

**Детектор Плагіату v. 2941 - Звіт оригінальності: 06.01.2026 22:06:19**

Перевірений документ: Lanovenko V M!!!.docx Ліцензія: ВОЛОДИМИР МАТІЄВСЬКИЙ

? Тип пошуку: Пошук Перепису ? Знайдено мову: Uk

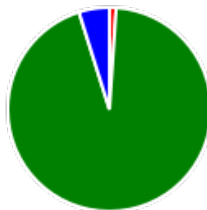
? Тип перевірки: Інтернет-перевірка

ТЕЕ і кодування: DocX n/a

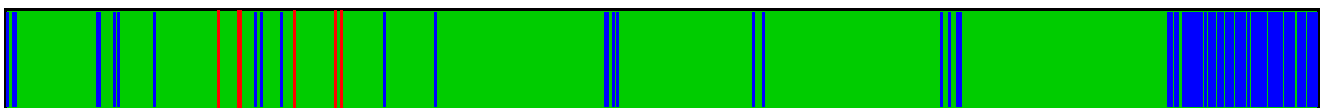
Детальний аналіз документа документа:

? Співвідношення:

Плагіат 1.06% Оригінал 94.1%  
Цитати 4.84% ШІ 0%



? Графік розподілу:







? Джерела плагіату: 4

- |        |       |     |  |
|--------|-------|-----|--|
| → 1%   | A B C | 111 | 1. <a href="https://mc.today/uk/najkrashhi-ukrayinski-ta-zarubizhni-birzhi-frilansu/">https://mc.today/uk/najkrashhi-ukrayinski-ta-zarubizhni-birzhi-frilansu/</a>   |
| → 1%   | A B C | 88  | 2. <a href="https://uaspectr.com/2022/01/06/najkrashhi-birzhi-frilansu/">https://uaspectr.com/2022/01/06/najkrashhi-birzhi-frilansu/</a>                             |
| → 0,2% | A B C | 10  | 3. <a href="https://web-crawler.plagiarism-detector.com/get-doc-pt?did=_kiiP4y7-zZBkA">https://web-crawler.plagiarism-detector.com/get-doc-pt?did=_kiiP4y7-zZBkA</a> |

? Дані оброблених ресурсів: 166 - ОК / 13 - Помилок

? Важливі примітки:

Wikipedia:	Google Books:	Сервіси платних робіт:	Античіт:
			
[не знайдено]	[не знайдено]	[не знайдено]	<b>Знайдена протидія!</b>


? Звіт проти обману UACE:

- Статус: Аналізатор **Увімкнений** Нормалізатор **Увімкнений** схожість символів встановлено **100%**
- Виявлений відсоток забруднення UniCode: **16,9%** з обмеженням: 4%
- Відсоток невизнаних символів після нормалізації: **9,9%**
- Усі підозрілі символи будуть позначені фіолетовим кольором: **Abcd...**
- Знайдено невидимі символи: 0

Рекомендація з оцінки:

Аналізу цього звіту слід приділити особливу увагу! Підозрюється, що цей документ містить значну кількість символів, чужих мові документа. Це є прямим вказівкою на те, що автор документа використав спеціальне програмне забезпечення/онлайн -веб -сервіс, щоб ефективно заплутати текст, намагаючись уникнути потенційного виявлення плагіату. Наполегливо рекомендується ескалювати цю справу! У разі сумнівів зверніться до служби підтримки детектора плагіату!

Статистика алфавіту та аналіз символів:

 Активні посилання (URL-адреси, витягнуті з документа):

URL не знайдені

 Виключено:

URL не знайдені

 Включено:

URL не знайдені

Детальний аналіз документу:

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ ЗАКЛАД

Знайдено посилань: **0,07%**

id: 1

„ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА”

Навчально-науковий інститут математики та інформаційних технологій Кафедра математики та інформатики Дослідження вебтехнологій розробки пошукових систем кваліфікаційна робота здобувача вищої освіти другого (магістерського) рівня освітньої програми

Знайдено посилань: **0,02%**

id: 2

«Комп'ютерні науки»

за спеціальністю 122

Знайдено посилань: **0,05%**

id: 3

„ Комп'ютерні науки ”

Особистий підпис \_\_\_\_\_ Науковий керівник \_\_\_\_\_ Завідувач кафедри \_\_\_\_\_ Лубни – 2026 ЗМІСТ Стор. ВСТУП7 РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ10 1.1. Аналіз з предметної області11 1.2. Аналіз систем пошуку виконавців ІТ проєктів19 1.3. Висновки до розділу 122 РОЗДІЛ 2. АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ24 2.1. Аналіз програмних веб технологій для розробки пошукової інформаційної системи24 2.2. Аналіз засобів реалізації баз даних у веб-технологіях27 2.3. Мова розмітки вебдокументів [HTML](#)28 2.4. Каскадні таблиці стилів – [CSS](#)29 2.5. Динамічна мова [JavaScript](#)34 2.5.1. [jQuery](#)36 2.6. Висновки до розділу 137 РОЗДІЛ 3. Архітектура та опис розробки39 3.1. Вимоги до проєкту веборієнтованої ІС39 3.2. [ER](#)-діаграма41 3.3. Структура таблиць бази даних43 3.4. Опис користувальницької частини додатку49 3.4.1. Гість49 3.4.2. Адміністратор52 3.4.3. Роботодавець55 3.3.4. Робітник56 3.5. Висновки до розділу 358 ВИСНОВКИ59 ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ61 ДОДАТКИ64 Додаток А. Головна сторінка роботодавця64 Додаток Б. Код [employee\\_home.php](#)65 ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ [CGI](#) [Common Gateway Interface](#); [CMS](#) система управління вмістом; [CSS](#) [Cascade Style Sheets](#); [DDOS](#) [Distributed Denial-of- service attack](#); [DOM](#) [Document Object Model](#); [DTD](#) [Document Type Definition](#); [EDGE](#) [Enhanced Data rates for GSM Evolution](#); [ERM](#) модель зв'язку сутностей; [FDA](#) [Food and Drug Administration](#); [FSM](#) [Finite-state machine](#); [GPRS](#) [General Packet Radio Service](#); [HTML](#) [HTTP](#) [Hyper Text Markup Language](#); [HyperText Transfer Protocol](#); [ISO](#) [International Organization for Standardization](#); [JS](#) [JavaScript](#); [PHP](#) [Hypertext Preprocessor](#); [SEO](#) [Search Engine Optimization](#); [SGML](#) [SQL](#) [Standard Generalized Markup Language](#); [Structured Query Language](#); [UML](#) [Unified Modeling Language](#); ІС інтелектуальна система; ІТ ОС інформаційні технології; операційна система; ПЗ програмне забезпечення; ПК персональний комп'ютер. ВСТУП Воєнні події в Україні призвели до суттєвих обмежень у багатьох сферах життєдіяльності, кардинально трансформували світовий ринок праці. Значна кількість людей втратила постійну роботу, однак, прагнучи адаптуватися до нових умов, вони все активніше переходять до дистанційних форм зайнятості, де інтернет став невід'ємною складовою професійного життя. Для більшості таким виходом став фріланс. Сучасний технологічний простір динамічно розвивається: щодня зростає кількість запитів від підприємців і власників бізнесу на виконання різноманітних завдань, зокрема створення логотипів, сайтів, дизайну чи програмного забезпечення. Водночас пошук надійного й компетентного виконавця стає дедалі складнішим, особливо з огляду на проблему довіри між замовниками та виконавцями. Хоча нині існує чимало онлайн-сервісів, що сприяють встановленню вигідних партнерських відносин, в Україні цей напрям залишається недостатньо розвиненим, а значна частина іноземних платформ є заблокованою. Тому створення вітчизняного сучасного порталу для фрілансерів є надзвичайно актуальним завданням. Метою роботи є аналіз вебтехнологій і розробка пошукової інформаційної системи — порталу для фрілансерів. Для досягнення поставленої мети необхідно розв'язати такі завдання: – дослідити ринок подібних сервісів; – проаналізувати вимоги до системи; – спроектувати архітектуру програмного продукту; – реалізувати програмне забезпечення. Об'єктом дослідження є сервіси, що забезпечують самостійне управління робочим процесом фрілансера та взаємодію із замовниками. Предметом дослідження є вебсервіси-посередники між клієнтами та фрілансерами, які працюють у науковій,

технічній і творчій сферах. Методи дослідження включають: – теоретичні: аналіз науково-технічних джерел та інтернет-ресурсів; – емпіричні: дослідження підходів до оптимізації сучасних вебтехнологій у процесі створення пошукових інформаційних систем. Наукова новизна полягає у розробці пошукової інформаційної системи

” Знайдено посилань: **0,01%**

id: 4

«FreeLance».

Практичне значення роботи визначається створенням програмного продукту

” Знайдено посилань: **0,01%**

id: 5

«FreeLance»

із використанням [PHP](#), [MySQL](#), [HTML](#), [CSS](#), [JavaScript](#) та [jQuery](#). Розроблений вебсервіс стане в нагоді для комунікації між клієнтами та виконавцями. Результати дослідження також можуть бути застосовані під час вивчення принципів побудови багаторівневих вебсистем і серверних алгоритмів обробки даних. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг становить 70 сторінок, кількість використаних джерел — 34. У першому розділі подано аналіз сучасних вебдодатків для фрілансерів та визначено основні вимоги до створення нового вебзастосування

” Знайдено посилань: **0,01%**

id: 6

«FreeLance».

