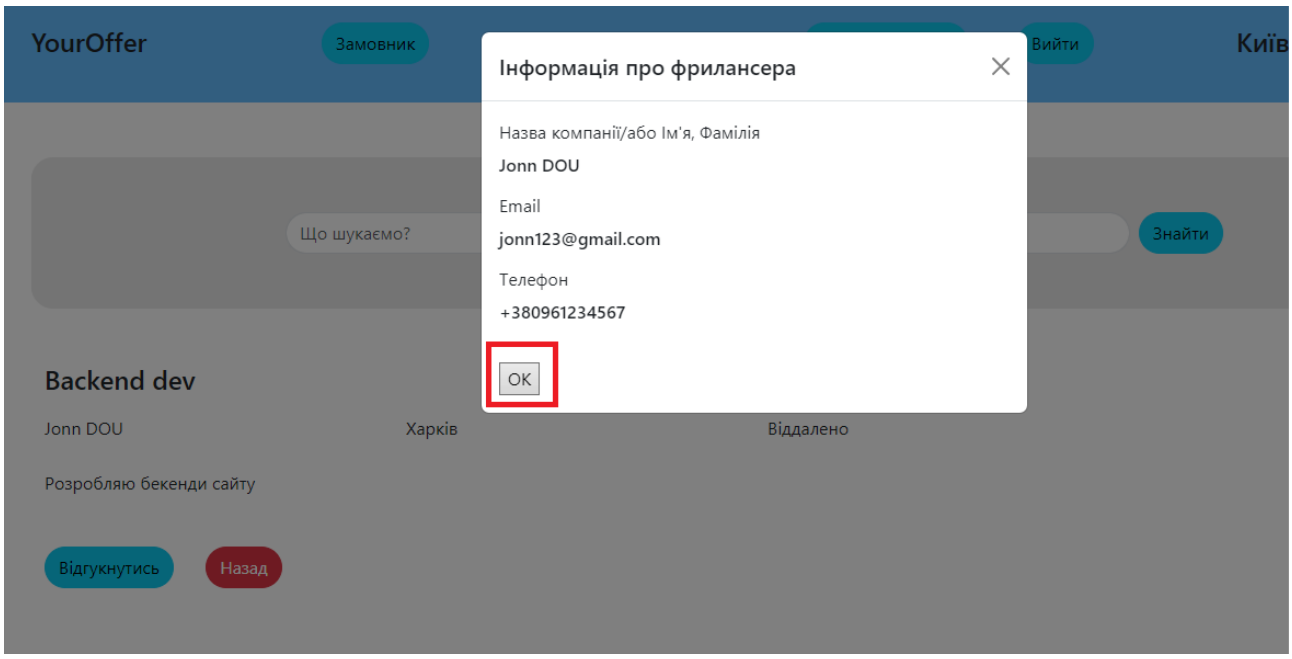


Рисунок 3.28 – Інформація про автора оголошення

Натиснемо на кнопку Додати вакансію в шапці. Це перекине нас на сторінку з створенням нової вакансії, доступної лише користувачу з активною сесією замовник. Нам відкриється сторінка з декількома формами, які необхідно заповнити інформацією про вакансію, кнопка Додати, що додає нову вакансію у випадку валідності даних, та можна через шапку профілю повернутися но головну (див. рис. 3.29).

