

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ЗАКЛАД
„ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА”

Навчально-науковий інститут фізики,
математики та інформаційних технологій

Кафедра фізико-технічних систем та інформатики

Климухіна Альона Миколаївна

АНАЛІЗ ПРОГРАМНИХ ВЕБ ТЕХНОЛОГІЙ ТА РОЗРОБКА
ПОШУКОВОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ФРІЛАНС БІРЖІ

кваліфікаційна робота
здобувача вищої освіти другого (магістерського) рівня
освітньої програми «Комп’ютерні науки та інформаційні технології»
за спеціальністю 122 „Комп’ютерні науки ”

Особистий підпис _____

Науковий керівник _____ Юрій КОЗУБ, д.т.н., доцент

Завідувач кафедри _____ Юрій КОЗУБ, д.т.н., доцент

Полтава – 2023

АНОТАЦІЯ

Климухіна А.М.

Тема: Аналіз програмних веб технологій та розробка пошукової інформаційної системи для фріланс біржі.

Спеціальність: 122 „Комп’ютерні науки”.

Установа: ДЗ ЛНУ імені Тараса Шевченка, 2023р.

Кваліфікаційна робота містить: 69 стор., 4 табл., 26 рис., 34 джерела, 3 додатки.

Об’єкт дослідження – веб технології, процес програмної реалізації пошукової інформаційної системи для фріланс біржі.

Предмет дослідження – технології створення пошукової інформаційної системи для фріланс біржі.

Мета роботи – аналіз програмних веб технологій та розробка пошукової інформаційної системи для фріланс біржі.

Результати роботи. Досліджено веб технології розробки пошукової інформаційної системи для фріланс біржі. Досліджено функціональні можливості існуючих сучасних аналогів, виявлено їх переваги та недоліки. Обрано вебтехнології та розроблено пошукову інформаційну систему для фріланс біржі.

Ключові слова: PHP, MySQL, HTML, CSS, JS, JQuery

ABSTRACT

Klymukhina A.M.

Theme: Analysis of software web technologies and development of a search information system for a freelance marketplace.

Specialty: 122 "Computer Science"

Institution: Luhansk Taras Shevchenko National University, 2023

Qualification work contains: 69 pages, 4 tables, 26 figures, 34 sources, 3 appendices.

Object of research is web technologies, the process of software implementation of a search information system for a freelance marketplace.

Subject of research - the technology of creating a search information system for the freelance marketplace

Purpose of the study is the analysis of software web technologies and the development of a search information system for the freelance marketplace.

Results of research. Web technologies for developing a search information system for a freelance marketplace were studied. Functional capabilities of existing modern analogues were studied, their advantages and disadvantages were revealed. Web technologies were chosen and a search information system was developed for the freelance marketplace.

Keywords: PHP, MySQL, HTML, CSS, JS, JQuery

ЗМІСТ

	Стор.
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ	5
ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ.....	9
1.1 Аналіз з предметної області	10
1.2 Аналіз систем пошуку виконавців ІТ проєктів.....	18
1.3 Висновки до розділу 1	21
РОЗДІЛ 2 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	23
2.1. Аналіз програмних веб технологій для розробки пошукової інформаційної системи.....	23
2.2 Аналіз засобів реалізації баз даних у веб-технологіях	26
2.3 Мова розмітки вебдокументів HTML.....	27
2.4 Каскадні таблиці стилів – CSS	28
2.5 Динамічна мова JavaScript	33
2.5.1 <i>jQuery</i>	35
2.6 Висновки до розділу 1	36
РОЗДІЛ 3 АРХІТЕКТУРА ТА ОПИС РОЗРОБКИ.....	38
3.1 Вимоги до проєкту веборієнтованої ІС	38
3.2 ER-діаграма.....	40
3.3 Структура таблиць бази даних	42
3.4 Опис користувальницької частини додатку.....	49
3.4.1 <i>Гість</i>	49
3.4.2 <i>Адміністратор</i>	52
3.4.3 <i>Роботодавець</i>	55
3.4.4 <i>Робітник</i>	56
3.5 Висновки до розділу 3	58
ВИСНОВКИ.....	59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТКИ.....	64
Додаток А. Головна сторінка роботодавця	64
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	-	Hyper Text Markup Language;
HTTP	-	HyperText Transfer Protocol;
ISO	-	International Organization for Standardization;
JS	-	JavaScript;
PHP	-	Hypertext Preprocessor;
SEO	-	Search Engine Optimization;
SGML	-	Standard Generalized Markup Language;
SQL	-	Structured Query Language;
UML	-	Unified Modeling Language;
IC	-	інтелектуальна система;
IT	-	інформаційні технології;
OC	-	операційна система;
ПЗ	-	програмне забезпечення;
ПК	-	персональний комп'ютер.

ВСТУП

Пандемія 2019 року охопила весь світ і змінила його так сильно, що змінила всіма нами улюблене та стабільне життя, а війна в Україні наклала величезну кількість обмежень та заборон, так чи інакше, повністю перевернувши світовий ринок праці, величезна кількість людей залишилася без роботи, але з сильним прагненням жити та адаптуючись до нового порядку існування, де інтернет став важливою частиною життя та переходом на віддалену діяльність. Для тієї ж більшості людей виходом став фріланс.

Технологічний світ не стоїть на місці, де запитів і замовлень людей, які володіють якимось бізнесом, наприклад створення того ж простого логотипу, постають з кожним днем все більше і більше, а питання, де знайти спеціалізовану людину, яка знається на їхньому завданні, все складніше і складніше, особливо якщо часто питання стоїть у довірі між людьми. Хоча в даний час існує безліч сервісів, які надають можливість знайти кваліфікованого фахівця і можуть привести їх до вигідних взаємин і партнерських угод, але в цілому дана тема в нашій країні майже не розвивається, тим більше якщо врахувати той факт, що величезна кількість подібних іноземних сайтів блокують, таким чином розробка порталу для фрілансерів з використанням сучасних засобів є актуальним питанням.

Мета роботи - аналіз програмних веб технологій та розробка пошукової інформаційної системи, порталу для фрілансерів.

Для досягнення мети необхідно вирішити наступні **завдання**:

- а
аналіз та дослідження ринку;
- а
аналіз вимог до системи;
- р
розробка архітектури системи;

—

р

розробка програми.

Об’єкт дослідження

— сервіси самостійного контролю свого дня та вирішення цікавих завдань, замовників, з боку розробника.

Предмет дослідження

— сервіси, які є уповноваженими посередниками між замовниками, тобто звичайними користувачами та фрілансерами, які є фахівцями в будь-якій науковій, технічній та мистецькій галузі.

Методи дослідження.

Теоретичні методи: аналіз науково-технічної літератури та інтернет-ресурсів з проблеми дослідження;

Емпіричні методи: аналіз оптимізації розробки сучасних засобів вебтехнологій у концепції веб-орієнтованої інформаційної системи пошуку виконавців IT-проектів.

Наукова новизна:

—

р

розроблено пошукову інформаційну систему «FreeLance».

Практичне значення

Полягає в програмній реалізації софту «FreeLance», засобами PHP, MySQL, HTML, CSS, JS, JQuery. Розроблений вебсервіс буде використовуватись, як відкрита фріланс-платформа, яка, в свою чергу, повинна полегшити знаходження замовлення та комунікацію між клієнтами та фрілансерами.

Структура і обсяг роботи

Робота складається з вступу, трьох розділів, висновків списку використаних джерел, додатків. Обсяг роботи становить 70 сторінок, обсяг використаної літератури – 34 джерела.

Перший розділ містить аналіз особливостей існуючих сучасних вебдодатків для фрілансерів. Надано огляд вітчизняних сервісів та визначено вимоги для створення нового вебзастосування «FreeLance».

У другому розділі проводиться дослідження існуючих сучасних технологій веброзробки, з'ясовано особливості деяких найпопулярніших веб-орієнтованих засобів розробки, досліджено недоліки існуючих інструментів розробки сайтів. Обрано найсучасніші інструменти для реалізації магістерського завдання.

У третьому розділі надано програмну реалізацію вебзастосування «FreeLance». Розглянуто основні етапи створення інструментів застосування та проведено дослідження значимості кожного з них.

Додатки містять скрін Головної сторінки пошукової інформаційної системи для фріланс біржі та головні елементи коду `employee_home.php`.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

У наш час широкої популярності набула робота поза штатом компаній, яку називають фрілансом. Також можна розглядати фріланс як перший крок до власного бізнесу. Основні переваги віддаленої роботи фрілансерів - праця вдома або там де вам зручно, самостійне складання свого робочого графіку та відсутність контролю.

Фріланс в Україні - рід діяльності, який безпосередньо належить до ІТ-сфери. Найчастіше це проектування, адміністрування сайтів та наповнення їх контентом, розробка веб або графічного дизайну. Фріланс ідеально підходить також і новачкам у сфері ІТ.

Ідеальний процес роботи на фрілансі виглядає так: обираєте біржу, подаєте заявку на цікаві проекти, виконуєте завдання і отримуєте гроші. На ділі все набагато складніше: важко знайти гарне замовлення і виділитися на тлі інших виконавців.

Фріланс-біржа — це сервіси, які надають можливість віддалено працювати, отримувати замовлення, виконувати складні та цікаві завдання у різних сферах, розміщувати свої послуги на онлайн-майданчиках, тим самим шукаючи потенційних замовників у пошуках виконавця для своїх проєктів.

Хоч фріланс і йде по всій планеті, залучаючи до себе все більше і більше людей, у нас в Україні, після ослаблення всіх обмежень і заборон, пошук віддаленої роботи або фріланс у травні впав, а потім різко зріс, це можна розглянути на статистиці, проведену за запитами слів «фріланс» та «віддалена робота», 2006 року на платформі Freelancehunt в Україні було зареєстровано 1,2 тис. фрілансерів, а через 15 років, 2021-го, на біржі був уже мільйон користувачів. Після початку війни в Україні фрілансерів менше не стало, навпаки, багато хто звернув увагу на такий формат роботи та почав реєструватися на біржах. Наприклад з 2020 року до травня 2022 бачимо зростання динаміки популярності, (рис. 1.1).