У другому розділі розглянуто сучасні технології веброзробки, охарактеризовано популярні інструменти та виявлено їх недоліки. Обґрунтовано вибір засобів для реалізації магістерського проекту. У третьому розділі описано процес програмної реалізації вебзастосування

” Знайдено посилань: **0,01%**

id: 7

«FreeLance»

та наведено основні етапи його створення. Додатки містять зображення головної сторінки системи та фрагменти програмного коду файлу [employee\\_home.php](#). РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ У сучасному світі дедалі більшої популярності набуває позаштатна форма зайнятості, відома як фріланс. Для багатьох вона стає не лише альтернативою традиційній роботі, а й першим кроком до створення власної справи. Серед основних переваг фрілансу — можливість працювати з будь-якого зручного місця, самостійно планувати робочий час і не залежати від жорсткого контролю керівництва. В Україні фріланс здебільшого асоціюється з ІТ-сферою та охоплює такі напрями, як проектування й адміністрування вебсайтів, наповнення їх контентом, веб- та графічний дизайн. Цей формат діяльності добре підходить і для початківців, які лише роблять перші кроки в галузі інформаційних технологій. Типова модель роботи фрілансера здається простою: вибір онлайн-платформи, подання заявок на проекти, виконання завдань і отримання оплати. Проте на практиці процес значно складніший — складно знайти вигідне замовлення та вирізнитися серед великої кількості конкурентів. Фріланс-біржі — це онлайн-сервіси, що забезпечують можливість дистанційної співпраці між замовниками та виконавцями, дозволяють розміщувати власні послуги, виконувати різнопланові завдання та знаходити потенційних клієнтів для реалізації проектів у різних сферах. Попри глобальне поширення фрілансу, в Україні інтерес до віддаленої роботи зазнавав коливальності. Після послаблення карантинних обмежень у травні спостерігалось тимчасове зниження пошукової активності за запитами

” Знайдено посилань: **0,01%**

id: 8

«фріланс»

i

” Знайдено посилань: **0,02%**

id: 9


«віддалена робота»,

яке згодом змінилося різким зростанням. Згідно зі статистикою платформи [Freelancehunt](#), у 2006 році в Україні було зареєстровано близько 1,2 тисячі фрілансерів, тоді як у 2021 році кількість користувачів уже сягнула одного мільйона. Після початку повномасштабної війни чисельність фрілансерів не зменшилася — навпаки, все більше людей почали обирати цей

формат роботи та реєструватися на спеціалізованих платформах. Це зумовлено тим, що на подібних платформах доволі складно отримувати високий дохід, а ключову роль у співпраці відіграє рівень довіри між замовником і виконавцем. Попри те, що портал виконує функцію посередника, він не завжди спроможний повністю убезпечити користувачів від шахрайства. В Україні також поширена думка, що стабільна робота в перевіреному колективі дозволяє швидше набути професійного досвіду, хоча саме фріланс дає змогу значно легше сформувати якісне портфоліо та напрацювати практичні навички.

1.1 Аналіз предметної області Портали є різновидом вебресурсів із розширеним функціоналом і зручною системою навігації. Такі сайти зазвичай орієнтовані на конкретну тематику та об'єднують спільноти користувачів зі схожими інтересами. Портал для фрілансерів — це вебсайт, що забезпечує можливість дистанційної співпраці для розробників, дизайнерів та інших спеціалістів, виконуючи роль посередника між виконавцем і клієнтом. Після проходження реєстрації користувач може розміщувати власні завдання або пропонувати послуги, які зберігаються в базі даних системи. Потенційний замовник, переглядаючи детальну інформацію про проєкт, має змогу ознайомитися з пропозиціями фрілансера та відомостями про нього, після чого зв'язатися з виконавцем безпосередньо через портал і погодити умови співпраці. Розробка подібних сайтів зазвичай здійснюється із застосуванням систем керування контентом. Великі портали використовують власні спеціалізовані рішення для управління та моніторингу процесів, тоді як для середніх і невеликих проєктів часто застосовуються безкоштовні платформи, зокрема [OpenCart](#), [Drupal](#) або [WordPress](#). Система управління сайтом може бути встановлена на хостинг самостійно, розроблена як приватне програмне забезпечення або використовуватися на основі тимчасових ліцензій. Виділяють два основні типи порталів для пошуку віддаленої роботи: сайти з вакансіями, де роботодавці публікують оголошення про набір співробітників для дистанційної зайнятості; фріланс-біржі, на яких спеціалісти працюють на себе, самостійно обираючи проєкти та регулюючи власний робочий час. Сьогодні існує чимало фріланс-платформ, які настільки міцно закріпилися на ринку, що автоматично з'являються серед перших результатів пошуку за запитами, пов'язаними з фрілансом. До українських сервісів такого типу належать, зокрема: [Freelance.ua](#), [Weblancer.net](#), [Free-lance.ua](#), [Kabanchik.ua](#). Водночас фахівцям не варто обмежуватися лише вітчизняними ресурсами, адже ефективний пошук замовлень передбачає використання й міжнародних платформ.

1.1.1 [Freelancehunt](#)

 **Знайдено плагіату: 0,12%** <https://uaspectr.com/2022/01/06/najk...> + 2 ресурсів! id: 10

Щодня на біржі з'являється понад тисячу активних завдань від замовників.

Перша фріланс-біржа в Україні була створена у 2003 році — нею став сервіс [Weblancer.net](#). Станом на сьогодні в Україні функціонує низка популярних платформ для віддаленої роботи [9]. [Freelancehunt](#) є однією з найбільших і найвідоміших фріланс-бірж в Україні. Платформа пропонує проєкти у різноманітних напрямках, зокрема:

- дизайн;
- розробка програмного забезпечення;
- системне адміністрування;
- бекенд-розробка;
- управління проєктами;
- маркетинг;
- переклад;
- створення та наповнення контенту;
- юридичні послуги;
- інші галузі. Щодня на платформі публікується понад тисячу активних завдань від замовників, що свідчить про її високу популярність і затребуваність серед користувачів. На рисунку 1.2 представлено стартову сторінку [Freelancehunt](#). Рисунок 1.2. Стартова сторінка [Freelancehunt](#) Механізм роботи фріланс-біржі є таким: замовник публікує проєкт із детальним описом завдання; фрілансери подають заявки, зазначаючи вартість, строки виконання та власні коментарі;

 **Знайдено плагіату: 0,35%** <https://mc.today/uk/najkrashhi-ukray...> + 2 ресурсів! id: 11

замовник обирає виконавця; фрілансер виконує поставлене завдання; замовник приймає результат роботи та здійснює оплату; після завершення проєкту сторони залишають взаємні відгуки. Фрілансери мають можливість подавати необмежену кількість заявок на різні проєкти. Платформа [Freelancehunt](#) утримує комісію за проведення угод, яка становить

9 %. Оплата комісії може здійснюватися як замовником, так і виконавцем, або розподілятися між ними порівну — по 5 % з кожної сторони. Згідно зі статистичними даними сервісу [Freelancehunt](#) [1], 83,3 % користувачів працюють удома, 7,2 % — в офісах, 5,5 % поєднують домашній і офісний формати роботи, 2,1 % виконують завдання в кафе, а 0,6 % — у коворкінгах. Окремою функцією платформи є розділ

” Знайдено посилань: **0,01%**

id: **12**

«Конкурси»,

у якому замовники можуть оголосити змагання з неймінгу, створення логотипів чи ілюстрацій. Після заповнення брифу та проходження модерації фрілансери надсилають власні варіанти робіт, а замовник обирає переможця. У 2021 році на платформі було запроваджено розділ

” Знайдено посилань: **0,01%**

id: **13**

«Вакансії»,

призначений для пошуку штатних працівників, а не для разових фріланс-проектів. Розміщення однієї вакансії коштує 99 гривень, при цьому вона залишається активною протягом одного місяця. 4 листопада 2021 року в Україні вперше відзначили День фрілансера з ініціативи сервісу [Freelancehunt](#), який був зареєстрований саме цього дня. 1.1.2 [Freelance.ua](#) Фріланс-біржа [Freelance.ua](#) розпочала свою діяльність в Україні у 2006 році [14]. На рисунку 1.3 зображено головну сторінку вебсайту. Рисунок 1.3. Стартова сторінка [freelance.ua](#) На платформі [Freelance.ua](#) користувачі можуть знаходити замовлення в різних напрямках, зокрема: – вебпрограмування; – дизайн вебсайтів; – верстка сторінок; – розробка логотипів; – копірайтинг; – відеомонтаж; – [SEO](#)-просування; – контекстна реклама та інші види діяльності. Для зручності пошуку вакансій сервіс пропонує декілька категорій: – разові проекти; – постійна віддалена робота; – робота в офісі. Усі фінансові операції на [Freelance.ua](#) здійснюються через систему

” Знайдено посилань: **0,02%**

id: **14**

«безпечна угода»,

комісія якої становить 7,5 % і сплачується замовником. Платформа підтримує два типи акаунтів: – базовий, що дозволяє подавати одну заявку на добу на проекти з бюджетом до 400 грн; – [Pro](#) — платний обліковий, який надає можливість необмежено подавати заявки та брати участь у проектах із бюджетом понад 400 грн. 1.1.3 [Weblancer.net](#) Сервіс [Weblancer.net](#) функціонує понад 18 років і об'єднує більше ніж 1,5 мільйона користувачів. На біржі представлені такі напрями, як веброботка, програмування програмного забезпечення, створення текстів і

🚫 Знайдено плагіату: **0,13%** <https://uaspectr.com/2022/01/06/najk...> + 2 [ресурсів!](#)

id: **15**

переклади, графіка та фотографія, поліграфія й айдентика, архітектура та інжиніринг, адміністрування


вебсайтів та багато інших сфер [14]. Рисунок 1.4. Стартова сторінка [weblancer.net](#) [Ошибка! Источник ссылки не найден.] На платформі можна безоплатно публікувати разові замовлення, конкурси, а також розмішувати резюме й портфоліо. Після реєстрації кожен фрілансер отримує п'ять безкоштовних заявок. Додатково ще 50 заявок нараховується за повне заповнення профілю, а саме: – додавання фотографії — +10 заявок; – заповнення резюме — +10 заявок; – перелік наданих послуг — +10 заявок; – завантаження робіт у портфоліо — +20 заявок. Фрілансери також мають можливість придбати додаткові заявки двома способами: оформлення передплати терміном на три місяці, вартість якої залежить від обраної категорії; – разова купівля універсальних заявок, тарифи на які стають доступними після підтвердження номера телефону. Комісія сервісу [Weblancer.net](#) становить 5 % від вартості виконаного замовлення. Усі спірні питання на платформі розглядаються професійними юристами. 1.1.4 [Free-lance.ua](#) До найпопулярніших категорій на платформі [Free-lance.ua](#) належать менеджмент, розробка вебсайтів, дизайн, арт, програмування, оптимізація, тексти, переклади та інші напрями (рис. 1.5). Біржа пропонує три основні формати співпраці між замовниками та фрілансерами: – проекти — разові швидкі завдання; – конкурси — замовник встановлює бюджет і отримує від виконавців різні варіанти виконання, найчастіше у сферах неймінгу, створення логотипів чи ілюстрацій; – вакансії — можливість працевлаштування фрілансера до штату компанії. Рисунок 1.5. Стартова сторінка [free-lance.ua](#) 1.1.5 [Kabanchik.ua](#) Платформа [Kabanchik.ua](#) (рис. 1.6)