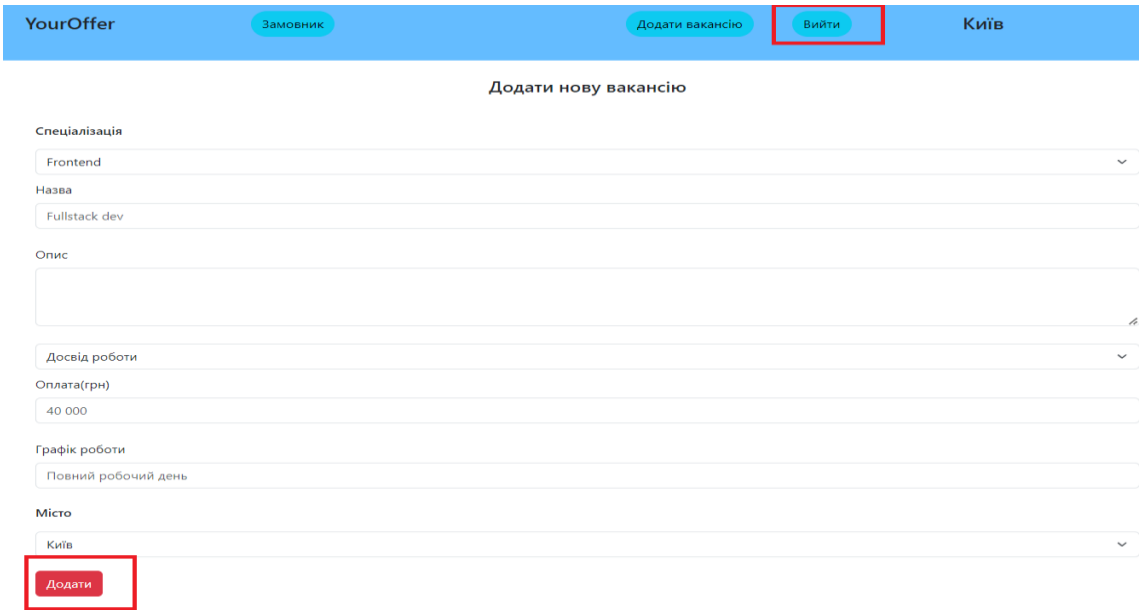


Рисунок 3.29 – Сторінка з створенням нової вакансії

3.2.2 В ролі фрилансера

Нас зустрічає головна сторінка сайту. Без реєстрації/авторизації більшість функцій і переадресацій на інші сторінки недоступні. Активні лише кнопки в шапці – Фрилансер, а після Реєстрація і Авторизація. Натиснемо спочатку на Фрилансер(кнопка Замовник зникне) а потім на реєстрацію або авторизацію (див. рис. 3.30).