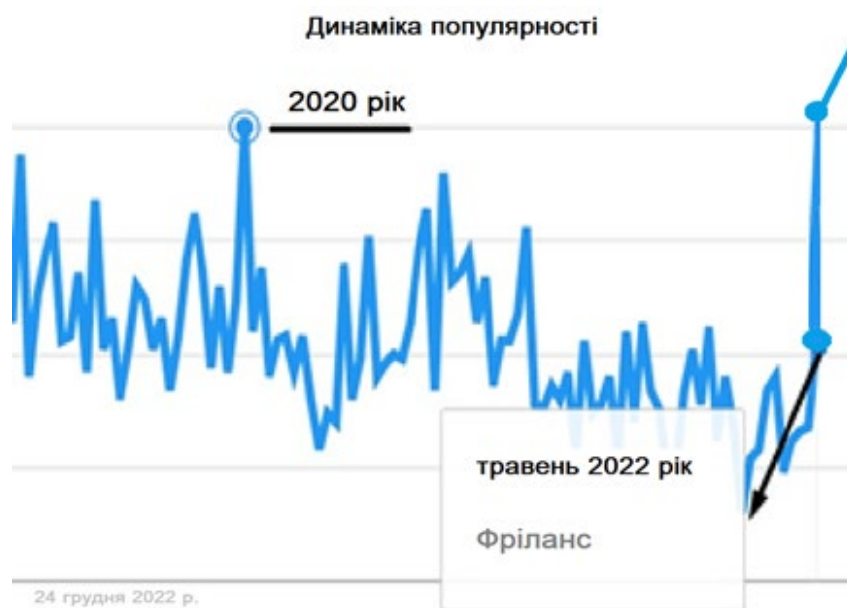


Рисунок 1.1. Статистика запитів пошуку за словом «фріланс»

Пов'язано це з тим, що на таких майданчиках досить важко заробляти багато, так само велику роль відіграє побудова довірчих відносин між клієнтом і фрілансером, хоч і портал відіграє роль посередника, але не завжди може запобігти обману. Так як в Україні дотримуються думки, що, працюючи на постійній роботі з перевіреними людьми, можна отримати набагато більше досвіду, незважаючи на те, що на фрілансі, напрацювати досвід і зібрати відмінне портфоліо набагато легше.

1.1 Аналіз з предметної області

Портали є різновидом веб-сайтів, функціонал якого набагато більше і навігація зручніше. Даний вид сайту орієнтовано на певну тематику та поєднують навколо себе користувачів з однаковими інтересами.

Портал для фрілансерів – сайт, що надає можливість віддаленої роботи розробникам, дизайнерам та іншим фахівцям, також є посередником між фрілансером та клієнтом. Пройшовши реєстрацію користувач може, скориставшись послугою сайту з розміщення контенту, виставити завдання на сайт і стати фрілансером. Виставлені завдання на сайті та вся інформація зберігається у базі даних. Потенційний клієнт, зайшовши на повну інформацію замовлення, може дізнатися як про надані послуги фрілансером, так і про

самого розробника. Після вибору завдання, що задовольняють усі умови користувача, він може написати фрілансеру перебуваючи на сайті та домовитися про ціну та умови замовлення.

Сайт створюється із застосуванням систем управління контентом. Кожен великий портал працює, вже зі спеціально розробленими системами управління та стеження за тим, що відбувається на сайті. Зазвичай середні або мінімальні проекти застосовують вільне ПЗ, наприклад як OpenCart, Drupal, WordPress тощо.

Система управління сайтом можна встановити самостійно на хостинг-майданчик, можуть бути приватною розробкою або програмним забезпеченням з тимчасовими підписками.

Існують два види порталів з пошуку віддаленої роботи:

1) Сайти з вакансіями з пошуку віддалених робіт, де роботодавці виставляють пости з вакансіями, в яких шукають працівників на віддалену роботу.

2) Сайти біржі фрілансу, де розробник сам працює на себе та виставляє на портал завдання, які він може вирішити, контролюючи свій вільний час та виконувати роботи замовників.

Існує величезна кількість проектів з біржі фрілансу, які на стільки закріпилися на цій позиції, що при одній лише згадці про фріланс, їх сайти з'являються в пошукових результатах.

На даний момент не складно знайти наш список бірж фрілансу, наприклад такі як [1]:

- Freelance.ua;
- Weblancer.net;
- Free-lance.ua;
- kabanchik.ua.

Незважаючи на те, що ми маємо свої сайти для фрілансерів, хороший фахівець не повинен обмежуватися пошуком роботи тільки по українських біржах, це буде не ефективно і неправильно.

1.1.1 Freelancehunt

Перша біржа фрілансу в Україні з'явилася 2003 року. То був Weblancer.net. Ось які українські біржі фрілансу популярні зараз [2].

Freelancehunt – одна із найбільших бірж фрілансу в Україні. На платформі є проекти у таких сферах:

- дизайн;
- розробка ПЗ;
- адміністрування;
- бекенд;
- менеджмент;
- маркетинг;
- переклад;
- контент;
- маркетинг;
- юридичні послуги;
- інші сфери.

Щодня на біржі з'являється понад тисячу активних завдань від замовників. На рисунку 1.2 представлено стартову сторінку Freelancehunt.

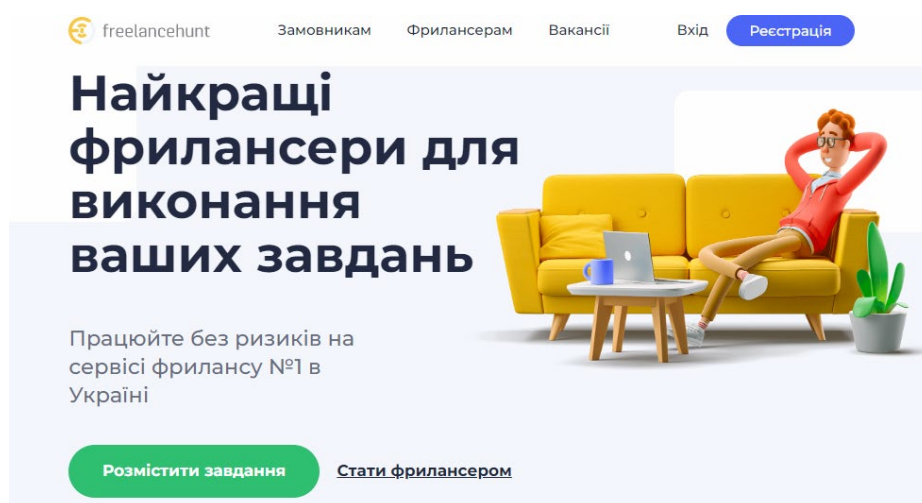


Рисунок 1.2. Стартова сторінка Freelancehunt

Ось як працює біржа:

1. Замовник виставляє проект та описує, що потрібно зробити.
2. Фрілансери роблять ставки – вказують ціну, терміни та коментарі до завдання.
3. Замовник обирає виконавця.
4. Фрілансер виконує роботу.
5. Замовник приймає та оплачує завдання.
6. Після завершення проекту замовник та фрілансер обмінюються відгуками.

Фрілансер може подати необмежену кількість заявок на проекти.

За правочини на біржі Freelancehunt стягує комісію за свої послуги. Вона становить 9% – оплату робить замовник чи фрілансер. Можна поділити її між двома учасниками, тоді вона буде 5% для кожного.

Згідно зі статистикою Freelancehunt [1]: 83,3% – працювали вдома; 7,2% – в офісі; 5,5% – поєднували роботу вдома та в офісі; 2,1% – у кафе; 0,6% - у коворкінгу .

На біржі є розділ "Конкурси", де можна розмістити конкурс на неймінг, створення логотипу чи ілюстрації. Замовник заповнює бриф, який проходить модерацию на сайті. Коли фрілансери ознайомилися із завданням, вони можуть надіслати свої варіанти на конкурс. Замовник оцінює роботи та обирає переможця.

2021 року на Freelancehunt з'явився розділ «Вакансії». Цей розділ призначений не для разових проектів із фрілансером, а для пошуку штатного співробітника. Розміщення 1 вакансії коштує 99 грн., вона активна один місяць.

4 листопада 2021 року в Україні вперше відзначили День фрілансера. Це сталося з ініціативи біржі фрілансу Freelancehunt, яку зареєстрували 4 листопада.

1.1.2 Freelance.ua

Біржа фрілансу Freelance.ua в Україні з'явилася 2006 року [1]. На рисунку 1.3 зображено головну сторінку вебсайту.

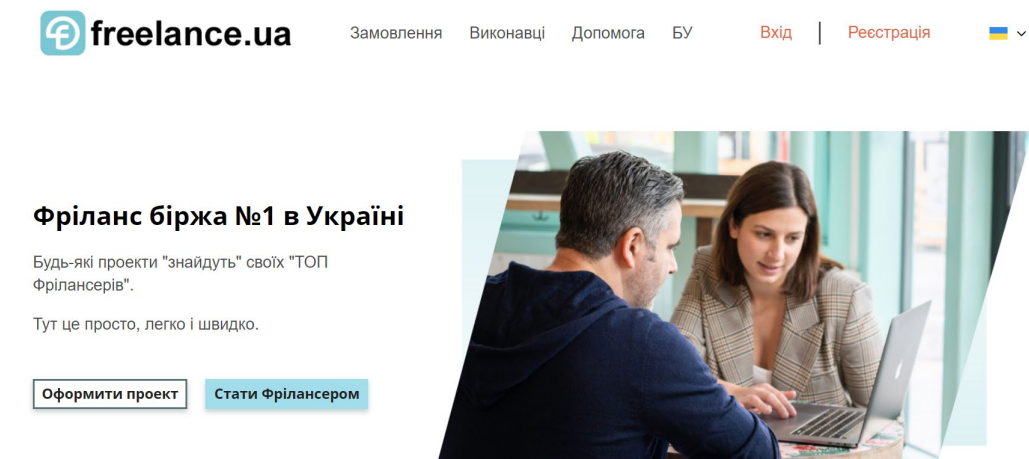


Рисунок 1.3. Стартова сторінка freelance.ua

На Freelance.ua можна шукати замовлення у таких сферах:

- веб програмування;
- дизайн сайту;
- верстка;
- логотипи;
- копірайтинг;
- відеомонтаж;
- пошукове просування;
- контекстна реклама; інші.

Біржа має кілька розділів для пошуку робіт:

- разові завдання;
- постійна робота;
- робота в офісі.