функціонує з 2012 року та суттєво відрізняється від класичних фріланс-бірж за принципом своєї роботи. Рисунок 1.5. Стартова сторінка [kabanchik.ua](https://kabanchik.ua) На [Kabanchik.ua](https://kabanchik.ua) можна замовляти різноманітні послуги через інтернет, зокрема: -

 **Знайдено плагіату: 0,14%** <https://mc.today/uk/najkrashhi-ukray...> + 2 ресурсів! id: 16

оздоблювальні та будівельні роботи; - кур'єрські та побутові послуги; - ремонт техніки; - послуги для авто; - дизайн; - організація свят; - репетиторство та інші види послуг. На платформі легко знайти фахівця, який догляне за домашнім улюбленцем або відремонтує побутову

 **Знайдено плагіату: 0,18%** <https://uaspectr.com/2022/01/06/najk...> + 2 ресурсів! id: 17

техніку. Окремий розділ присвячено саме фріланс-послугам, які поділяються на такі категорії: робота в інтернеті, розробка

вебсайтів та додатків, дизайн, інтернет-реклама та інші напрями. Сервіс має власний мобільний додаток, доступний у [Google Play](https://play.google.com/store/apps/details?id=com.kabanchik.ua) та [App Store](https://apps.apple.com/ua/app/kabanchik-ua/id1444444444). У 2015 році [Kabanchik.ua](https://kabanchik.ua) придбав український майданчик [Prom.ua](https://prom.ua). Українські фахівці також можуть працювати на міжнародних платформах, серед яких найбільш популярні: - [Upwork.com](https://www.upwork.com/); - [Fiverr.com](https://www.fiverr.com/); - [WorkingNomads.co/jobs](https://www.workingnomads.co/jobs); - [DesignHill.com](https://www.designhill.com/); - [Gigster.com](https://www.gigster.com/). Ринок фрілансу в Україні продовжує зростати, навіть під час війни. За даними [Freelancehunt](https://freelancehunt.com/), у 2022 році кількість виконаних замовлень зросла на 31 % порівняно з попереднім роком, а обсяг фінансових операцій збільшився на 29 %. 1.2 Аналіз систем пошуку виконавців IT-проектів Процес отримання замовлення на фрілансі зазвичай виглядає так: спеціаліст обирає платформу, подає заявку на цікавий проект, виконує роботу та отримує оплату. Проте на практиці все значно складніше: знайти вигідний проект і виділитися серед інших виконавців буває непросто. Існує безліч спеціалізованих вебсайтів, які допомагають фрілансерам знаходити нові замовлення. Кожна біржа має власні правила участі, механізми безпечної оплати та гарантії для користувачів. Для порівняння були проаналізовані чотири провідні фріланс-платформи, які мають високі позиції на ринку (табл. 1.1). Таблиця 1.1 – Порівняльна характеристика № Назва Переваги Недоліки 1 [Freelancehunt](https://freelancehunt.com/) - Різні способи оплати: - Прозора робота: правила сервісу та інструкції детально описані в базі знань. - Зворотній зв'язок: команда відповідає на всі запитання через форум, соцмережі та блог. - Інструменти безпеки: Сейф, Бізнес-Сейф. - Легке просування по рейтингу. - Різноманітні можливості заробітку: проекти, конкурси та бонусна програма. Висока конкуренція на популярні види робіт. 2 [Weblancer](https://www.weblancer.net/) - Можна отримати консультації від досвідчених колег. - Досвідчений і перевірений механізм роботи, збалансований під потреби користувачів. - Оплата через

 **Знайдено посилань: 0,02%** id: 18

«безпечну угоду»

можлива лише на гаманець [WebMoney](https://www.webmoney.com/). - Висока комісія за операції. - Прив'язка до IP-адреси. - Своєрідна робота адміністрації з клієнтами. 3 [Freelancer](https://www.freelancer.com/) - Можливість знайти виконавців з усього світу без прив'язки до часових поясів. - Зручна служба підтримки, проблеми вирішуються швидко. - Безпечні платежі. - Зручний чат для спілкування з клієнтами. - Тести на компетентність допомагають оцінити навички фрілансера. - Велика кількість неякісних або фейкових проектів. - Автоматизовані заявки фрілансерів ускладнюють оцінку реальних можливостей виконавця. 4 [UpWork](https://www.upwork.com/) - Високі тарифи на проекти, порівняно з бюджетними платформами. - Платежі вбудовані в систему, легко здійснювати оплату. - Нова структура подання пропозицій підвищує шанси на працевлаштування. - Віддалена робота забезпечує гнучкість. - Висока конкуренція. - Комісія за проект від 5 % до 20 %, залежно від заробітку. - Користувацький досвід на деяких сторінках застарілий. Кожна існуюча інформаційна фріланс-платформа виступає своєрідним містком між традиційною економікою та цифровими ринками. Проаналізувавши сучасні сервіси, можна виділити ключові розділи та функції, які доцільно врахувати при розробці власної інформаційної платформи. По-перше, важливо приділити особливу увагу зручності та надійності використання. Візуальна складова вебсайту є критичною, оскільки саме вона впливає на перше враження користувача: привертає увагу, зацікавлює та утримує від переходу з платформи. По-друге, необхідно враховувати політику комісій та оплату за використання платформи. На початковому етапі не планується стягування плати за реєстрацію, проте ця функція може

бути впроваджена пізніше. Важливо, щоб відсоток комісії був помірним, адже він безпосередньо впливає на залученість користувачів та загальну активність платформи. 1.3 Висновки до розділу 1 Проаналізувавши сучасні вітчизняні пошукові сервіси, можна визначити основний функціонал, який характерний для всіх платформ, а також спільні недоліки. Базовий функціонал таких сайтів включає: реєстрацію та авторизацію користувачів; перегляд локальної інформації та новин порталу; зручну навігацію по сайту; пошукову систему; створення завдань і замовлень для фрілансерів та клієнтів; перегляд завдань і замовлень; можливість залишати відгуки на виконані завдання; обмін особистими повідомленнями між користувачами. Спільні недоліки існуючих платформ включають: складну або незручну навігацію; інтерфейс із

» Знайдено посилань: **0,01%**

id: 19

«кривими»

елементами та надмірною інформаційною навантаженою; надмірну кількість умов для початку роботи: необхідність оформлення передплати, введення зайвих даних, велика кількість реклами. Для розробки проєкту важливо чітко визначити вимоги до інформаційної системи (IC): Функціональні вимоги — забезпечення основних операцій користувачів; Вимоги до інтерфейсу — зручність, простота та інтуїтивність; Вимоги до продуктивності — швидка обробка даних та стабільність роботи платформи; Вимоги до безпеки — захист персональних даних та фінансових операцій. РОЗДІЛ 2 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ПОШУКОВОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ Існує багато платформ, які дозволяють користувачам знаходити роботу або отримувати допомогу в різних сферах діяльності. Через зростання кількості таких сервісів та конкуренції на ринку доступні робочі місця часто не забезпечують дохід на рівні прожиткового мінімуму, що створює труднощі для багатьох людей. З іншого боку пошук кваліфікованого фахівця потребує від кандидатури вимагає відповідного рівня професіоналізму, особистісних якостей та інших характеристик. В Україні вже існують сервіси, які дозволяють знайти компетентних виконавців і сприяють встановленню вигідних ділових відносин. Паралельно розробляються нові платформи з розширеними можливостями, що використовують сучасні веб-технології. Для подальшої реалізації пошукової інформаційної системи доцільно провести аналіз сучасних технологій створення веб-орієнтованих інформаційних систем. 2.1 Аналіз програмних веб-технологій для розробки пошукової інформаційної системи Сучасні веб-технології спрямовані на ефективне керування інформаційним наповненням сайту та обробку даних, що надходять від відвідувачів. Зазвичай такі рішення базуються на серверних технологіях, наприклад [ASP](#), [ASP.NET](#), [JSP](#), [PHP](#), або на готових платформах для створення корпоративних сайтів. [ASP / ASP.NET \(Active Server Pages\)](#) — технологія [Microsoft](#), що дозволяє створювати динамічні веб-сторінки з відокремленням функціональної частини від дизайну. [ASP](#)-сторінки містять [HTML](#)-текст, а також сценарії на [JavaScript](#) або [VBScript](#). Сервер обробляє запити та генерує [HTML](#)-код, який відображається у браузері. Ця технологія стала базою для розвитку інших платформ, таких як [PHP](#) та [JSP](#). [JSP \(Java Server Pages\)](#) — технологія для створення серверних сторінок на [Java](#), що є розширенням [Java Servlet API](#). [JSP](#) забезпечує кросплатформенність та дозволяє розробляти зручні для користувача веб-додатки. Специфікація [JSF \(Java Server Faces\)](#) визначає правила побудови серверних компонентів інтерфейсу користувача. [CGI \(Common Gateway Interface\)](#) — одна з перших технологій створення серверних веб-застосовань. [CGI](#) дозволяє виконувати серверні скрипти, які генерують [HTML](#)-код у відповідь на запит браузера. [CGI](#)-застосування можуть бути написані на різних мовах програмування та запускатися як консольні програми під керуванням операційної системи, що обслуговує веб-сервер. [PHP \(Personal Home Pages / Hypertext Preprocessor\)](#) — вільна та популярна технологія для створення серверних веб-сторінок. [PHP](#) заснована на використанні [CGI](#)-застосовань і дозволяє інтегрувати скрипти безпосередньо в [HTML](#)-код. Головною перевагою [PHP](#) є її практичність та гнучкість: мова легко інтегрується з [HTML](#), [JavaScript](#), [WML](#), [XML](#) та іншими мовами програмування для вебу. [PHP](#) надає програмістам інструменти для швидкої та ефективної реалізації завдань. Результати порівняльного аналізу сучасних технологій розробки веб-ресурсів наведено в таблиці 2.1. Розглянуті технології забезпечують сучасну функціональність веб-додатків, ефективний супровід процесів створення сайтів та управління інформаційним контентом. Таблиця 2.1 – Порівняння сучасних технологій розробки веб-сайтів [PHP JSP ASP.NET](#) Багатофункціональність + + - Продуктивність +/- +/- + Простота використання + +/- +/- - Наявність доступних програмних бібліотек + + + Розподіл дизайну та логіки +/- +/- + Результати проведеного аналізу дозволяють зробити висновок про переваги [PHP](#)-



технології в експлуатації. [PHP](#) користується значною популярністю серед вебпрограмістів і на сьогодні є однією з найпоширеніших мов програмування для створення вебдодатків та серверних скриптів. Основні переваги [PHP](#) полягають у практичності, легкості застосування, ефективності, високій продуктивності та гнучкості, що робить її оптимальним вибором для розробки динамічних вебсайтів.