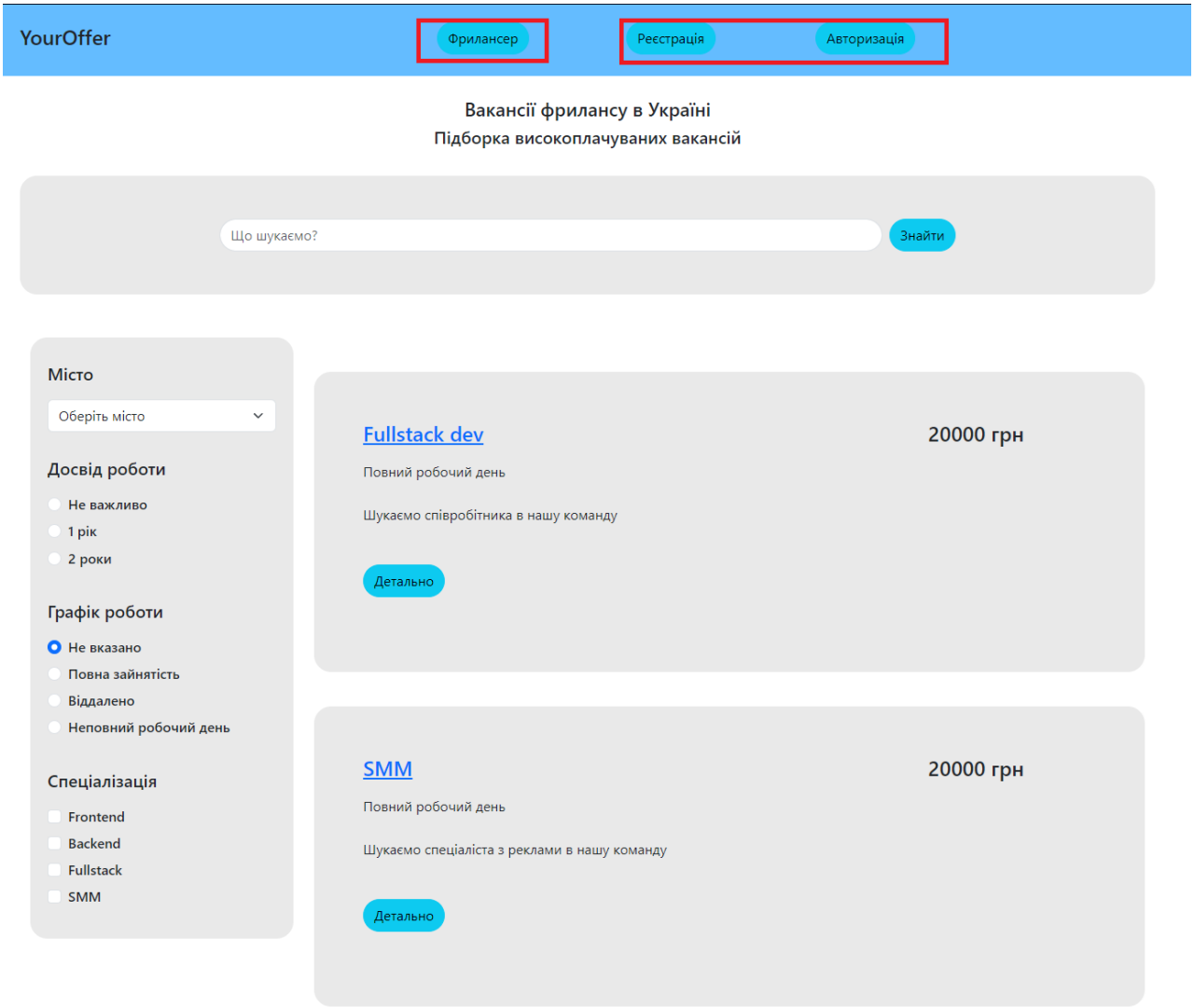


Рисунок 3.30 – Неавторизована головна сторінка в ролі фрилансера

Далі нас зустріне форма реєстрації/авторизації яка потребує інформації від користувача для початку активної сесії (див. рис. 3.31). Введені дані

перевіряються на валідність нативними інструментами PHP. Якщо дані не валідні то сесія не почнеться.