Усі сплати на біржі Freelance.ua проходять через сервіс «безпечна угода». Комісія цього сервісу – 7,5%, її сплачує замовник. На фріланс-біржі існує два види облікових записів:

- Базовий – з ним ви можете зробити одну заявку на замовлення протягом доби. Бюджет замовлення не повинен перевищувати 400 грн.
- Про – платний обліковий запис. Коштує 199 грн/місяць, 499 грн/3

місяці або 1499 грн/рік. З ним фрілансер може надсилати заявки на проекти вартістю понад 400 грн та робити необмежену кількість таких ставок.

1.1.3 Weblancer.net

Weblancer.net працює вже 18 років. Тут зареєстровано понад 1,5 млн. користувачів. На біржі є різні напрямки: веб-програмування, програмування ПЗ, тексти та переклади, графіка та фотографія, поліграфія та айдентика, архітектура та інжиніринг, адміністрування сайтів та інші (рис. 1.4) [1].

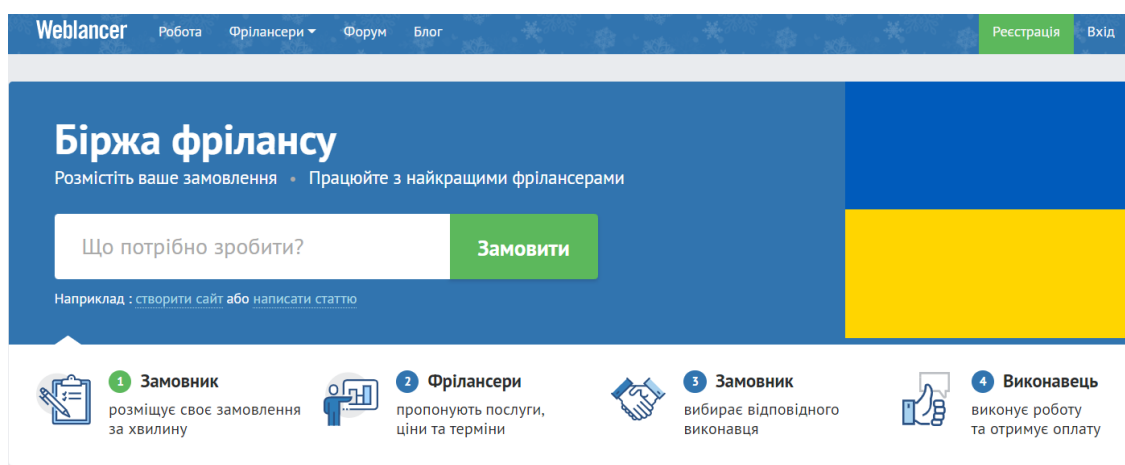


Рисунок 1.4. Стартова сторінка weblancer.net [33]

На біржі можна безкоштовно розмістити разові замовлення та конкурси, резюме та портфоліо. Після реєстрації фрілансер отримує п'ять безкоштовних заявок. Ще 50 безкоштовних заявок можна отримати, якщо заповнити профіль:

- завантажити фотографію дає +10 заявок;
- заповнити резюме – це ще 10 заявок;
- перерахувати свої послуги - також +10 заявок;
- додати роботи до портфоліо дає +20 заявок.

Додаткові заявки можна купувати двома способами:

- передплата – фрілансер може оформити передплату на три місяці.

Її ціна залежить від вибраної категорії;

– покупка - фрілансер може придбати універсальні заявки. Тарифи на них можна буде побачити після підтвердження телефонного номера.

На Weblancer.net потрібно платити на біржі 5% від виконаного замовлення. Усі суперечки розглядають професійні юристи.

1.1.4 Free-lance.ua

Найпопулярніші категорії Free-lance.ua (рис. 1.5): менеджмент, розробка сайтів, дизайн, арт, програмування, оптимізація, тексти, переклади та інші. На біржі є три варіанти робіт для фрілансерів та замовників:

- Проекти – швидке разове завдання.
- Конкурси – замовник подає завдання з певним бюджетом та отримує від фрілансерів різні варіанти виконаного проекту. Найчастіше там завдання на неймінг, логотипи та ілюстрації.
- Вакансії – фрілансера наймають до штату компанії.

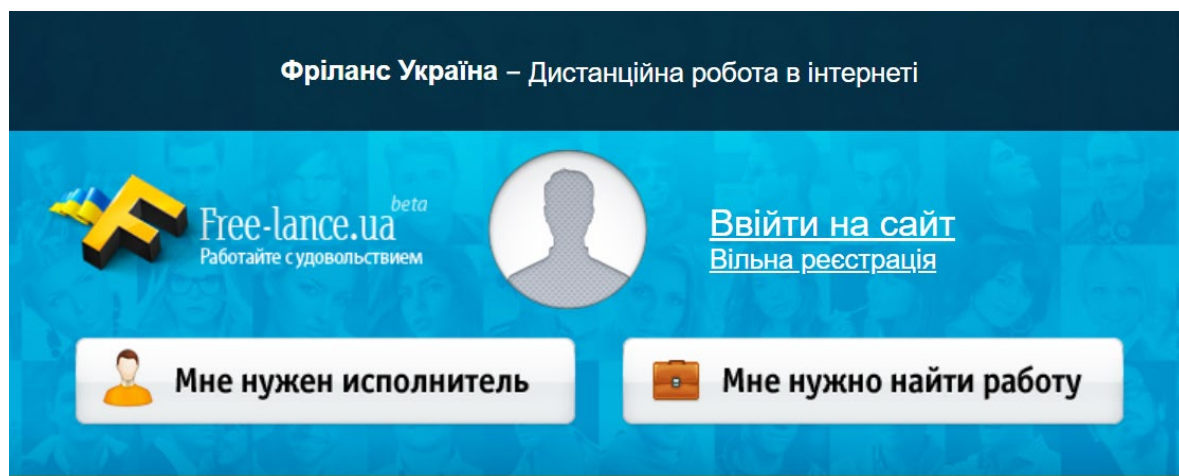


Рисунок 1.5. Стартова сторінка free-lance.ua

1.1.5 Kabanchik.ua

Kabanchik.ua (рис. 1.6) існує з 2012 року та не схожий на традиційні біржі.

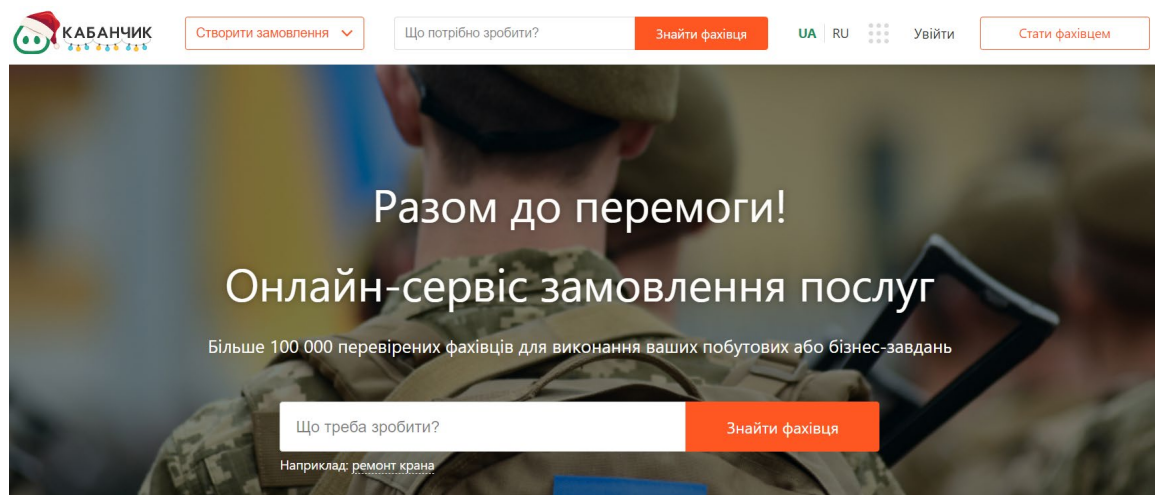


Рисунок 1.5. Стартова сторінка kabanchik.ua

Тут можна замовити через інтернет:

- оздоблювальні та будівельні роботи;
- кур'єрські та побутові послуги;
- ремонт техніки;
- авто;
- дизайн;
- організацію свят;
- репетиторство і таке інше.

На сервісі можна знайти людину, яка наглядить за вашим собакою або відремонтує домашню техніку. Окремий розділ присвячено роботі на фрілансі. Він складається з таких категорій: робота в інтернеті, розробка сайтів та додатків, дизайн, реклама в інтернеті та інші напрямки.

Сервіс має свою програму, яка доступна в Google Play і App Store. Kabanchik.ua у 2015 році купив український майданчик Prom.ua.

Українці можуть працювати і на закордонних майданчиках фрілансу. Ось найпопулярніші з них:

- upwork.com;
- fiverr.com;
- workingnomads.co/jobs;
- designhill.com;

– gigster.com.

Ринок фрілансу в Україні росте навіть під час війни. За інформацією сервісу Freelancehunt, у 2022 році на біржі виконали на 31% більше замовлень, ніж торік. Грошей в операціях порівняно з минулим роком теж побільшало на 29%.

1.2 Аналіз систем пошуку виконавців ІТ проектів

Процес отримання замовлення, зазвичай на фрілансі виглядає так: фахівець обирає біржу, подає заявку на проект, який його зацікавив, виконує роботу та отримує гроші. Насправді все складніше: нелегко знайти прийнятну роботу та виділитися на тлі інших виконавців.

Існує чимало спеціальних веб-сайтів, які допомагають фрілансерам отримати чергову роботу. Кожна така біржа має свої умови участі та гарантії оплати праці. Виконаємо аналіз чотирьох фріланс-платформ конкурентів, які мають високі позиції на ринку (табл.1.1).

Таблиця 1.1 – Порівняльна таблиця фріланс-платформ

№	Назва	Переваги	Недоліки
1	Freelancehunt	Вибір способу оплати послуг - Користувачі можуть оплачувати послуги сервісу картами, електронними грошима, крипто валютою або готівкою.	Висока конкуренція на деякі види робіт
		Прозора робота – дії співробітників Freelancehunt.com регламентовані правилами сервісу і публічним договором з надання послуг або проведення конкурсів [2]. Також опис послуг та інструкції по роботі на сервісі детально викладені в базі знань.	