## 2.2 Аналіз засобів реалізації баз даних у вебтехнологіях

Робота з базами даних є ключовою складовою процесу розробки динамічних вебсайтів. Бази даних використовуються для зберігання різнопланової інформації, що необхідна для роботи ресурсу. У спрощеному вигляді база даних являє собою набір взаємозв'язаних таблиць, розміри та кількість яких можуть бути різними. На сервері бази даних акумулюють інформацію статистичного та оперативного характеру. Серед найбільш популярних систем керування базами даних (СКБД) виділяють [MySQL](#), [SQL](#), [Oracle Database](#) та інші. Вибір конкретної СКБД визначається функціональними вимогами та характеристиками інформаційної системи. [MySQL](#) — одна з найпоширеніших безкоштовних СКБД, яка активно використовується для створення динамічних веб-сторінок. Вона підтримує роботу з різними мовами програмування, є багатопотоочною, проста у встановленні та налаштуванні. Серед її переваг: підтримка паралельної роботи багатьох користувачів, обробка великих обсягів даних (до 50 млн рядків у таблицях), висока швидкість виконання запитів та надійна система безпеки. [Oracle Database](#) — об'єктно-реляційна СКБД, яка орієнтована на операційні системи [Windows](#), [Unix](#), [Linux](#) та [MacOS](#). На відміну від [MySQL](#), [Oracle](#) має ширшу сферу застосування і використовується у великих комерційних та державних інформаційних системах. Основними обмеженнями при розробці невеликих вебсайтів є висока вартість ліцензії та складність у пошуку хостингу з підтримкою [Oracle](#).

## SQL (Structured Query Language)

— декларативна мова для роботи з реляційними базами даних, що дозволяє формувати запити, оновлювати та керувати даними, створювати схеми бази даних і модифікувати їх, а також контролювати доступ до інформаційних ресурсів. [SQL](#) може виконувати інтерактивні запити або використовуватися як вбудована мова у прикладних програмах для управління даними. Крім того, стандарт [SQL](#) підтримує механізми перевірки та захисту інформації, що забезпечує надійність роботи з базою даних.

Надійність	Швидкість	Простота	Зручність використання
+++	++	++	+/-

## 2.3 Мова розмітки веб-документів HTML

Більшість веб-сторінок створюються із використанням мови розмітки [HTML](#) або її строгого варіанту [XHTML](#). [HTML](#) інтерпретується браузером та відображається у вигляді документа, зручного для користувача. До п'ятої версії [HTML](#) належить до стандарту [SGML \(ISO 8879\)](#), тоді як [XHTML](#) дотримується суворих правил [XML](#), фактично розширюючи [HTML](#) для забезпечення сумісності з [XML](#)-документами. З появою [HTML5](#) були введені нові елементи та атрибути, що спрощують семантичну розмітку та замінюють універсальні елементи [div](#) і [span](#). Наприклад: [nav](#) — блок навігації сайту; [footer](#) — нижня частина сторінки; [audio](#) та [video](#) — мультимедійні елементи замість [object](#). [HTML5](#) також включає [API](#), доступне для взаємодії з [JavaScript](#), і підтримує розширені можливості [DOM \(Document Object Model\)](#), який забезпечує програмний доступ до структури документів [HTML](#), [XHTML](#) і [XML](#). [XHTML5](#) — це [XML](#)-серіалізація [HTML5](#), яка вимагає суворого дотримання синтаксису [XML](#) і визначає [MIME](#)-тип документа для правильного відображення. Вибір між [HTML5](#) і [XHTML5](#) визначається обранням типу медіа ([MIME-type](#)), який визначає формат документа.

## 2.4 Каскадні таблиці стилів (CSS)

[CSS \(Cascading Style Sheets\)](#) — формальна мова опису зовнішнього вигляду документів, створених за допомогою мов розмітки ([HTML](#), [XHTML](#)). Основна мета [CSS](#) — розділити логічну структуру документа від його презентації, що дозволяє легко змінювати стиль оформлення без зміни [HTML](#)-коду. Правила [CSS](#) можуть зберігатися: у зовнішньому файлі і підключатися за допомогою [link](#) у [head](#): [link rel=](#)

” Знайдено посилань: <b>0,01%</b>	id: 20
"stylesheet"	
href=	
” Знайдено посилань: <b>0,02%</b>	id: 21
"style.css"	
У зовнішньому файлі, підключеному через директиву @import всередині тегу style : style media=	
” Знайдено посилань: <b>0,01%</b>	id: 22
"all"	

@import url(style.css); /style У внутрішньому CSS, розташованому всередині тегів style у head : style body { color: red; } /style У елементі документа, застосованому локально до конкретного тега через атрибут style: p style=

” Знайдено посилань: 0,1%

id: 23

"font-size: 20px; color: green; font-family: Arial, Helvetica, sans-serif;"

... /p Для XML-документів CSS підключається через спеціальне посилання: ?xml-stylesheet type=

” Знайдено посилань: 0,01%

id: 24

"text/css"

href=

” Знайдено посилань: 0,02%

id: 25

"style.css"?

До появи CSS оформлення здійснювалося безпосередньо у HTML-кодi. CSS дозволяє: Застосовувати різні стилі для різних пристроїв (екран, друк, мобільні пристрої, Брайль); Зменшувати час завантаження сторінок, оскільки браузер кешує зовнішні стилі; Легко змінювати дизайн сайту, редагуючи лише один файл CSS; Використовувати додаткові ефекти, наприклад, фіксоване меню або обтікання тексту блоками. Недоліки CSS: Різні відображення в різних браузерах; Іноді необхідно редагувати HTML-код через складні взаємозв'язки між селекторами CSS; Деякі розширення CSS3 замінюються зображеннями для кросбраузерності (наприклад, закруглені кути). CSS використовує модель блоку, де властивість width визначає ширину вмісту, без урахування відступів і рамок. Селектори CSS дозволяють задавати стилі для окремих елементів або груп елементів (дочірні, нащадки, сестринські). Це забезпечує гнучке управління структурою документа і підвищує доступність контенту для різних типів відтворення.

### 2.5 Динамічна мова JavaScript

Для реалізації інтерактивності та анімації на веб-платформі фріланс-біржі буде використана мова JavaScript (JS) та її популярні фреймворки. JavaScript — прототипно-орієнтована скриптова мова програмування, діалект ECMAScript, широко застосовується у браузерах для додавання інтерактивності веб-сторінок. Основні характеристики мови: динамічна та слабка типізація; автоматичне керування пам'яттю; прототипне програмування; функції як об'єкти першого класу; C-подібний, але спрощений синтаксис. JS використовується для: створення інтерактивних веб-сторінок; односторінкових веб-застосунків (ReactJS, AngularJS, Vue.js); серверного програмування (Node.js); десктопних застосунків (Electron, NW.js); мобільних застосунків (React Native, Cordova); сценаріїв у прикладному ПЗ (наприклад, Adobe Creative Suite); інтеграції в PDF-документи та інші середовища. Поява AJAX (Asynchronous JavaScript and XML) дозволила виконувати запити до сервера в асинхронному режимі, оновлюючи окремі частини сторінки без її повного перезавантаження. Це значно підвищило ефективність і інтерактивність веб-застосунків. JS підтримує концепції об'єктно-орієнтованого та функціонального програмування: об'єкти з прототипами, функції як об'єкти першого класу, каррінг, анонімні функції та замикання (closures). Базові вбудовані об'єкти включають: Global, Object, Array, Function, Date, Math та ін.

#### 2.5.1 Бібліотека jQuery

jQuery — популярна бібліотека JavaScript, яка спрощує взаємодію з HTML та DOM, обробку подій, анімацію та роботу з AJAX. Основні принципи: Відокремлення поведінки від структури HTML (ненав'язливий JS); Модульність через ядро та плагіни, що дозволяє підключати лише необхідний функціонал; Спрощена робота з DOM-елементами та AJAX-запитами (\$.post()). Переваги jQuery: Малий розмір дистрибутива; Простота у використанні та добре документована; Лаконічний та розширюваний синтаксис. Недоліки jQuery: Можливе уповільнення роботи при великих проектах; Проблеми сумісності з деякими браузерами.