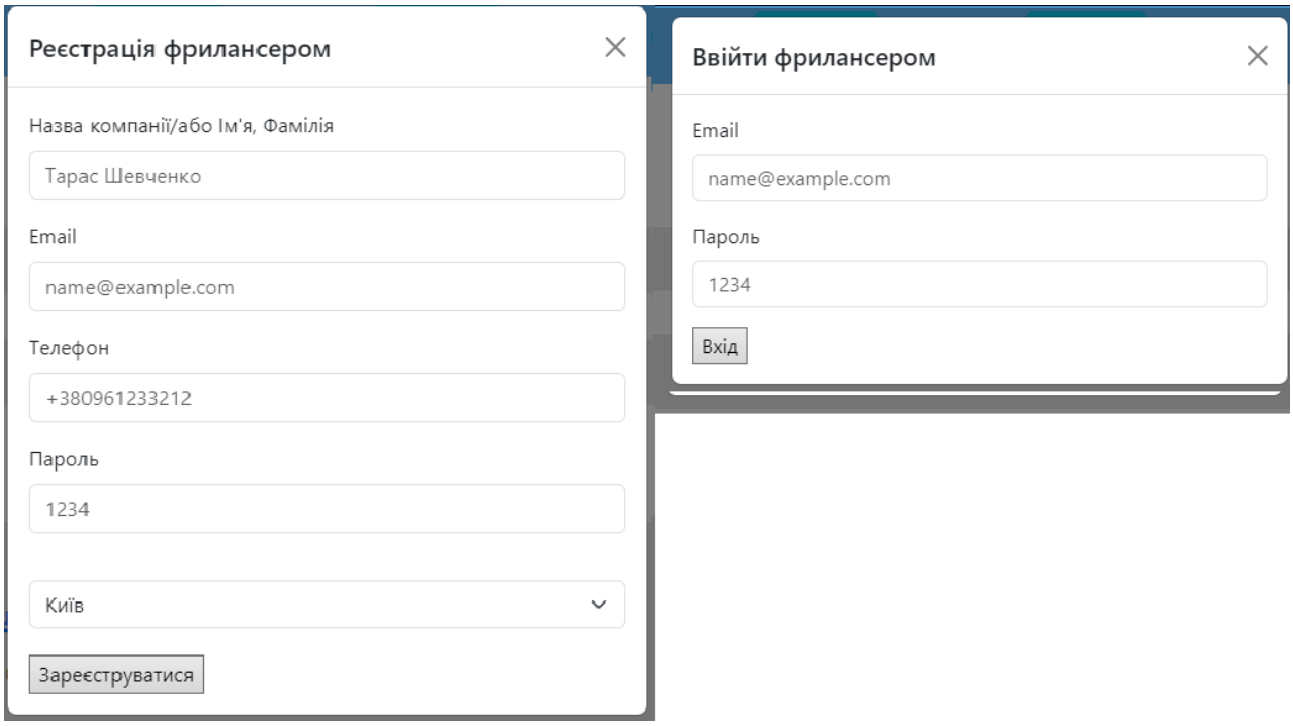


Рисунок 3.31 – Сторінка реєстрації/авторизації фрилансером

Після того як сесія була активована і користувач отримав роль фрилансера, з’являться нові кнопки в шапці(Додати оголошення, Вийти), локація, яку обрав при реєстрації користувач, зміниться вітальний текст а також на сторінці з’явиться підборка вакансій від замовників, яка вже буде для нас активна.Також, стане активним пошук за ключовими словами серед вакансій та фільтрація за критеріями (див. рис. 3.32).

Кнопка Вийти в шапці завершить активну сесію, і надасть нам можливість обрати іншу роль і іншу активну сесію.

Поглянемо на вакансії від замовників. Ми бачимо картки вакансії з мінімальною інформацією про автора. Можемо натиснути на кнопку Детально під кожним з них або на саму назву (див. рис. 3.33). Це перекине нас на сторінку з детальною інформацією про конкретно цю вакансію.