№	Назва	Переваги	Недоліки
		Зворотній зв'язок – команда відповідає на всі питання і побажання клієнтів, роблячи все, щоб сервіс був комфортним і зручним. Для цього створені: форум ідей і пропозицій, паблік в соцмережах, блог про фріланс.	
		Інструменти безпеки - Сейф, Бізнес Сейф	
		Порівняно легке просування по рейтингу	
		Кілька можливостей для заробітку - проекти, конкурси і бонусна програма, в якій ви маєте можливість заробляти і отримувати бонуси, запрошуючи інших користувачів	
2	Weblancer	Якщо виникають питання на початковому етапі роботи, їх можна задати колегам «по цеху».	Обмежує той факт, що для «безпечної угоди» Введення / виведення коштів здійснюється тільки на гаманець Webmoney.
		Проекту вже багато років, це майже «стара школа» фрілансу [28]. Механізм роботи відточений, перевірений і підігнаний під вимоги виконавців та замовників, можна сказати збалансований	Висока комісія за операцію, яка викликає відчуття подвійної оплати.
		Дана біржа фріланса в Україні, Росії та інших країнах СНД має прив'язку до IP-адреси.	За можливість виконувати проекти на даній біржі потрібно будесплачувати певну комісію.

№	Назва	Переваги	Недоліки
			Досить своєрідна робота адміністрації по відношенню до клієнтів.
3	Freelancer	На цій платформі можливо знайти фрілансерів з усього світу, якщо потрібно працювати з наявністю часового поясу чи ні.	Багато неякісних проектів із надмірними вимогами та незначною віддачою.
		Гарна служба підтримки, проста у використанні та використанні [27]. Проблеми можуть бути вирішені за лічені секунди.	Велика кількість фейкових проектів на даному сервісі.
		Можливість виконувати безпечні платежі.	Автоматизовані заявки фрілансерів, у результаті чого важко зрозуміти, може фрілансер виконати цю роботу чи ні.
		Легко спілкуватися з поточними та майбутніми клієнтами за допомогою чату на сторінці.	
		Тести на компетентність дають клієнтам можливість зрозуміти, чим насправді виділяється фрілансер.	
4	Upwork	Більш високооплачувані проекти. У порівнянні з бюджетними фріланс-сайтами, такими як Fiverr або Freelancer.com, клієнти UpWork частіше платять вищі тарифи за послуги.	Крім оплати за подання пропозицій, UpWork також стягує відсоток за кожен проект на основі ваших загальних рахунків. Плата за послуги коливається від 5% до 20% залежно від заробітку виконавця.
		Однією з основних переваг UpWork є те, що платежі вбудовуються в систему одним натисканням кнопки.	На UpWork існує велика конкуренція.

№	Назва	Переваги	Недоліки
		Нова структура підвищує і шанси на найм: UpWork має нову структуру, де стягується з фрілансерів невелика плата за подання пропозицій до проектів [14]. Завдяки меншій кількості загальних пропозицій та більш якісним фрілансерам, ця нова структура збільшує шанси на працевлаштування.	На перший погляд, UpWork здається досить сучасним веб-сайтом. Але коли почнеш переглядати та шукати проекти, швидко з'ясується, наскільки застарілий користувацький досвід.
		Віддалена робота - це величезна вигода від найму на UpWork.	

Кожна із існуючих інформаційних фріланс-платформ забезпечує міст між традиційними продажами та новою цифровою економікою.

Проаналізувавши існуючу інформацію, треба виділити розділи для подальшої праці над своєю інформаційною платформою.

Перше за все, потрібно зробити наголос на зручність та надійність у використанні. Візуальна частина є особливою частиною у розробці веб-сайту. Оскільки, це те, що має як і привернути увагу та зацікавити так і змусити користувачів піти зі сторінки.

Друге – це відсоток за використання інформаційної платформи. На поточному етапі не планується стягування плати за реєстрацію на створюваному веб-сайті, але ця функція можливо буде реалізована потім. Відсоток не повинен бути занадто великим для користувачів, оскільки даний аспект теж впливає на кількість користувачів та загальну активність.

1.3 Висновки до розділу 1

Взявши до уваги та проаналізувавши сучасні вітчизняні сайти, можна виділити наступний основний функціонал, який пов'язує їх усіх та недоліки, які так само їх поєднують.

Загальний основний та базовий функціонал сайтів з фрілансу, полягає в наявності:

- реєстрації та авторизації;
- перегляд локальної інформації на тему порталу;
- проста та зручна навігації по сайту;
- пошукова система;
- створення завдань та замовлень для фрілансерів та клієнтів;
- перегляд завдань та замовлень;
- можливість писати відгуки до замовлень;
- особисті повідомлення між користувачами сайту.

Також можна виділити незначні та грубі недоліки на даних сайтах, а саме:

- складна навігація сайтом;
- складний та в деяких місцях кривий інтерфейс;

занадто багато умов для початку роботи, такі як: передплата, введення зайвих даних, багато з'являється реклами та інформації.

Для опрацювання магістерського проєкту варто приділити високу увагу вимогам до ІС, та визначити такі вимоги до системи:

- функціональні вимоги;
- вимоги до інтерфейсу;
- вимоги до продуктивності;
- вимоги до безпеки.

Описуючи переваги, треба звернути увагу на комунікацію. Інформаційна система для фріланс біржі повинна не лише надати користувачам підтримувати постійний зв'язок один з одним, але й надати підтримку з боку адміністрації сервісу. Дана функція буде реалізована через чат.

РОЗДІЛ 2

АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Існує чимало платформ для людей, які можуть знайти роботу чи отримати допомогу в різних видах діяльності, через збільшення кількості таких платформ та очевидну конкуренцію, яка виникає, багато доступних робочих місць, які не досягають рівня прожиткового мінімуму, що може бути важким для багатьох людей. Пошук потрібного фахівця може бути дуже складним завданням. Це вимагає глибокого знання культури організації та чіткого розуміння особистості кандидата, сильних сторін, інтересів, стилю роботи та інших характеристик. Такі сервіси, які надають можливість знайти кваліфікованого фахівця і можуть привести їх до вигідних взаємин і партнерських угод в нашій країні існують та розробляються нові портали з більшими можливостями з використанням сучасних засобів. Проведемо аналіз найсучасніших технологій створення веб-орієнтованої інформаційної системи.

2.1. Аналіз програмних веб технологій для розробки пошукової інформаційної системи

Сучасні технології створення та підтримки веб-сайтів орієнтовані на платформи, що дозволяють ефективно керувати інформаційним наповненням і даними, які надходять від відвідувачів сайту. Як правило, такі рішення базуються на серверних технологіях типу ASP, ASP.NET, JSP, PHP або використовують готові потужні засоби для створення корпоративних сайтів, що орієнтовані на впровадження вказаних технологій.

Розглянемо детальніше популярні інформаційні технології.

Створення веб-сторінок за фрагментами серверного коду є технологією ASP, ASP.NET (Active Server Pages) [15]. Це розроблена Microsoft комерційно доступна технологія, за допомогою якої веб-майстер може самостійно

формувати динамічно поновлювальні веб-сторінки. Характерною особливістю такої технології є можливість відокремлення функціональної частини розробки від процесів створення дизайну. ASP-сторінки можуть містити HTML-текст, змішаний зі сценаріями мов JavaScript і VBScript. У процесі обробки запиту нової сторінки його виконує сервер і динамічно генерує браузеру потік HTML-тексту, що відображується на екрані монітора. ASP-технологія Microsoft набула подальшого розвитку у технологіях JSP, PHP та ін.

Технологія JSP (Java Server Pages) – це технологія створення серверних сторінок Java. Специфікація JSP є розширенням Java Servlet API для генерації динамічних веб-сторінок на веб-сервері. Така крос-платформа є альтернативою технології ASP корпорації Microsoft. Специфікація Sun за назвою JSF (Java Server Faces) реалізує технологію JSP, що описує правила створення веб-додатків зі зручним для користувача інтерфейсом та орієнтована на розробку серверних компонентів створення інтерфейсу [3].

Однією з перших технологій створення веб-застосунків, які виконуються сервером, була Common Gateway Interface (CGI) технологія [10]. Вона дозволила розробку і виконання серверних застосунків, звернення до яких відбувається за допомогою зазначеного в URL імені (та параметрів). Залежно від обраного протоколу вхідною інформацією таких веб-додатків вважають безпосередньо код HTTP-заголовка або запит пошукової системи. CGI-застосування – це консольні додатки, які генерують HTML-код, переданий браузеру. Подібні застосування можуть являти собою код на скриптових мовах, який інтерпретується на сервері. Крім того, CGI-застосування презентують робочий файл, котрий можна створити за допомогою будь-якого засобу розробки, що генерує консольні застосування для операційної системи, під керуванням якої функціонує веб-сервер.

Серед інших популярних технологій, які реалізують створення вебсторінок із фрагментами коду, виконуваного на сервері, виділимо

некомерційну, вільно розповсюджену технологію PHP (Personal Home Pages) [25]. Ця технологія заснована на використанні CGI-застосувань, що інтерпретують впроваджений у HTML-сторінку код на скриптовій мові. Головною особливістю мови PHP є її практичність. PHP надає програмісту інструмент для швидкого й ефективного вирішення поставлених завдань. Вона вирізняється винятковою гнучкістю до потреб розробника. Хоча PHP традиційно рекомендують використовувати у поєднанні з HTML-кодом, проте PHP з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови Інтернет-програмування.

Результати критеріального порівняння технологій розробки та впровадження веб-ресурсів зведено в табл. 2.1. Розглянуті технології забезпечують сучасну функціональність, ефективний супровід процесів створення сайтів та їх наповнення інформаційними ресурсами.

Таблиця 2.1 – Порівняння сучасних технологій розробки веб-сайтів

	PHP	JSP	ASP.NET
Багатоплатформеність	+	+	-
Продуктивність	+/-	+/-	+
Простота використання	+	+/-	+/-
Наявність доступних програмних бібліотек	+	+	+
Розподіл дизайну та логіки	+/-	+/-	+

Результати проведеного аналізу дозволяють зробити висновок про наявність переваг експлуатаційних характеристик PHP-технології. PHP користується значною популярністю серед вебпрограмістів і сьогодні вважається мало чи не найпопулярнішою мовою для створення вебдодатків (скриптів). Головними перевагами PHP вбачаємо практичність, легкість у застосуванні, ефективність, продуктивність та гнучкість.