### 2.6 Висновки до розділу 2

У цьому розділі проведено аналіз сучасних веб-технологій для розробки інформаційних систем: PHP, MySQL, HTML, CSS, JavaScript та jQuery. Основні висновки: PHP — популярна серверна мова, гнучка та ефективна для створення динамічних веб-додатків; MySQL — надійна, швидка та доступна СКБД, оптимальна для зберігання даних динамічних сайтів; HTML5 та XHTML5 — забезпечують структуру та семантичну розмітку документа; CSS3 — відокремлює зовнішній вигляд від структури документа, дозволяє кросбраузерне та адаптивне оформлення; JavaScript — динамічна мова сценаріїв для інтерактивності, з підтримкою прототипного ООП та функціонального програмування; jQuery — бібліотека JS, що спрощує роботу з DOM, подіями та AJAX-запитами. Підхід на основі вбудовування інтерпретованого коду в HTML-шаблони дозволяє

ефективно реалізовувати динамічні веб-застосунки. Кожна технологія має свої переваги та обмеження, що визначає їх область застосування. Для створення пошукової інформаційної системи фріланс-біржі оптимально використовувати комбінацію: [PHP](#) + [MySQL](#) + [HTML](#) + [CSS](#) + [JavaScript](#) + [jQuery](#), що забезпечує інтерактивність, динамічність, зручний інтерфейс та надійну роботу з базами даних. РОЗДІЛ 3 Архітектура та опис розробки Метою даного проєкту є розроблення веб-орієнтованої інформаційної системи для пошуку виконавців іт-проєктів, яка буде корисною як для замовників, так і для фахівців-виконавців. Запропонований сервіс забезпечує зручний пошук замовлень, спрощує процес комунікації між клієнтами та фрілансерами, а також дозволяє оцінювати професійний рівень спеціалістів за допомогою портфоліо, відгуків, рейтингу та статистичних показників профілю. Уся зазначена інформація формується безпосередньо під час роботи на платформі та є достовірною. У цьому розділі наведено вимоги до системи, описано моделі даних, концептуальні схеми та принципи функціонування розробленого програмного продукту. 3.1 Вимоги до проєкту веб-орієнтованої іс Цільовими користувачами системи є програмісти, дизайнери, [seo](#)-спеціалісти, фахівці з відео- та аудіомонтажу, копірайтери і рерайтери, менеджери інтернет-проєктів, адміністратори соціальних мереж, а також розробники мобільних додатків. Система передбачається як відкрита платформа для розміщення та виконання іт-проєктів. До основних функціональних вимог віднесено: реєстрацію та авторизацію користувачів; створення і публікацію замовлень з можливістю додавання файлів; перегляд переліку робіт; реалізацію механізму пошуку за різними критеріями; формування заявок на виконання проєктів; обмін повідомленнями між замовником і виконавцем у режимі чату; ведення рейтингу та історії проєктів; систему сповіщень; можливість блокування користувачів і функціонування служби підтримки. У системі передбачено три основні групи користувачів: адміністратори, замовники та виконавці. Адміністратори здійснюють блокування облікових записів і надають підтримку користувачам. Замовники мають змогу створювати, редагувати та видаляти завдання, а також взаємодіяти з виконавцями та оцінювати їхні роботи. Виконавці переглядають доступні завдання, подають заявки на участь у проєктах, спілкуються із замовниками, оцінюють роботи інших фахівців, формують командні акаунти та генерують власні резюме. Особливістю системи є можливість створення облікового запису типу

” Знайдено посилань: **0,01%**

id: **26**

«команда»,

що дозволяє об'єднувати декілька виконавців для спільної роботи над проєктами. Крім того, передбачено вбудований конструктор резюме, який автоматично формує документ на основі навичок і досвіду користувача, а також функцію ведення блогу, доступну як замовникам, так і виконавцям. Реалізація системи здійснюється із застосуванням сучасних веб-технологій [php](#), [html](#), [css](#), [javascript](#), [jquery](#) та скбд [mysql](#). Загальні вимоги до програмного забезпечення охоплюють відповідність операційній системі, зручність інтерфейсу та необхідний функціональний набір. 3.2 [ER](#)-діаграма Модель

” Знайдено посилань: **0,01%**

id: **27**

«сутність-зв'язок»

([ERM](#)) є концептуальним представленням даних та використовується для створення реляційної бази даних. Основні елементи [ER](#)-діаграми: Сутність – об'єкт, про який зберігається інформація (Рис. 3.1). Зв'язок – іменована асоціація між сутностями, яка відображає взаємодію об'єктів (Рис. 3.2). Атрибути – властивості сутностей та зв'язків, що описують їх характеристики (Рис. 3.3). Використовуючи зазначені елементи, була побудована [ER](#)-діаграма для розробленого продукту (Рис. 3.4), що відображає: користувачів (адміністратори, замовники, виконавці); замовлення та заявки; взаємодію через чат; рейтинг, портфоліо та резюме; ведення блогу та історію проєктів. [ER](#)-діаграма дозволяє чітко структурувати дані та забезпечує ефективну розробку реляційної бази даних для веб-орієнтованої ІС. Рис. 3.1. Позначення сутності в [ER](#)-діаграмах Рис. 3.2. Позначення зв'язку в [ER](#)-діаграмах Рис. 3.3. Позначення атрибуту в [ER](#)-діаграмах Користуючись зазначеними вище фігурами була побудована [ER](#)-діаграма для розробленого продукту (рис. 3.4) Рис. 3.4. [ER](#)-діаграма додатку 3.3 Структура таблиць бази даних Веб-додаток оперує базою даних на засадах [MySQL](#), що складається за 15 таблиць створених відповідно до їх доменних сутностей. До цього переліку входить: [Employee\\_regitbl](#) - таблиця використовується для зберігання всієї інформації про робітника. Таблиця 3.1. Структура таблиці [Employee\\_regitbl](#) Поле Тип Обмеження Опис [employee\\_id int\(11\)](#) [Auto increment](#)

Унікальний ідентифікатор [Fn Text Not Null](#) Зберегти ім'я [Ln Text Not Null](#) Зберегти прізвище [Em Text Not Null](#) Електронна пошта [Un Text Not Null](#) Ім'я користувача. [Pass Text Not Null](#) Пароль. [cn1 Text Not Null](#) Мобільний номер. [cn2 Text Not Null](#) Запасний мобільний номер. [Address Text Not Null](#) Адреса магазину. [City Text Not Null](#) Назва міста магазину. [Que Text Not Null](#) Магазин питання. [Ans Text Not Null](#) Відповідь. [Balance int\(11\) Not Null](#) Баланс для працівника [Img\\_upld Text Not Null](#) Зображення працівника магазину. [Employer\\_regitbl](#) - таблиця використовується для зберігання всієї інформації про роботодавця. Таблиця 3.2. Структура таблиці [Employer\\_regitbl](#) Поле Тип Обмеження Опис [employer\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [Fn Text Not Null](#) Ім'я [Ln Text Not Null](#) Прізвище [Em Text Not Null](#) Адреса електронної пошти магазину [Un text Not Null](#) Ім'я користувача [Pass text Not Null](#) Пароль [Cnm text Not Null](#) Назва компанії [Cadd text Not Null](#) Адреса компанії [Cem text Not Null](#) Електронна пошта [cn1 text Not Null](#) Мобільний номер [City text Not Null](#) Місто [Que text Not Null](#) Питання [Ans text Not Null](#) Відповідь [Balance int\(11\) Not Null](#) Баланс [Img\\_upld text Not Null](#) Зображення роботодавця [Adminlogin](#) - таблиця використовується для зберігання інформації про адміністратора. Таблиця 3.3. Структура таблиці [Adminlogin](#) Поле Тип Обмеження Опис [Admid int\(11\) Auto increment](#) Унікальний ідентифікатор [Admemail text Not Null](#) Електронна пошта [Admpass text Not Null](#) Пароль [Admdp text Not Null](#) Нікнейм [Secque text Not Null](#) Питання [Secans text Not Null](#) Відповідь [Catetbl](#) - таблиця використовується для зберігання всієї інформації про категорію проєкту. Таблиця 3.4. Структура таблиці [Catetbl](#) Поле Тип Обмеження Опис [cate\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [cate\\_name text Not Null](#) Назва категорії [Subcatetbl](#) - таблиця використовується для зберігання інформації про підкатегорію. Таблиця 3.5. Структура таблиці [Subcatetbl](#) Поле Тип Обмеження Опис [sub\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [sub\\_name text Not Null](#) Назва підкатегорії [cate\\_id int\(11\) Not Null](#) Ідентифікатор категорії [Skilltbl](#) - таблиця використовується для зберігання інформації про навички робітника. Таблиця 3.6. Структура таблиці [Skilltbl](#) Поле Тип Обмеження Опис [skill\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [skill\\_name text Not Null](#) Назва навички [cate\\_id int\(11\) Not Null](#) Ідентифікатор категорії [sub\\_id int\(11\) Not Null](#) Ідентифікатор підкатегорії [Contactustbl](#) - таблиця використовується для зберігання всіх повідомлень адміністратору, які можуть надсилати будь-які зареєстровані або незареєстровані користувачі. Таблиця 3.7. Структура таблиці [Contactustbl](#) Поле Тип Обмеження Опис [comp\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [Email text Not Null](#) Електронна пошта [Subject text Not Null](#) Тема [Msg text Not Null](#) Повідомлення [Prjtbl](#) - Ця таблиця використовується для зберігання всієї інформації про проєкт, який може опублікувати роботодавець. Таблиця 3.8. Структура таблиці [Prjtbl](#) Поле Тип Обмеження Опис [p\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [employer\\_id int\(11\) Not Null](#) Ідентифікатор роботодавця [p\\_name text Not Null](#) Назва проєкту [cate\\_id int\(11\) Not Null](#) Ідентифікатор категорії [skill\\_id int\(11\) Not Null](#) Ідентифікатор навичок [Details text Not Null](#) Деталі проєкту [Budget text Not Null](#) Бюджет проєкту [Days int\(11\) Not Null](#) День закінчення [Files text Not Null](#) Вкладений файл [Date date Not Null](#) Дата розміщення [Prj\\_allocated](#) - таблиця використовується для зберігання всієї інформації про проєкт з виділеним бюджетом, що виділяється замовником. Таблиця 3.9. Структура таблиці [Prj\\_allocated](#) Поле Тип Обмеження Опис [allocated\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [employer\\_id int\(11\) Not Null](#) Ідентифікатор роботодавця [employee\\_id int\(11\) Not Null ID](#) робітника [p\\_id int\(11\) Not Null](#) Ідентифікатор проєкту [Budget int\(11\) Not Null](#) Бюджет проєкту [Date date Not Null](#) Дату розміщення [Message](#) - таблиця використовується для зберігання всіх повідомлень, які можуть надсилати працівник і роботодавець. Таблиця 3.10. Структура таблиці [Message](#) Поле Тип Обмеження Опис [id int\(11\) Auto increment](#) Унікальний ідентифікатор [p\\_id int\(11\) Not Null](#) Ідентифікатор проєкту [Username text Not Null](#) Ім'я користувача [Message text Not Null](#) Повідомлення [Bidtbl](#) - таблиця використовується для зберігання всієї інформації про тендери на проєкти, які може робити робітник. Таблиця 3.11. Структура таблиці [Bidtbl](#) Поле Тип Обмеження Опис [bid\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [employer\\_id int\(11\) Not Null](#) Ідентифікатор роботодавця [employee\\_id int\(11\) Not Null ID](#) робітника [Budget text Not Null](#) Бюджет проєкту [Days int\(11\) Auto increment](#) Днів до завершення [project\\_id int\(11\) Not Null](#) Ідентифікатор проєкту [Status text Not Null](#) Статус заказу [Date date Not Null](#) Дату розміщення [Choosed\\_skill](#) - таблиця використовується для зберігання всієї інформації про навички, які може вибрати працівник. Таблиця 3.12. Структура таблиці [Choosed\\_skill](#) Поле Тип Обмеження Опис [cs\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [skill\\_id int\(11\) Not Null](#) Ідентифікатор навичок [employee\\_id int\(11\) Not Null ID](#) робітника [Completeprj](#) - таблиця використовується для зберігання інформації про завершені проєкти. Таблиця 3.13. Структура таблиці [Completeprj](#)