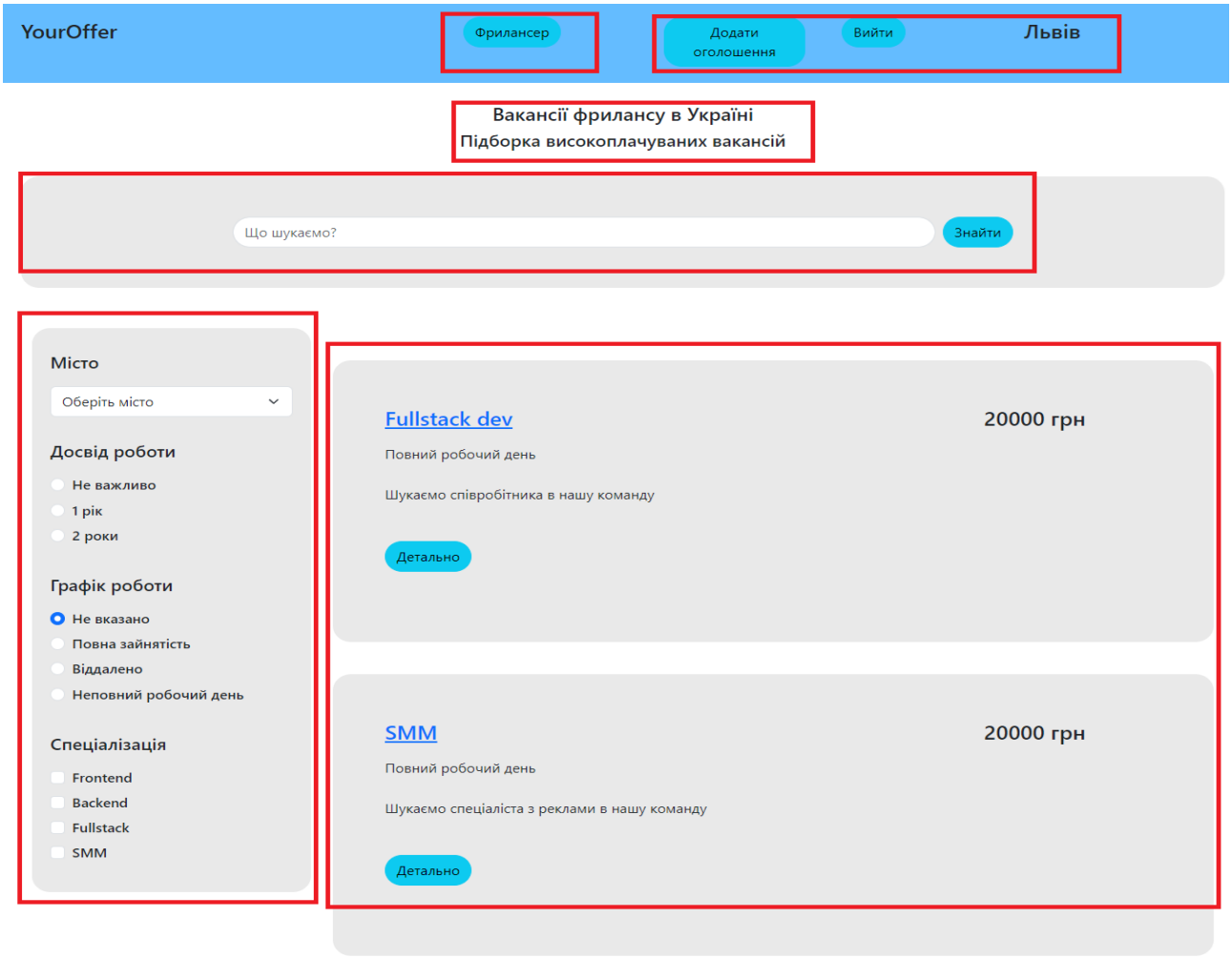


Рисунок 3.32 – Сторінка з активною сесією фрилансера



Рисунок 3.33 – Вакансії замовників

Після натискання нас перекине на сторінку конкретно цієї вакансії. До них мають доступ лише користувачі з активною сесією фрилансер. Можемо побачити більш детальну інформацію про вакансію, схожі оголошення, кнопку Назад, що поверне нас на головну сторінку та кнопку Відгукнутись (див. рис. 3.34). Кнопка Детально на схожій вакансії перекине нас на сторінку з тою вакансією.