2.2 Аналіз засобів реалізації баз даних у веб-технологіях

Робота з базами даних є однією з головних складових процесу програмування сайту динамічного типу. Базы даних для сайтів використовують з метою зберігання різнопланової інформації. Спрощено бази даних являють собою певний набір взаємозалежних таблиць. Розміри таблиць у базах є різними, а їх кількість – довільною. Базы даних на сервері акумулюють необхідну для роботи сайту інформацію статистичного характеру. До найпопулярніших мережево орієнтованих баз даних відносять SQL, MySQL, Oracle Database та інші. Вибір потрібної системи керування базою даних (СКБД) обумовлюється вимогами до інформаційних характеристик та функціональних можливостей системи. Однією з найпоширеніших систем керування базами даних у наш час вважається MySQL [22], яка є альтернативою комерційним системам. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування. MySQL – компактний багатопоточний сервер баз даних. Серед переваг сервера MySQL виділимо простоту у встановленні та використанні, підтримку паралельної роботи значної кількості користувачів. Кількість рядків у таблицях даних може сягати 50 млн. Висока швидкість виконання команд, наявність простої та ефективної системи безпеки обумовлюють зростання популярності MySQL. Проте основною перевагою СКБД MySQL є можливість її безкоштовного використання. Об'єктно-реляційна система керування базами даних компанії Oracle (Oracle Database) орієнтована під операційні системи Windows, Unix, Linux та MacOS. Oracle Database, на відміну від MySQL, має більш широку сферу застосування [30]. СКБД Oracle широко відома як у нашій країні, так і у світі. На ній базується безліч сучасних інформаційних систем. Однак, у процесі програмування сайтів для невеликих і середніх компаній через свою високу вартість вона використовується нечасто. Крім того, досить складно знайти хостинг з підтримкою такої СКБД.

SQL (англ. Structured Query Language – мова структурованих запитів) –

декларативна мова програмування засобів інтерфейсної взаємодії користувача з базами даних, що реалізує процеси формування запитів, оновлення і керування реляційними базами даних, створення схеми бази даних і її модифікації, систему контролю за доступом до інформаційних ресурсів. SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати в якості інструкцій для керування даними. Крім того, стандарт SQL утримує функції визначення процесів зміни, перевірки і захисту даних.

Табл. 2.2 акумулює результати критеріального аналізу сучасних СКБД.

Таблиця 2.2. – Порівняльна характеристика СКБД

	SQL	Oracle	MySQL
Надійність	+	+	+
Швидкодія	-	+	+
Простота	-	-	+
Зручність використання	+/-	+	+

Проаналізувавши базові характеристики розглянутих СКБД, можна зробити висновок про зручність використання засобів MySQL для створення динамічних веб-сторінок, оскільки сервер MySQL забезпечується підтримкою з боку різноманітних мов програмування, зокрема і PHP. MySQL задовольняє основні вимоги до СКБД у процесі розробки веб-сайту, так як вона є надійною, має високу швидкість, просту у встановленні та зручну у використанні. Завдяки своїй доступності, швидкості та безпеці MySQL забезпечує ефективний доступ до баз даних з Internet.

2.3 Мова розмітки вебдокументів HTML

Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузерами і відображається у вигляді документа в зручній для людини формі. Мова HTML до п'ятої версії є

додатком SGML (стандартної узагальненої мови розмітки) і відповідає міжнародному стандарту ISO 8879. У свою чергу XHTML є більш строгим варіантом HTML, він наслідує всі обмеження XML і, фактично, XHTML можна сприймати як додаток мови XML до області розмітки гіпертексту [31].

З появою HTML5 вводять кілька нових елементів і атрибутів, які відображають типове використання розмітки на сучасних веб-сайтах. Деякі з них - семантичні заміни для використання універсальних блокових (`<div>`) і малих (``) елементів, наприклад, `<nav>` (блок навігації по сайту), `<footer>` (зазвичай відноситься до нижньої частини сторінки або останньому рядку HTML коду) або `<audio>` і `<video>` замість `<object>` [31].

На додаток до визначення розмітки HTML5 встановлює API, який може бути використаний з JavaScript. Можливості DOM (Document Object Model - «об'єктна модель документа» - це незалежний від платформи і мови програмний інтерфейс, що дозволяє програмам і скриптам отримати доступ до вмісту HTML-, XHTML- і XML-документів) розширені і фактично використані властивості задокументовані. XHTML5 - це XML-серіалізація мови HTML5. Документи XML повинні бути забезпечені XML Internet media type. XHTML5 вимагає суворого і правильно оформленого синтаксису XML. Вибір між HTML5 і XHTML5 зводиться до вибору типу MIME / вмісту: тип медіа, який буде обраний, визначить, який тип документа повинен бути використаний [31].

2.4 Каскадні таблиці стилів – CSS

CSS (Cascading Style Sheets - каскадні таблиці стилів) - формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки переважно використовується як засіб опису, оформлення зовнішнього вигляду веб-сторінок [32].

CSS використовується розробниками веб-сторінок для завдання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду цих веб-сторінок. Основною метою

розробки CSS було розділення опису логічної структури веб-сторінки (яке проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (яке тепер проводиться за допомогою формальної мови CSS) [33].

Правила CSS пишуться на формальній мові CSS і розташовуються в таблицях стилів, тобто таблиці стилів містять у собі правила CSS. Ці таблиці стилів можуть розташовуватися як у самому веб-документі, зовнішній вигляд якого вони описують, так і в окремих файлах, що мають формат CSS. Тобто ці таблиці стилів можуть бути підключені, впроваджені в описуваний ними веб-документ чотирма різними способами) [31].

1. Коли таблиця стилів знаходиться в окремому файлі, вона може бути підключена до веб-документу за допомогою тега `<link>`, розташованого в цьому документі між тегами `<head>` і `</head>`. Усі правила цієї таблиці діють протягом усього документа. Цей спосіб підключення показаний у коді) [33]:

```
<!doctype html>
<html>
<head>
.....
  <link rel="stylesheet" href="style.css">
</head>
<body>
.....
</body>
</html>
```

2. Коли таблиця стилів знаходиться в окремому файлі, вона може бути підключена до веб-документу за допомогою директиви `@import`, яка розташовується в цьому документі між тегами `<style>` і `</style>` та яка вказує у своїх дужках після слова `url` на адресу цієї таблиці стилів, як у коді нижче. Усі правила цієї таблиці діють протягом усього документа) [32].

```
<!doctype html>
<html>
  <head>
.....
    <style media="all">
      @import url(style.css);
```

```

        </style>
    </head>
</html>

```

3. Коли таблиця стилів описана в самому документі, вона може розташовуватися в ньому між тегами `<style>` і `</ style>`. Цей спосіб показаний у коді далі. Усі правила цієї таблиці діють протягом усього документа [33].

```

<!doctype html>
<html>
    <head>
        ....
        <style>
            body {
                color: red;
            }
        </style>
    </head>
    <body>
        ....
</body>
</html>

```

4. Коли таблиця стилів описана в самому документі, вона може розташовуватися в ньому в тілі якогось окремого тега цього документа. Усі правила цієї таблиці діють тільки на вміст цього тега. Цей спосіб продемонстрований у коді [31]:

```

<!doctype html>
<html>
<head>
    ....
</head>
<body>
    <p style="font-size: 20px; color: green; font-family: arial,
helvetica, sans-serif">
        ....
    </p>
</body>
</html>

```

У перших двох випадках до документа застосовані зовнішні таблиці стилів, а в інших двох – внутрішні таблиці стилів.

Для додавання CSS до XML-документу останній повинен містити спеціальне посилання на таблицю стилів. Наприклад:

```
<?xml-stylesheet type="text/css" href="style.css"?> [33].
```

До появи CSS оформлення веб-сторінок здійснювалося виключно засобами HTML безпосередньо всередині вмісту документа. Однак з появою CSS стало можливим принципове розділення змісту й представлення документа. За рахунок цього нововведення стало можливим легке застосування єдиного стилю оформлення для маси схожих документів, а також швидка зміна цього оформлення.

Переваги CSS полягають у наступному.

1. Кілька дизайнів сторінки для різних пристроїв перегляду. Наприклад, на екрані дизайн буде розрахований на велику ширину, під час друку меню не виводитиметься, а на ПК і стільниковому телефоні меню буде йти за вмістом.

2. Зменшення часу завантаження сторінок сайту за рахунок перенесення правил представлення даних у окремий CSS-файл. У цьому випадку браузер завантажує тільки структуру документа та дані, що зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і може бути закешоване (збережене).

3. Простота подальшої зміни дизайну. Не потрібно правити кожну сторінку, а лише змінити CSS-файл.

4. Додаткові можливості оформлення. Наприклад, за допомогою CSS-верстки можна зробити блок тексту, який буде обтікати зовнішній текст, або зробити так, щоб меню було завжди видно при прокручуванні сторінки [33].

Водночас у CSS існують недоліки.

1. Різне відображення верстки в різних браузерах (особливо застарілих), які по-різному інтерпретують одні й ті самі правила CSS.

2. Часто зустрічається необхідність на практиці виправляти не тільки один CSS-файл, але й теги HTML, які складним способом пов'язані із

селекторами CSS, що іноді зводить нанівець простоту застосування єдиних файлів стилів і значно подовжує час редагування й тестування [33].

Багато веб-майстрів для кросбраузерності стилів не використовують нововведення в CSS3, замінюючи їх зображеннями. Наприклад, замість заокруглення кутів використовують фонове зображення, на якому змальований цей блок без змісту (тексту) з закругленими кутами. «CSS3 – розширення CSS 2.1 – додає потужну функціональність до існуючих можливостей» [31, с. 16].

У стандартах CSS від Консорціуму W3C використовується модель, у якій властивість `width` визначає ширину вмісту блоку, не включаючи до неї відступи й рамки. Відмінності в реалізації CSS різними браузерами змушують веб-розробників шукати рішення, як примусити всі браузери відображати сторінку однаково. CSS-фільтри дозволяють вибірково застосовувати стилі до різних елементів.

Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне уявлення, друковане подання, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля. Як відомо, HTML-документи будуються на підставі ієрархії елементів, яка може бути наочно представлена в деревовидної формі. Елементи HTML один для одного можуть бути батьківськими, дочірніми, елементами-предками, елементами-нащадками, сестринськими .

Елемент є батьком іншого елемента, якщо в ієрархічній структурі документа він знаходиться відразу, безпосередньо над цим елементом. Елемент є предком іншого елемента, якщо в ієрархічній структурі документа він знаходиться десь вище цього елемента.

В CSS можуть задаватися за допомогою селекторів не лише поодинокі елементи, але і елементи, які є нащадками, дочірніми або сестринськими елементами інших елементів.