Поле Тип Обмеження Опис [completepri\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [employer\\_id int\(11\) Not Null](#) Ідентифікатор роботодавця [employee\\_id int\(11\) Not Null ID](#) робітника [Prifile text Not Null](#) Комп. файл проєкту [complete\\_date date Not Null](#) Дата завершення [Employee\\_acc](#) - таблиця використовується для зберігання інформації про обліковий записи працівника. Таблиця 3.14. Структура таблиці [Employee\\_acc](#) Поле Тип Обмеження Опис [acc\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [employee\\_id int\(11\) Not Null ID](#) робітника [Name text Not Null](#) Назва облікового запису [ac\\_no text Not Null](#) Номер рахунку [Employerpaytbl](#) - таблиця використовується для зберігання інформації про оплачені проєкти, якими може керувати роботодавець. Таблиця 3.15. Структура таблиці [Employerpaytbl](#) Поле Тип Обмеження Опис [employerpay\\_id int\(11\) Auto increment](#) Унікальний ідентифікатор [p\\_id int\(11\) Not Null](#) Ідентифікатор проєкту [Amount int\(11\) Not Null](#) Кількість робітників [employer\\_id int\(11\) Not Null](#) Ідентифікатор роботодавця [employee\\_id int\(11\) Not Null ID](#) робітника 3.4 Опис користувальницької частини додатку Додаток має 4 типи користувачів: Гість Адміністратор Робітник Роботодавець 3.4.1 Гість Гостьовий тип створено для незареєстрованих користувачів, які або планують реєстрацію, або бажають ознайомитись з додатком та представленими на ньому пропозиціями. Сторінку для гостьового типу авторизації наведено на рис. 3.5. Рис. 3.5 Головний екран додатку для незареєстрованих користувачів Навігаційне меню містить усі можливі варіанти роботи з додатком для гостю. У змісті головної сторінки для гостьового користувача наведено усі відкриті проєкти. Якщо користувач не гість, можливо відгукнутись на проєкт при натисненні на

» Знайдено посилань: **0,02%**

id: 28

«[Bid now](#)».

В іншому випадку пропонується пройти авторизацію як робітник. Форма авторизації (рис. 3.6) використовується однакова для усіх типів користувачів. Рис. 3.6 Форма авторизації У випадку відсутності користувача в базі він може зареєструватися використовуючи форму реєстрації (рис. 3.7). Рис. 3.7 Форма реєстрації При натисненні на

» Знайдено посилань: **0,02%**

id: 29

«[More info](#)»

біля проєкту здійснюється перехід на інформаційну сторінку проєкту (рис. 3.8). Вона доступна без авторизації. Рис. 3.8 Інформаційна сторінка проєкту Для гостьового користувача була створена форма зв'язку з адміністратором для донесення зауважень щодо роботи додатку. Рис. 3.9 Форма зв'язку з адміністратором А також сторінку

» Знайдено посилань: **0,02%**

id: 30

«[About us](#)»

де стисло описано основні напрямки проєктів які присутні в пропозиціях. Рис. 3.10. Сторінка

» Знайдено посилань: **0,02%**

id: 31

«[About us](#)»

3.4.2 Адміністратор Адміністратор має таку саму сторінку перегляду проєктів як гість, але в свою чергу має можливість редагувати системні дані додатку. Під системними даними рахується дані про категорії проєкту, підкатегорії, навички, користувачі та адміністративні повідомлення. Можливі категорії проєктів редагуються зі спеціальної сторінки, що наведена на рис. 3.11. Рис. 3.11 Сторінка редагування категорій проєктів Категорії залежні за зв'язком [one-to-many](#) до підкатегорій. Сторінка редагування підкатегорій проілюстровано на рис. 3.12. Рис. 3.12. Сторінка редагування підкатегорій Так само як категорії залежні від підкатегорій, підкатегорії зв'язані з навичками, які необхідні для виконання проєкту. Для їх редагування було створено окрему сторінку (рис. 3.13). Рис. 3.13. Сторінка редагування навичок Для перегляду та видалення повідомлень користувачів була створена спеціальна сторінка. Рис. 3.14 Сторінка з повідомленнями для адмінстратора 3.4.3 Роботодавець Головна сторінка роботодавця ( див. Додаток А) суттєво відрізняється від попередніх. На верхній частині присутнє фото роботодавця та функція поповнення рахунку. Список проєктів відрізняється групуванням проєктів за їх статусами – закінчені проєкти, проєкти на які виділено бюджет, проєкти за які проведено повну або часткову оплату. В останній групі показуються абсолютно всі проєкти від поточного замовника (роботодавця). Зі списку проєктів можливо проводити оплату роботи виконавця



зі списку. Виконавці проекту обираються на інформаційній сторінці проекту з поміж робітників, що запропонували свої послуги. У арсеналі роботодавця знаходиться основний функціонал наповнення додатку – можливість створювати проекти. Функціонал створення проектів показано на рис. 3.15. Рис. 3.15 Сторінка створення проекту Якщо у роботодавця виникає необхідність оновити дані про себе, він може зробити це на сторінці редагування профілю (рис. 3.16). Рис. 3.16 Сторінка редагування профілю роботодавця

### 3.3.4 Робітник

Головна сторінка ідентична зі сторінкою роботодавця. Відмінності є лише в відсутності групи проектів на які виділено бюджет, а також наявності форми заповнення платіжних даних. Основним функціоналом є відгуки на проекти замовників. На інформаційній сторінці проекту (рис. 3.17), на відміну від роботодавця, робітник може залишити заявку на виконання роботи з вказанням термінів та вартості робіт. Усі заявки подані робітником відображаються на спеціальній сторінці заявок (рис. 3.18). Рис. 3.17 Інформаційна сторінка проекту для робітника Рис. 3.18. Сторінка заявок робітника Проте не кожен робітник, може подавати заявку на проведення робіт. Необхідна наявність співпадіння навичок вказаних в проекті та профілі робітника. Навички робітника можливо редагувати на спеціальній сторінці редагування навичок (рис. 3.19). Контактні дані та фото редагуються на сторінці аналогічній сторінці редагування профілю роботодавця. Рис. 3.19. Сторінка редагування навичок робітника

### Задля полегшення пошуку пропозицій створено сторінку пошуку проектів за вказаними в профілі навичками (рис. 3.20). На ній перераховані абсолютно усі проекти, що підходять робітнику. Рис. 3.20 Сторінка відібраних проектів

## 3.5 Висновки до розділу 3

В ході проведеної роботи розроблено веб-додаток, який було спроектовано та реалізовано фріланс біржі з пошуковою системою. Структура проекту проста і добре організована, щоб користувач мав змогу легко користуватися додатком, а розробник з легкістю і мінімальними затратами часу зміг модернізувати під потреби. Для подальшого розвитку до розробленої системи можна внести наступні поліпшення: -розробити мобільні версії додатку, щоб надати користувачам можливість бути більш мобільним; -додати систему трекінгу проекту для робітників та роботодавців Розроблена система може бути впроваджена для будь-яких організацій та закладів, які користуються фріланс послугами.

## ВИСНОВКИ

Сьогодні фріланс активно розвивається, зокрема на ринку ІТ-послуг. Існує значна кількість спеціалізованих веб-ресурсів, що допомагають виконавцям знаходити нові замовлення. Проаналізувавши діяльність таких платформ, їхні функціональні можливості та недоліки, було обґрунтовано доцільність подальших досліджень у цьому напрямі та прийнято рішення про створення власної веб-орієнтованої інформаційної системи для пошуку виконавців ІТ-проектів. У межах випускної кваліфікаційної роботи проведено огляд і аналіз сучасного стану проблеми, зокрема розглянуто особливості розробки веб-орієнтованих інформаційних систем та сформовано вимоги до їх функціональних можливостей. Виконано дослідження предметної області, визначено вимоги до проектованої системи, обрано методи й інструменти її реалізації, а також змодельовано інформаційну систему, спроектовано інтерфейс і структуру бази даних. До основних функцій розробленого програмного продукту належать: реєстрація й авторизація користувачів, розміщення замовлень із можливістю додавання файлів, перегляд завдань зареєстрованими користувачами, пошук робіт за різними критеріями, створення заявок на виконання проектів, обмін повідомленнями між замовником і виконавцем, формування сповіщень, рейтингу та історії проектів, а також блокування користувачів. Система надає виконавцям можливість виконувати замовлення, створювати портфоліо та взаємодіяти з клієнтами. Замовники, у свою чергу, можуть публікувати завдання, обирати виконавців і спілкуватися з ними за допомогою чату. Важливою особливістю сервісу є підтримка різних типів облікових записів — як для індивідуального користування, так і для команд розробників або компаній, що значно розширює функціональні можливості платформи та дозволяє отримувати комплексні послуги з мінімальними затратами часу на пошук фахівців. Проект має добре структуровану архітектуру, що забезпечує зручність роботи для користувачів та дозволяє розробникам оперативно модернізувати систему відповідно до нових вимог, зокрема створювати мобільні версії або впроваджувати модулі трекінгу проектів. Розроблена пошукова інформаційна система може бути використана як відкрита фріланс-платформа, яка спрощує процес пошуку замовлень і комунікацію між клієнтами та виконавцями. Інтерфейс веб-системи є інтуїтивно зрозумілим і надійним, а реалізовані механізми захисту забезпечують доступ до функціоналу лише для авторизованих користувачів.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

[Adriaan de Jonge, Phil Dutson. jQuery, jQuery UI, and jQuery Mobile: Recipes and Examples – Addison-Wesley Professional, 2012. 400 pages.](#)  
[ASP.NET Core7. URL: https://learn.microsoft.com/aspnet/core/tutorials/](#)