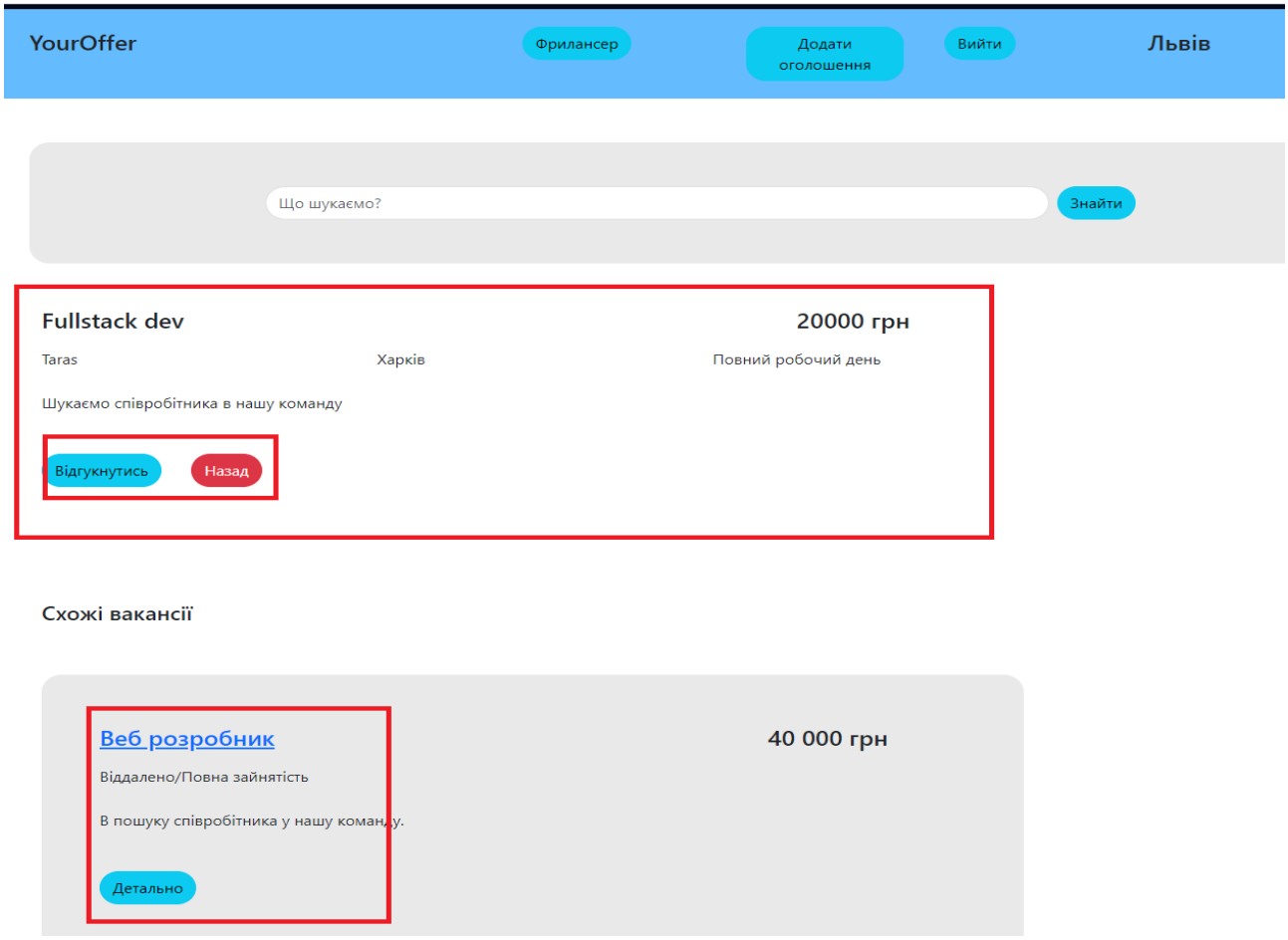


Рисунок 3.34 – Детальна сторінка вакансії

Після натискання на кнопку Відгукнутись відкриється форма з контактами вказаними автором вакансії (див. рис. 3.35). Після цього можемо натиснути ОК та на кнопку Назад для повернення на головну сторінку або Детально на схожій вакансії для переходу на неї, або на Додати оголошення для створення нового оголошення, що буде відображатись замовникам.

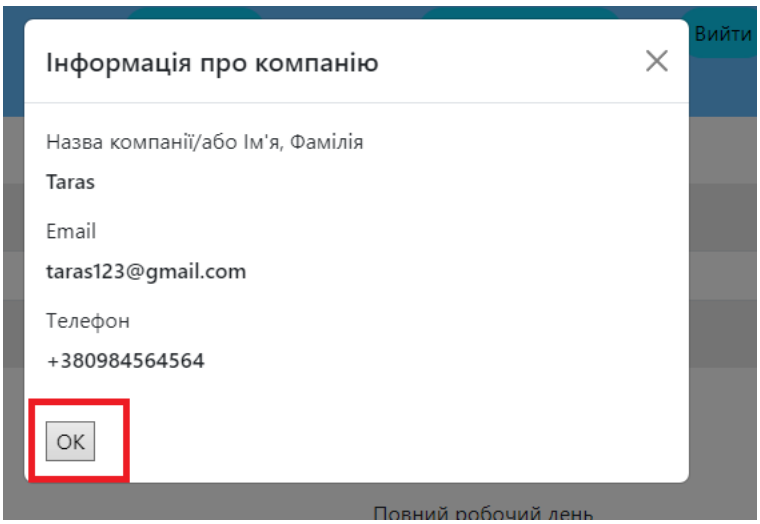


Рисунок 3.35 – Інформація про автора вакансії

Натиснемо на кнопку Додати оголошення в шапці. Це перекине нас на сторінку з створенням нового оголошення, доступної лише користувачу з активною сесією фрилансер. Нам відкриється сторінка з декількома формами, які необхідно заповнити інформацією про оголошення, кнопка Додати, що додає нове оголошення у випадку валідності даних, та можна через шапку профілю повернутися но головну (див. рис. 3.36).