2.5 Динамічна мова JavaScript

Для реалізації додаткової анімації та взаємодії зареєстрованих осіб на сайті буде використовуватись JavaScript. На даний момент часу вона має кілька відомих фреймворків [34].

JavaScript (JS) - прототипна-орієнтована скриптова мова програмування. Є діалектом мови ECMAScript. JS зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків [18].

Найбільш широке застосування знаходить в браузерях як мова сценаріїв для додання інтерактивності веб-сторінок. Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу. На JS вплинули багато мов, при розробці була мета зробити мову схожим на Java, але при цьому легким для використання не програмістів. Мовою JS не володіє будь-яка компанія або організація, що відрізняє його від ряду мов програмування, використовуваних в Інтернет [34].

Мова JS використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосувань (ReactJS, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосувань (Electron, NW.js);
- мобільних застосувань (React Native, Cordova);
- сценаріїв у прикладному ПЗ (наприклад, у програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів та ін. [34].

JS наразі є однією з найпопулярніших мов програмування в інтернеті. Але спочатку багато професійних програмістів скептично ставилися до мови,

цільова аудиторія якої складалася з програмістів-аматорів. Поява AJAX змінила ситуацію та повернула увагу професійної спільноти до мови, а подальші модифікації мови за стандартами ES2015 та ES2017 внесли багато корисних можливостей, яких не вистачало для ефективного програмування. „Уперше про AJAX заговорили після появи в лютому 2005 року статті Джесі Джеймса Гарретта (Jesse James Garrett) «Новий підхід до Web-додатків»” [14, с. 44]. У результаті були розроблені та покращені багато практик використання JS (зокрема, тестування та налагодження), створені бібліотеки та фреймворки, поширилося використання JS поза браузером. «Суть технології AJAX, як це і випливає з її назви, полягає в створення запитів сервера, які виконуються в асинхронному режимі. Це означає, що серверна частина веб-додатків, що використовує AJAX, працює незалежно від клієнтської частини» [34, с. 429].

«Оскільки JavaScript є мовою, що інтерпретується, дуже часто він позиціонується як мова сценаріїв, а не як мова програмування, при цьому мається на увазі, що мови сценаріїв простіше й більшою мірою орієнтовані не на програмістів, а на звичайних користувачів» [34, с. 21].

Назва AJAX – це акронім, що розкривається як Asynchronous JS and XML і що означає асинхронний JS і XML. «Якщо ця назва, на ваш погляд, мало про що говорить, ми погодимося з вами. Простіше кажучи, можна вважати, що AJAX – це «JavaScript з розширеними правами», оскільки за своєю суттю ця технологія є сценаріями на мові JavaScript, які в міру необхідності у фоновому режимі виконують запити до сервера і отримують додаткові дані, оновлюючи окремі частини сторінки й тим самим виключаючи необхідність повторного її завантаження цілком» [31, с. 24].

JS має низку властивостей об’єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об’єктів у ній відрізняється від традиційних мов об’єктно-орієнтованого програмування (ООП). Крім того, JS має ряд властивостей, притаманних функціональним мовам, – функції як об’єкти

першого класу, об'єкти як списки, каррінг, анонімні функції, замикання (closures), – що додає мові додаткову гнучкість.

JS має C-подібний синтаксис, але в порівнянні з мовою Cі має такі докорінні відмінності:

- об'єкти з можливістю інтроспекції та динамічної зміни типу через механізм прототипів;
- функції як об'єкти першого класу;
- обробка винятків;
- автоматичне наведення типів;
- автоматичне прибирання сміття;
- анонімні функції [34].

JS містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JS містить набір вбудованих операцій, які не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм [10]. Синтаксис JS в основному відповідає синтаксису мови Java (тобто, зрештою, успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення.

2.5.1 *jQuery*

jQuery – бібліотека JS, що фокусується на взаємодії JS і HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними. jQuery надає зручний API для роботи з AJAX [19].

Точно так же, як CSS відокремлює візуалізацію від структури HTML, JQuery відокремлює поведінку від структури HTML. Наприклад, замість прямої вказівки на обробник події натискання кнопки управління передається JQuery, яка ідентифікує кнопки і потім перетворює його в обробник події кліка. Такий поділ поведінки і структури також називається принципом ненав'язливого JS [20].

Бібліотека jQuery містить функціональність, корисну для максимально

широкого кола завдань. Проте, розробниками бібліотеки не ставилося завдання суміщення в jQuery функцій, які підійшли б усюди, оскільки це призвело б до великого коду, велика частина якого не затребувана. Тому була реалізована архітектура компактного універсального ядра бібліотеки і плагінів. Це дозволяє зібрати для ресурсу саме ту JS-функціональність, яка на ньому була б затребувана [26].

jQuery надає допоміжний метод `post`, призначений для відправки даних сервера. Метод `post` отримує кілька параметрів (у числі яких URL-адреса, за якою відправляється інформація), передану інформацію та функцію-обробник, яка виконується після завершення відправки POST [19, с. 363].

Синтаксис jQuery розроблено, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій і розробки AJAX-застосувань. jQuery також надає можливості для розробників у створенні плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня [34].

Плюси jQuery:

- малий розмір дистрибутива;
- низький поріг входження, вичерпна документація в інтернеті;
- лаконічний синтаксис;
- легко розширюваний синтаксис.

Мінуси jQuery:

- уповільнена робота програм;
- проблеми сумісності з браузером.

2.6 Висновки до розділу 1

В розділі проведено дослідження сучасних технологій розробки вебзастосувань: CSS3, MySQL., JS і PHP , для яких застосовується підхід вбудовування програмного коду в шаблони HTML-сторінок. При запиті шаблони заповнюються динамічним вмістом, зазвичай створюваним

інтерпретованою мовою сценаріїв. Підхід на основі інтерпретованих сценаріїв надзвичайно зручний при розробці вебдодатків. При розробці складних програмних систем варіант вбудовування коду в шаблони сторінок ускладнює взаємодію між компонентами і ускладнює реалізацію складної архітектури.

Аналіз показав, що кожна з представлених технологій має переваги і недоліки, і, як наслідок, свою область застосування. Тому поставлене завдання розробити пошукову інформаційну систему для фріланс біржі, будемо вирішувати обраними технологіями програмування: PHP; MySQL; HTML; CSS; JS; JQuery.

РОЗДІЛ 3

АРХІТЕКТУРА ТА ОПИС РОЗРОБКИ

Метою проекту є створення веб-орієнтованої інформаційної системи пошуку виконавців ІТ-проектів, що буде корисною як для замовників тих або інших ІТ проектів, так і для виконавців. Даний сервіс повинен сприяти пошуку замовлень та комунікацію між клієнтами та фрілансерами. Перевагами співпраці з віддаленими фахівцями є і більш низькі ціни на послуги, і можливість проглянути портфоліо фахівця, відгуки від інших замовників, рейтинг та статистику у профілі. Всі ці дані формуються в процесі роботи та їх неможливо підробити. У розділі описано моделі даних, концептуальні схеми та роботу розробленого програмного забезпечення.

3.1 Вимоги до проекту веб-орієнтованої ІС

Для реалізації даного проекту сформовано вимоги до створюваної системи. Цільовою аудиторією веб-орієнтованої інформаційної системи пошуку виконавців ІТ-проектів є:

- програмісти;
- дизайнери;
- SEO-фахівці;
- фахівці з відео та аудіо монтажу;
- фахівці по роботі з текстом (копірайтери, рерайтери);
- менеджери інтернет-проектів;
- адміністратори соціальних мереж;
- розробники мобільних додатків.

Розроблений веб-сайт буде використаний як відкрита біржа для різних ІТ проектів.

Функціональні вимоги до веб-орієнтованої інформаційної системи пошуку виконавців ІТ-проектів:

- Реєстрація та авторизація користувачів.

- Розміщення на сайті замовлень з можливістю прикріплення файлів.
- Перегляд замовлень зареєстрованими виконавцями та клієнтами.
- Розробка пошукової системи для формування списку робіт за різними критеріями.
- Створення заявок на одержання замовлення з можливістю прикріплення файлів.
- Можливість спілкування між клієнтом та робітником за допомогою чату.
- Формування рейтингу та історії проектів.
- Формування оповіщень.
- Блокування користувачів.
- Сервіс підтримки користувачів.

Для роботи з веб-орієнтованою інформаційною системою передбачено три групи користувачів, які повинні пройти авторизацію: адміністратори інформаційної системи (ІС), замовники, та виконавці.

Адміністратори ІС:

- блокувати користувачів;
 - відповідати на питання користувачів.
- Замовники:
- можуть створювати завдання з різноманітною роботою;
 - переглядати, редагувати та видаляти свої завдання;
 - спілкуватися з виконавцем роботи;
 - оцінювати роботи в портфоліо виконавців;
- Виконавці:
- перегляд існуючих завдань на сайті;
 - створення заявки на отримання роботи;
 - спілкуватися з замовником роботи;
 - оцінювати роботи в портфоліо інших виконавців;
 - створювати акаунт типу «Команда»;
 - генерація резюме.

Головною особливістю даного сервісу є можливість створення акаунту,

як для особистого використання, так і для компанії. Даний тип акаунту дозволяє приєднувати людей до власної команди. Дана специфіка розширює можливості роботи та підвищує ймовірність виконавця отримати замовлення й розвиватися в цілому. Крім того, полегшується робота для замовника з вибором виконавця. Можна отримати повний пакет послуг витративши на пошуки менше зусиль.

Інша відмінність – конструктор для генерації резюме відразу на сторінці профілю. Дана можливість надається лише виконавцю, базуючись на вміннях та досвіду роботи.

На відміну від резюме, ведення блогу буде доступно як виконавцю, так і замовнику. Кожен бажаючий зможе робити різні публікації на своєму профілі, переглядати інші теми які їх зацікавили та залишати свої коментарі.

3.2 ER-діаграма

Модель зв'язку сутностей (ERM) у розробці програмного забезпечення є абстрактним і концептуальним представленням даних. Моделювання зв'язків сутностей - це метод моделювання бази даних реляційної схеми, який використовується для створення певного типу концептуальної схеми або семантичної моделі даних системи, часто реляційної бази даних, і її вимог у низхідному порядку.