[razor-pages/razor-pages-start?view=aspnetcore-7.0&tabs=visual-studio](#). (Date of access: 24.12.2022). [Browser Support | jQuery](#). URL: <https://jquery.com/browser-support/>. (Date of access: 24.12.2022). [Chris Dawson, Ben Straub. Building Tools with GitHub: Customize Your Workflow - O'Reilly Media](#), 2016. 302 pages. [David Geary, Cay S. Horstmann. Core JavaServer Faces, 3rd Edition](#). 2011. 656 p. [Deepu K Sasidharan, Sendil Kumar N. Full Stack Development with JHipster - Packt Publishing](#), 2018. 380 pages. [Eric Elliott. Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries - O'Reilly Media](#), 2014. 254 pages. [free-lance.ua](#). URL: [free-lance.ua](https://free-lance.ua). (Date of access: 24.12.2022). [Freelancehunt](#). URL: [freelancehunt.com](https://freelancehunt.com). (Date of access: 24.12.2022). [Jason Price, Oracle Database 11g SQL](#). McGraw-Hill Osborne Media, 2007. 656c. [jQuery Overview](#). URL: <https://www.tutorialspoint.com/jquery/jquery-overview.htm>. (Date of access: 24.12.2022). [Kabanchik.ua](#). URL: [kabanchik.ua](https://kabanchik.ua). (Date of access: 24.12.2022). [Matt Frisbie. Professional JavaScript for Web Developers, 4th Edition](#). Wrox. 2019. 1200p. [MC.today](#). С. Кіт. Найкращі українські та зарубіжні біржі фрілансу (07 Dec 2022). URL: <https://mc.today/birzhi-frilansa/>. (Date of access: 24.12.2022). <https://mc.today/birzhi-frilansa/>. [MySQL Tutorial](#). URL: <https://www.w3schools.com/MySQL/default.asp>. (Date of access: 24.12.2022). [Philip A. Koryaka. CGI - Simple Common Gateway Interface Class](#) URL: <http://www.xserver.ru/computer/langprogr/cgi/>. (Date of access: 24.12.2022). [UML моделирование](#). URL: <https://habr.com>. (Date of access: 24.12.2022). [UpWork](#). URL: <https://www.upwork.com/>. (Date of access: 24.12.2022). [Weblancer.net](#). URL: [weblancer.net](https://weblancer.net). (Date of access: 24.12.2022). [Why Quill - Quill Rich Text Editor](#). URL: <https://quilljs.com/guides/why-quill>. (Date of access: 24.12.2022). Адміністрування. URL: <https://workspace.google.com/intl/uk/products/admin/>. (Date of access: 24.12.2022). Введение в поисковую оптимизацию. URL: <https://developers.google.com/search/docs/fundamentals/seo-starter-guide?hl>. (Date of access: 24.12.2022). Джон Дакетт - [HTML](#) и [CSS](#). Разработка и дизайн веб-сайтов [пер. С англ. М.А. Райтмана]. М. : Эксмо, 2013. 480 с. Доценко С. І. Людино-машинний інтерфейс: навч.посібник. Харків: УкрДУЗТ, 2022. 135 с. Дронов В. А. [HTML5](#), [CSS3](#) и [Web 2.0](#). Разработка современных [Web](#)-сайтов. СПб. : БХВ, 2011. 416 с. Кевин Мельтцер Разработка [CGI](#)-приложений на [Perl](#). Вильямс. 2011. 400с. Кешування та продуктивність веб-додатків. URL: <https://habr.com/company/ruvds/blog/350310/>. (Date of access: 24.12.2022). Котеров Д.В., Симдянов И.В. PHP 8. СПб.: БХВ, 2021. – 850с. Основы [CSS3](#). URL: <https://metanit.com/web/html5/>. (Date of access: 24.12.2022). Пол МакФедрис. [Web Design Playground](#) (2019). URL: [https://vk.com/wall-54530371\\_296845](https://vk.com/wall-54530371_296845). (Date of access: 24.12.2022). Посібник з [HTML](#) та [CSS3](#). URL: <https://metanit.com/web/html5/>. (Date of access: 24.12.2022). Учебник [HTML](#) с [CSS](#). URL: [https://www.schoolsw3.com/html/html\\_css.php](https://www.schoolsw3.com/html/html_css.php). (Date of access: 24.12.2022). Хеник Б. [HTML](#) и [CSS](#): путь к совершенству. СПб. : Питер, 2011. 336 с. Язык программирования [Python](#). URL: <https://metanit.com/python/>. (Date of access: 24.12.2022). ДОДАТКИ Додаток А. Головна сторінка роботодавця Додаток Б. Код [employee\\_home.php](#) ?[php\\_session\\_start\(\)](#); [ob\\_start\(\)](#); [include\(\)](#)

Знайдено посилань: <b>0,02%</b>	id: <b>32</b>
<a href="#">"header.php"</a>	
); <a href="#">include()</a>	
Знайдено посилань: <b>0,02%</b>	id: <b>33</b>
<a href="#">"Config.php"</a>	
); <a href="#">\$eid</a> = <a href="#">\$_SESSION['employer_id']</a> ; <a href="#">\$em</a> = <a href="#">\$_SESSION['em']</a> ; <a href="#">\$sql</a> =	
Знайдено посилань: <b>0,07%</b>	id: <b>34</b>
<a href="#">"select * from employer_regitbl where em='\$em'"</a>	
; <a href="#">\$res</a> = <a href="#">mysql_query(\$sql)</a> ; <a href="#">\$row</a> = <a href="#">mysql_fetch_array(\$res)</a> ; if (isset( <a href="#">\$_POST['submit']</a> ))) { <a href="#">mysql_query()</a>	
Знайдено посилань: <b>0,06%</b>	id: <b>35</b>
<a href="#">"update employer_regitbl set balance="" . <a href="#">\$_POST['addbal']</a>"</a>	
.	
Знайдено посилань: <b>0,02%</b>	id: <b>36</b>
<a href="#">"where employer_id="</a>	

. \$eid) or die(	
” Знайдено посилань: 0,01%	id: 37
"Error"	
); } ? style type=	
” Знайдено посилань: 0,01%	id: 38
"text/css"	
.link { font-family: Arial, Helvetica, Sans-serif; font-size: 10px; display: inline-block; border: solid 1px #333; background: #CCC; color: #333; text-decoration: none; padding: 5px; } .link:hover { border: solid 1px #CCC; background: #333; color: #CCC; } /style div id=	
” Знайдено посилань: 0,01%	id: 39
"tooplate_middle"	
div id=	
” Знайдено посилань: 0,01%	id: 40
"mid_title"	
Welcome ?php echo \$_SESSION['unm']; ? /div img src=	
” Знайдено посилань: 0,06%	id: 41
"UserImages/ ?php echo \$row[14]; ? "	
height=	
” Знайдено посилань: 0,01%	id: 42
"90"	
width=	
” Знайдено посилань: 0,01%	id: 43
"90"	
/ div class=	
” Знайдено посилань: 0,01%	id: 44
"cleaner"	
/div /div !-- end of middle -- div id=	
” Знайдено посилань: 0,01%	id: 45
"tooplate_main_top"	
/div div id=	
” Знайдено посилань: 0,01%	id: 46
"tooplate_main"	
div id=	
” Знайдено посилань: 0,01%	id: 47
"gallery"	
form method=	
” Знайдено посилань: 0,01%	id: 48
"POST"	
table tr td input type=	
” Знайдено посилань: 0,01%	id: 49
"text"	
name=	
” Знайдено посилань: 0,01%	id: 50
"addbal"	
placeholder=	
” Знайдено посилань: 0,06%	id: 51
"Add Amount to your A/c"	

/td td input type=	
” Знайдено посилань: 0,01%	id: 52
"submit"	
name=	
” Знайдено посилань: 0,01%	id: 53
"submit"	
value=	
” Знайдено посилань: 0,02%	id: 54
"Add Balance"	
/td /tr /table /form div class=	
” Знайдено посилань: 0,01%	id: 55
"cleaner"	
/div /div div class=	
” Знайдено посилань: 0,01%	id: 56
"cleaner"	
/div /div !-- end of main -- div id=	
” Знайдено посилань: 0,01%	id: 57
"tooplate_main_bot"	
/div div id=	
” Знайдено посилань: 0,01%	id: 58
"tooplate_middle"	
div id=	
” Знайдено посилань: 0,01%	id: 59
"mid_title"	
/div div id=	
” Знайдено посилань: 0,01%	id: 60
"mid_title"	
Completed Project List /div /div !-- end of middle -- div id=	
” Знайдено посилань: 0,01%	id: 61
"tooplate_main_top"	
/div div id=	
” Знайдено посилань: 0,01%	id: 62
"tooplate_main"	
div id=	
” Знайдено посилань: 0,01%	id: 63
"gallery"	
?php \$result1 = mysql_query(	
” Знайдено посилань: 0,07%	id: 64
"select * from completeprj where employer_id="	
. \$eid, \$con) or die(mysql_error()); if (mysql_num_rows(\$result1) 0) { while (\$r1 =	
mysql_fetch_array(\$result1)) { \$result2 = mysql_query(	
” Знайдено посилань: 0,08%	id: 65
"select * from prjtbl where p_id=" . \$_1[1]	
, \$con) or die(	
” Знайдено посилань: 0,05%	id: 66
"Error in Select Query 2"	

```
); if (mysql_num_rows($result2) 0) { while ($_2 = mysql_fetch_array($result2)) { ? table width=
” Найдено посилань: 0,01% id: 67
"100%"
border=
” Найдено посилань: 0,01% id: 68
"2"
tr td align=
” Найдено посилань: 0,01% id: 69
"left"
style=
” Найдено посилань: 0,1% id: 70
"font-family:Arial;font-size:20px;color:#B6300E" ?php echo $ 2[2]
; ? /td tr td align=
” Найдено посилань: 0,06% id: 71
"left" ?php echo $_2[5]
; ? /td /tr ?php $result10 = mysql_query(
” Найдено посилань: 0,08% id: 72
"select * from skilltbl where skill_id=" . $_2[4]
); if (mysql_num_rows($result10) 0) { while ($_10 = mysql_fetch_array($result10)) { ? tr td
align=
” Найдено посилань: 0,01% id: 73
"left"
style=
” Найдено посилань: 0,08% id: 74
"font-family:Arial;font-size:15px;color:#6E103E" ?php echo $_10[1]
; ? /td /tr ?php } } ? tr td a href=
” Найдено посилань: 0,14% id: 75
"employercompletepriview.php?pid= ?php echo $_2[0]; ? & cid= ?php echo $_1[0]; ? "
class='link' More Info /a a href=
” Найдено посилань: 0,14% id: 76
"employerpayment.php?pid= ?php echo $ 2[0]; ? & cid= ?php echo $ 1[0]; ? "
class='link' Payment /a /td /tr /table ?php } } } ? div class=
” Найдено посилань: 0,01% id: 77
"cleaner"
/div /div div class=
” Найдено посилань: 0,01% id: 78
"cleaner"
/div /div !-- end of main -- div id=
” Найдено посилань: 0,01% id: 79
"tooplate_main_bot"
/div div id=
” Найдено посилань: 0,01% id: 80
"tooplate_middle"
div id=
” Найдено посилань: 0,01% id: 81
"mid_title"
```