Рисунок 3.36 – Сторінка з створенням нового оголошення

ВИСНОВКИ

В ході виконання кваліфікаційної роботи на основі зібраної інформації було спроектовано і розроблено веб-сайт фриланс-біржі з дотриманням основних вимог до цього проекту.

Було реалізовано функціонал реєстрації, авторизації, поділ на замовника і фрилансера, створення оголошення/вакансії, перегляд існуючих оголошень/вакансій, пошук за ключовими словами, сортування, вихід з сесії, перегляд обраної оголошення/вакансії більш детально з можливістю дізнатися контакти автора.

Також було приділено уваги до того куди намагається дібратися користувач, та перекриті додатковими умовами перевірок можливі переходи за точними посиланнями без реєстрації та відповідності певній категорії юзерів.

Приділено увагу і до того що буде намагатися вводити користувач у базу даних – за допомогою нативних інструментів HTML та PHP розроблено примітивну систему фільтрації введених даних.

У подальшому цей проект може бути використаний в якості прикладу для розробки схожих завдань або як основа для комерційного проекту вже з більшим функціоналом.

ПЕРЕЛІК ПОСИЛАНЬ

1. Microsoft. Official site. URL : https://uk.wikipedia.org/wiki/Microsoft (дата звернення 26.11.2024).

2. Britannica Money. Official site. URL : https://www.britannica.com/money/Bill-Gates (дата звернення 26.11.2024).

3. Kabanchik.ua. Official site. URL : https://kabanchik.ua (дата звернення 26.11.2024).

4. Freelancehunt. Official site. URL : https://freelancehunt.com/ua (дата звернення 26.11.2024).

5. Freelance.ua. Official site. URL : https://freelance.ua (дата звернення 26.11.2024).

6. PHP. Official site. URL : https://uk.wikipedia.org/wiki/PHP (дата звернення 26.11.2024).

7. AWS Amazon. Official site. URL : https://aws.amazon.com/what-is/sql/?nc1=h_ls (дата звернення 26.11.2024).

8. HTML. Official site. URL : https://html-css.co.ua (дата звернення 26.11.2024).

9. JavaScript. Official site. URL : https://uk.wikipedia.org/wiki/JavaScript (дата звернення 26.11.2024).

10. PhpMyAdmin. Official site. URL : https://www.phpmyadmin.net (дата звернення 26.11.2024).

11. MySQL. Official site. URL : https://www.mysql.com (дата звернення 26.11.2024).

12. Xampp. Official site. URL : https://www.apachefriends.org/index.html (дата звернення 26.11.2024).

13. VS Code. Official site. URL : https://code.visualstudio.com (дата звернення 26.11.2024).

14. Figma. Official site. URL : https://www.figma.com/ (дата звернення 26.11.2024).

15. Мартін Р. С. Чистий код. Київ : Фабула, 2019. 368 с.

16. Сьерра К., Бейтс Б. Head First. Київ : Фабула, 2022. 720 с.

17. Суркова Н. Е., Остроух А. В. Методологія структурного проєктування інформаційних систем. Харків : Науково-інноваційний центр, 2014. 184 с.

18. Norhaidah A. H., Nurdatillah H. PHP Frameworks Usability in Web Application Development. International Journal of Recent Technology and Engineering (IJRTE). 2019. Vol. 8, no. 3S. P. 109–116.

19. Prokofyevaa N., Boltunovaa V. Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. Procedia Computer Science : Conference, Riga, 15 December 2017. Riga, 2017. P. 51–56.

20. Ali J. Mastering PHP Design Patterns. Packt Publishing – ebooks Account, 2016. 270 p.