Діаграма зв'язку сутностей - це спеціальна модель, яка ілюструє взаємозв'язок між сутностями в базі даних. ER-діаграми часто використовують фігури для представлення 3 різних типів інформації:

Сутність (рис. 3.1) - це елементарний базовий будівельний блок для зберігання інформації про бізнес-процес. Сутність представляє об'єкт, визначений в інформаційній системі, про який потрібно зберігати інформацію.

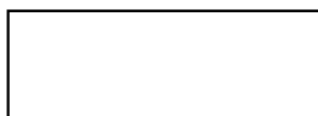


Рис. 3.1. Позначення сутності в ER-діаграмах

Зв'язок (рис. 3.2) - це іменована асоціація між сутностями. Використовується для зв'язування двох чи більше сутностей із деякими спільними атрибутами чи значущою взаємодією між об'єктами.

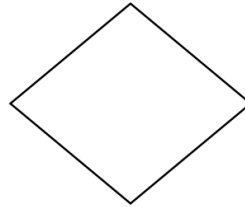


Рис. 3.2. Позначення зв'язку в ER-діаграмах

Атрибути (рис. 3.3) - це властивості сутностей і зв'язків, Дескриптор сутностей. Атрибути - це елементарні фрагменти інформації, прикріплені до сутності.

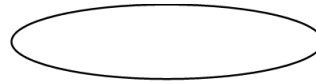


Рис. 3.3. Позначення атрибуту в ER-діаграмах

Користуючись зазначеними вище фігурами була побудована ER-діаграма для розробленого продукту (рис. 3.4)

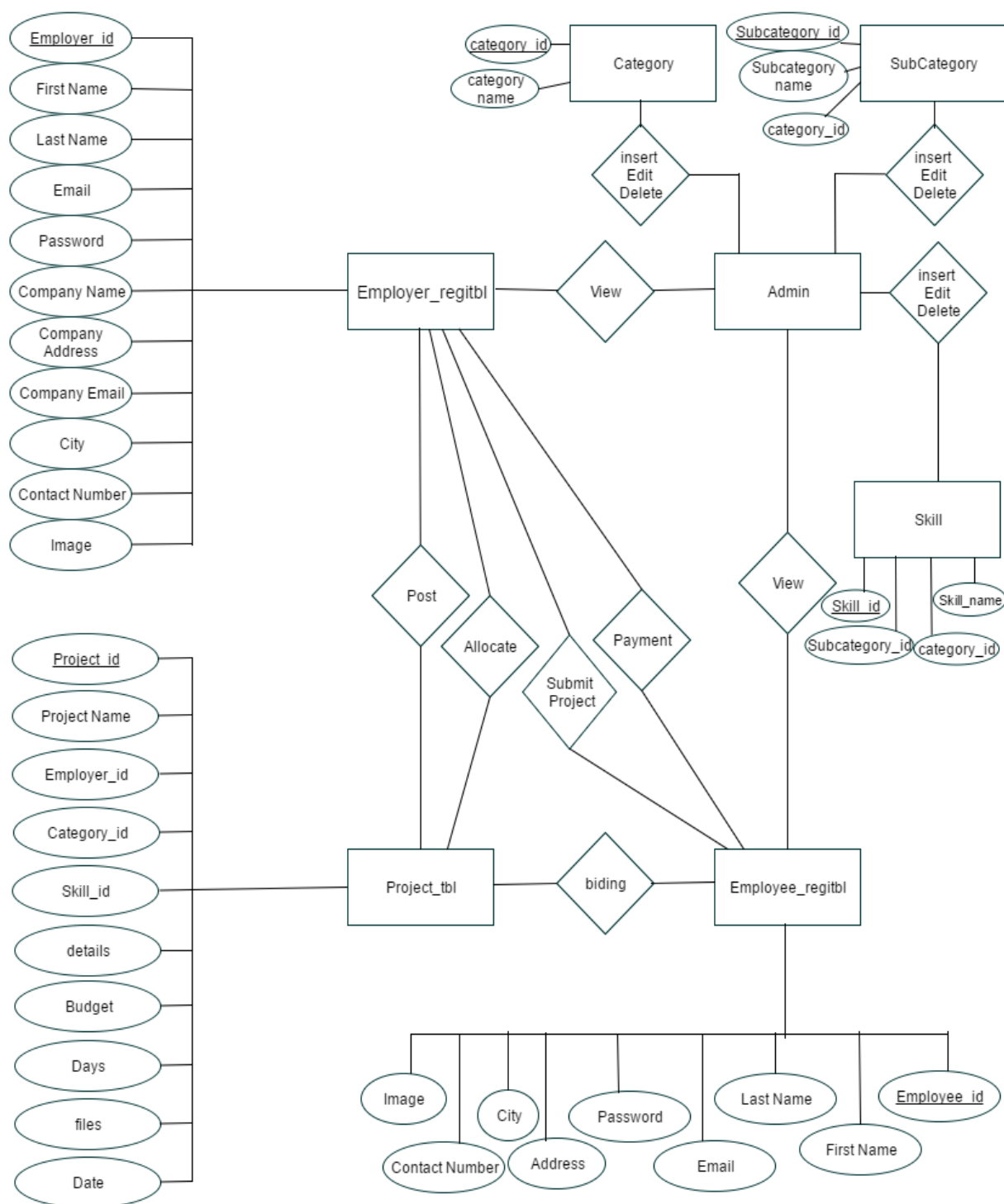


Рис. 3.4. ER-діаграма додатку

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_regtbl - 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
cnl	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 проекту.

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 проекту
cate_id	int(11)	Not Null	Ідентифікатор категорії
skill_id	int(11)	Not Null	Ідентифікатор навичок
Details	text	Not Null	Деталі проекту
Budget	text	Not Null	Бюджет проекту
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 проект 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 проекту
Budget	int(11)	Not Null	Бюджет проекту
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 проекту
Username	text	Not Null	Ім'я користувача
Message	text	Not Null	Повідомлення

– Bidtbl - таблиця використовується для зберігання всієї інформації про тендери на проекти, 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 проекту
Days	int(11)	Auto increment	Днів до завершення
project_id	int(11)	Not Null	Ідентифікатор проекту
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 проекти.

Таблиця 3.13. Структура таблиці Completeprj

Поле	Тип	Обмеження	Опис
completeprj_id	int(11)	Auto increment	Унікальний ідентифікатор
employer_id	int(11)	Not Null	Ідентифікатор роботодавця
employee_id	int(11)	Not Null	ID робітника
Prjfile	text	Not Null	Комп. файл проекту
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 проекти, 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 проекту
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.

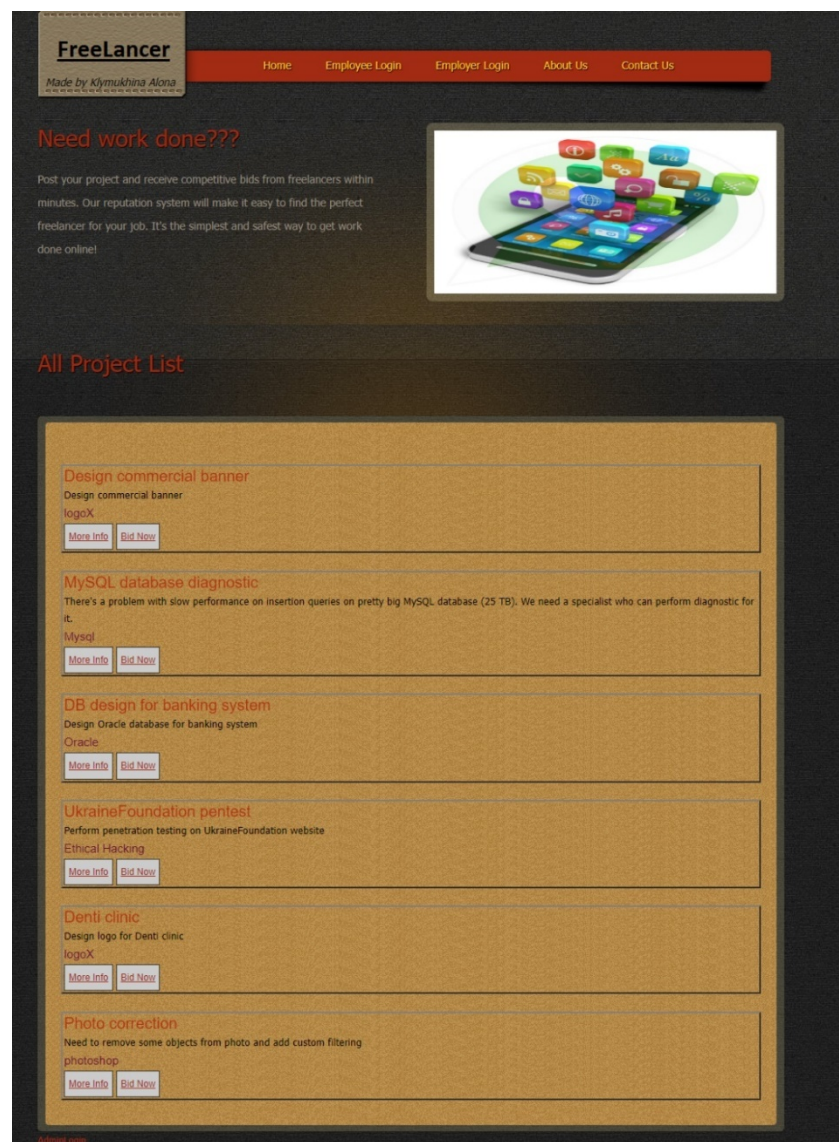


Рис. 3.5 Головний екран додатку для незареєстрованих користувачів

Навігаційне меню містить усі можливі варіанти роботи з додатком для гостю. У змісті головної сторінки для гостьового користувача наведено усі відкриті проекти.

Якщо користувач не гість, можливо відгукнутись на проєкт при натисненні на «Від now». В іншому випадку пропонується пройти авторизацію як робітник. Форма авторизації (рис. 3.6) використовується однакова для усіх типів користувачів.

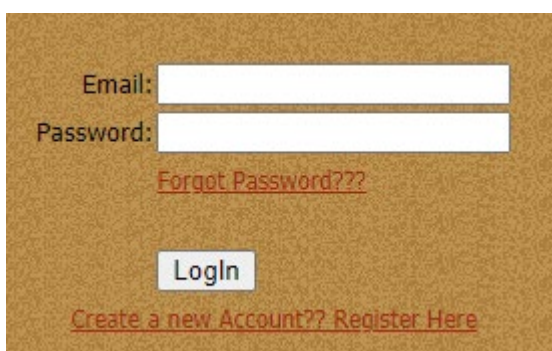
The image shows a login form on a brown textured background. It includes two input fields for 'Email:' and 'Password:'. Below the password field is a red link 'Forgot Password???'. A 'Login' button is centered below the links. At the bottom is a red link 'Create a new Account?? Register Here'.