/div div id=	
” Знайдено посилань: 0,01%	id: 82
"mid_title"	
Allocated Project List /div /div !-- end of middle -- div id=	
” Знайдено посилань: 0,01%	id: 83
"tooplate_main_top"	
/div div id=	
” Знайдено посилань: 0,01%	id: 84
"tooplate_main"	
div id=	
” Знайдено посилань: 0,01%	id: 85
"gallery"	
?php \$result11 = mysql_query(	
” Знайдено посилань: 0,07%	id: 86
"select * from prj_allocated where employer_id="	
. \$eid, \$con) or die(mysql_error()); if (mysql_num_rows(\$result11) 0) { while (\$r11 = mysql_fetch_array(\$result11)) { \$result12 = mysql_query(	
” Знайдено посилань: 0,08%	id: 87
"select * from prjtbt where p_id=" . \$r11[1]	
, \$con) or die(	
” Знайдено посилань: 0,05%	id: 88
"Error in Select Query 2"	
); if (mysql_num_rows(\$result12) 0) { while (\$r12 = mysql_fetch_array(\$result12)) { ? table width=	
” Знайдено посилань: 0,01%	id: 89
"100%"	
border=	
” Знайдено посилань: 0,01%	id: 90
"2"	
tr td align=	
” Знайдено посилань: 0,01%	id: 91
"left"	
style=	
” Знайдено посилань: 0,1%	id: 92
"font-family:Arial;font-size:20px;color:#B6300E" ?php echo \$r12[2]	
; ? /td tr td align=	
” Знайдено посилань: 0,06%	id: 93
"left" ?php echo \$r12[5]	
; ? /td /tr ?php \$result1 = mysql_query(	
” Знайдено посилань: 0,08%	id: 94
"select * from skilltbl where skill_id=" . \$r12[4]	
); if (mysql_num_rows(\$result1) 0) { while (\$r1 = mysql_fetch_array(\$result1)) { ? tr td align=	
” Знайдено посилань: 0,01%	id: 95
"left"	
style=	
” Знайдено посилань: 0,08%	id: 96
"font-family:Arial;font-size:15px;color:#6E103E" ?php echo \$r1[1]	

; ? /td /tr ?php } } ? tr td a href=	
” Знайдено посилань: <b>0,08%</b>	id: <b>97</b>
"prj_allocate.php?pid= ?php echo \$ 12[0]; ? "	
class='link' More Info /a a href=	
” Знайдено посилань: <b>0,08%</b>	id: <b>98</b>
"message.php?pid= ?php echo \$ 12[0]; ? "	
class='link' message /a /td /tr /table ?php } } } ? div class=	
” Знайдено посилань: <b>0,01%</b>	id: <b>99</b>
"cleaner"	
/div /div div class=	
” Знайдено посилань: <b>0,01%</b>	id: <b>100</b>
"cleaner"	
/div /div !-- end of main -- div id=	
” Знайдено посилань: <b>0,01%</b>	id: <b>101</b>
"tooplate_main_bot"	
/div div id=	
” Знайдено посилань: <b>0,01%</b>	id: <b>102</b>
"tooplate_middle"	
div id=	
” Знайдено посилань: <b>0,01%</b>	id: <b>103</b>
"mid_title"	
/div div id=	
” Знайдено посилань: <b>0,01%</b>	id: <b>104</b>
"mid_title"	
Paid Project List /div /div !-- end of middle -- div id=	
” Знайдено посилань: <b>0,01%</b>	id: <b>105</b>
"tooplate_main_top"	
/div div id=	
” Знайдено посилань: <b>0,01%</b>	id: <b>106</b>
"tooplate_main"	
div id=	
” Знайдено посилань: <b>0,01%</b>	id: <b>107</b>
"gallery"	
?php \$result10 = mysql_query(	
” Знайдено посилань: <b>0,08%</b>	id: <b>108</b>
"select employerpay_id,p_id from employerpaytbl where employer_id="	
. \$eid, \$con) or die(	
” Знайдено посилань: <b>0,05%</b>	id: <b>109</b>
"Error in Select Query 1"	
); if (mysql_num_rows(\$result10) 0) { while (\$r10 = mysql_fetch_array(\$result10)) { \$result20 =	
mysql_query(	
” Знайдено посилань: <b>0,08%</b>	id: <b>110</b>
"select * from prjtbl where p_id=" . \$_10[1]	
, \$con) or die(	
” Знайдено посилань: <b>0,05%</b>	id: <b>111</b>
"Error in Select Query 2"	

```

); if (mysql_num_rows($result20) 0) { while ($20 = mysql_fetch_array($result20)) { ? table
width=
” Найдено посилань: 0,01% id: 112
"100%"
border=
” Найдено посилань: 0,01% id: 113
"2"
tr td align=
” Найдено посилань: 0,01% id: 114
"left"
style=
” Найдено посилань: 0,1% id: 115
"font-family:Arial;font-size:20px;color:#B6300E" ?php echo $ 20[2]
; ? /td tr td align=
” Найдено посилань: 0,06% id: 116
"left" ?php echo $ 20[5]
; ? /td /tr ?php $result1 = mysql_query(
” Найдено посилань: 0,08% id: 117
"select * from skilltbl where skill_id=" . $ 20[4]
); if (mysql_num_rows($result1) 0) { while ($1 = mysql_fetch_array($result1)) { ? tr td align=
” Найдено посилань: 0,01% id: 118
"left"
style=
” Найдено посилань: 0,08% id: 119
"font-family:Arial;font-size:15px;color:#6E103E" ?php echo $ 1[1]
; ? /td /tr ?php } } ? tr td a href=
” Найдено посилань: 0,14% id: 120
"employercompletepreview.php?pid= ?php echo $ 20[0]; ? & cid= ?php echo $ 10[0]; ? "
class='link' More Info /a /td /tr /table ?php } } } ? div class=
” Найдено посилань: 0,01% id: 121
"cleaner"
/div /div div class=
” Найдено посилань: 0,01% id: 122
"cleaner"
/div /div !-- end of main -- div id=
” Найдено посилань: 0,01% id: 123
"tooplate_main_bot"
/div div id=
” Найдено посилань: 0,01% id: 124
"tooplate_middle"
div id=
” Найдено посилань: 0,01% id: 125
"mid_title"
All Project List /div div class=
” Найдено посилань: 0,01% id: 126
"cleaner"

```

```
/div /div !-- end of middle -- div id=
```

” Знайдено посилань: **0,01%**

id: 127

```
"tooplate_main_top"
```

```
/div div id=
```

” Знайдено посилань: **0,01%**

id: 128

```
"tooplate_main"
```

```
div id=
```

” Знайдено посилань: **0,01%**

id: 129

```
"gallery"
```

```
?php $result = mysql_query(
```

” Знайдено посилань: **0,36%**

id: 130

```
"select * from prtbl WHERE p_id NOT IN (select p_id from completepr) AND p_id NOT IN (select p_id from prj_allocated) AND p_id NOT IN (select p_id from employerpaytb) AND employer_id="
```

```
. $eid, $con) or die(
```

” Знайдено посилань: **0,05%**

id: 131

```
"Error in Select Query "
```

```
); if (mysql_num_rows($result) 0) { while ($r = mysql_fetch_array($result)) { ? br / table width=
```

” Знайдено посилань: **0,01%**

id: 132

```
"100%"
```

```
border=
```

” Знайдено посилань: **0,01%**

id: 133

```
"2"
```

```
tr td align=
```

” Знайдено посилань: **0,01%**

id: 134

```
"left"
```

```
style=
```

” Знайдено посилань: **0,08%**

id: 135

```
"font-family:Arial;font-size:20px;color:#B6300E" ?php echo $ [2]
```

```
; ? /td /tr tr td align=
```

” Знайдено посилань: **0,06%**

id: 136

```
"left" ?php echo $ [5]
```

```
; ? /td /tr ?php $result1 = mysql_query(
```

” Знайдено посилань: **0,08%**

id: 137

```
"select * from skilltbl where skill_id=" . $ [4]
```

```
); if (mysql_num_rows($result1) 0) { while ($r1 = mysql_fetch_array($result1)) { ? tr td align=
```

” Знайдено посилань: **0,01%**

id: 138

```
"left"
```

```
style=
```

” Знайдено посилань: **0,08%**

id: 139

```
"font-family:Arial;font-size:15px;color:#6E103E" ?php echo $r1[1]
```

```
; ? /td /tr ?php } } ? tr td a href=
```

” Знайдено посилань: **0,08%**

id: 140

```
"employerviewbid.php?pid= ?php echo $ [0]; ? "
```

```
class='link' More Info /a a href=
```

” Знайдено посилань: **0,08%**

id: 141

```
"employereditpr.php?pid= ?php echo $ [0]; ? "
```

class='link' Edit /a a href=	
<div>”</div> Знайдено посилань: 0,08%	id: 142
"employerdelprj.php?pid=?php echo \$ [0]; ? "	
class='link' Delete /a /td /tr /table ?php} } ? ?php include(	
<div>”</div> Знайдено посилань: 0,02%	id: 143
"footer.php"	
); ?	

Відмова від відповідальності:

Цей звіт повинен бути правильно інтерпретований та проаналізований кваліфікованою особою, яка несе відповідальність за оцінку!

Будь -яка інформація, наведена у цьому звіті, не є остаточною та підлягає ручному огляду та аналізу. Дотримуйтесь вказівок: [Рекомендації з оцінки](#)