Рис. 3.6 Форма авторизації

У випадку відсутності користувача в базі він може зареєструватися використовуючи форму реєстрації (рис. 3.7).

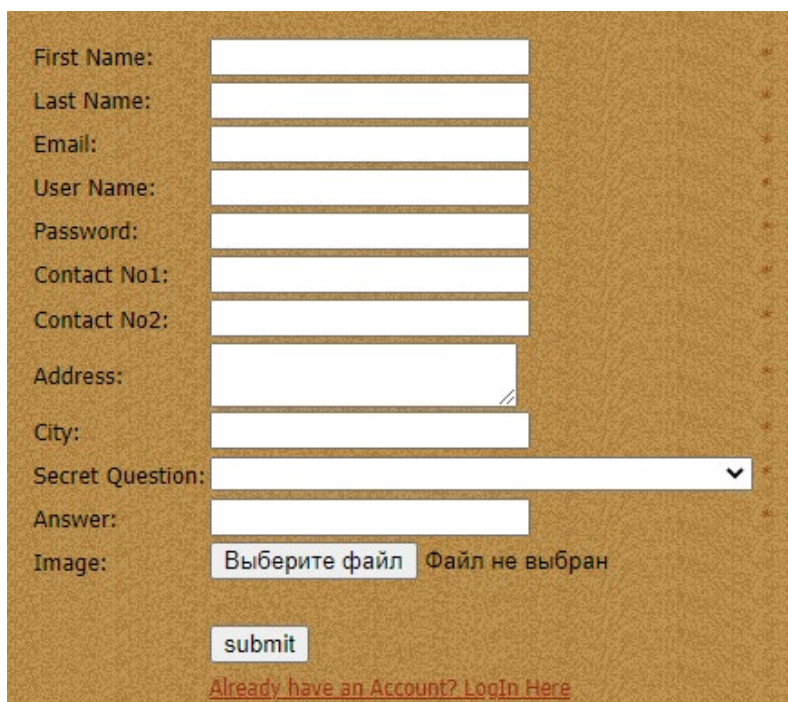
The image shows a registration form on a brown textured background. It contains multiple input fields for 'First Name:', 'Last Name:', 'Email:', 'User Name:', 'Password:', 'Contact No1:', 'Contact No2:', 'Address:', 'City:', 'Secret Question:' (with a dropdown arrow), 'Answer:', and 'Image:'. The 'Image:' field has a button 'Выберите файл' and the text 'Файл не выбран'. A 'submit' button is at the bottom. Below the button is a red link 'Already have an Account? LogIn Here'.

Рис. 3.7 Форма реєстрації

При натисненні на «More info» біля проєкту здійснюється перехід на інформаційну сторінку проєкту (рис. 3.8). Вона доступна без авторизації.

Project Name	Murchik detector
Details	Looking for an AI specialist who can create a neural network for detecting my cat "Murchik" among other street cats based on an online video camera output. Murchik likes walking on the streets and I wanna automate home door opening for him.
Budget	\$500 to \$1,000
Skill	Artificial Intelligence
Days to Complete	30
Posted by	Klymukhina
Date of Posted	2023-01-12
Required file	view file

[Bid Now](#)

Рис. 3.8 Інформаційна сторінка проєкту

Для гостьового користувача була створена форма зв'язку з адміністратором для донесення зауважень щодо роботи додатку.

Contact Us

Your Email:

Subject:

message:

Рис. 3.9 Форма зв'язку з адміністратором

А також сторінку «About us» де стисло описано основні напрямки проєктів які присутні в пропозиціях.



Рис. 3.10. Сторінка «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 підкатегорії 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 адміністратора

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 проєкту обираються на інформаційній сторінці проєкту з поміж робітників, що запропонували свої послуги.

У арсеналі роботодавця знаходиться основний функціонал наповнення додатку – можливість створювати проєкти. Функціонал створення проєктів показано на рис. 3.15.

Freelancer
Made by Kymahana Alana

Home My profile Post Project Logout

Need work done???

Post your project and receive competitive bids from freelancers within minutes. Our reputation system will make it easy to find the perfect freelancer for your job. It's the simplest and safest way to get work done online!

Post a Project

What type of work do you require?
Computer science

What skills are required?
Artificial Intelligence

What is your project about?
Murchik detector

Describe your project.
Looking for an AI specialist who can create a neural network for detecting my cat "Murchik" among other street cats based on an online video camera output. Murchik likes walking on the streets and I wanna automate home door opening for him.

What budget do you have in mind?
\$500 to \$1,000

How Many Days to Complete Project?
30

Attach Required File:
Вибрати файл

Post a Project

Рис. 3.15 Сторінка створення проєкту

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 проєктів на які виділено бюджет, а також наявності форми заповнення платіжних даних.

Основним функціоналом є відгуки на проєкти замовників. На інформаційній сторінці проєкту (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).

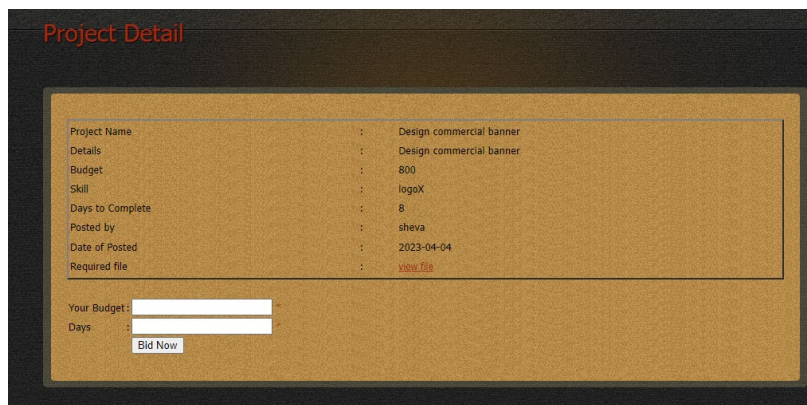


Рис. 3.17 Інформаційна сторінка проєкту для робітника

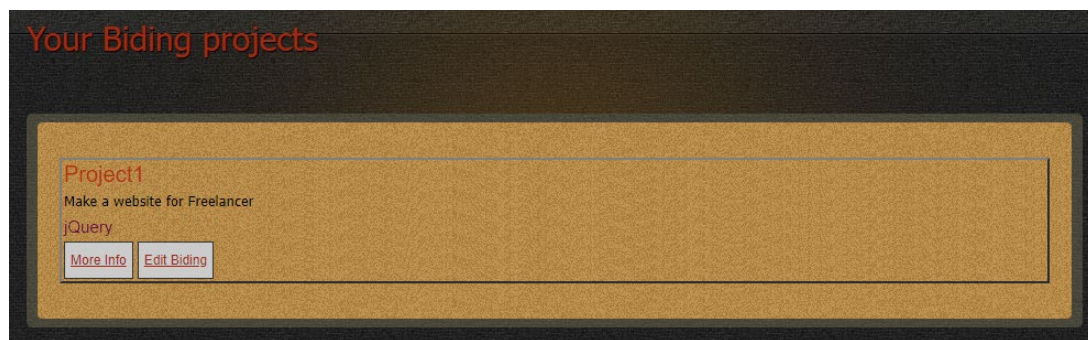


Рис. 3.18. Сторінка заявок робітника

Проте не кожен робітник, може подавати заявку на проведення робіт. Необхідна наявність співпадіння навичок вказаних в проєкті 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.

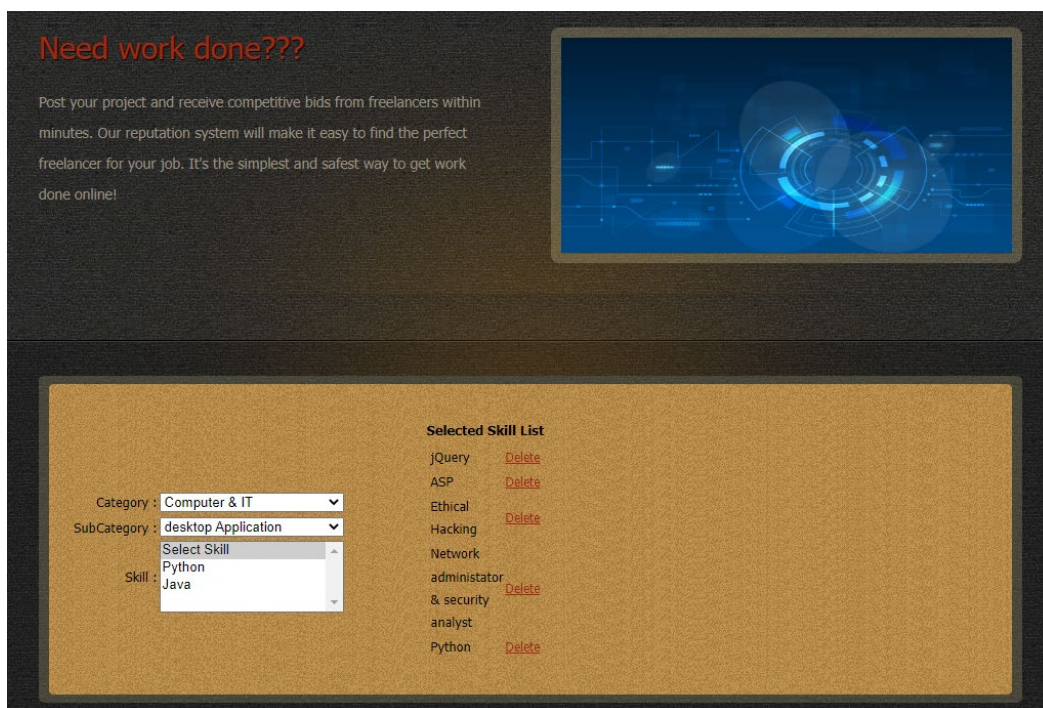


Рис. 3.19. Сторінка редагування навичок робітника

Задля полегшення пошуку пропозицій створено сторінку пошуку проєктів за вказаними в профілі навичками (рис. 3.20). На ній перераховані абсолютно усі проєкти, що підходять робітнику.

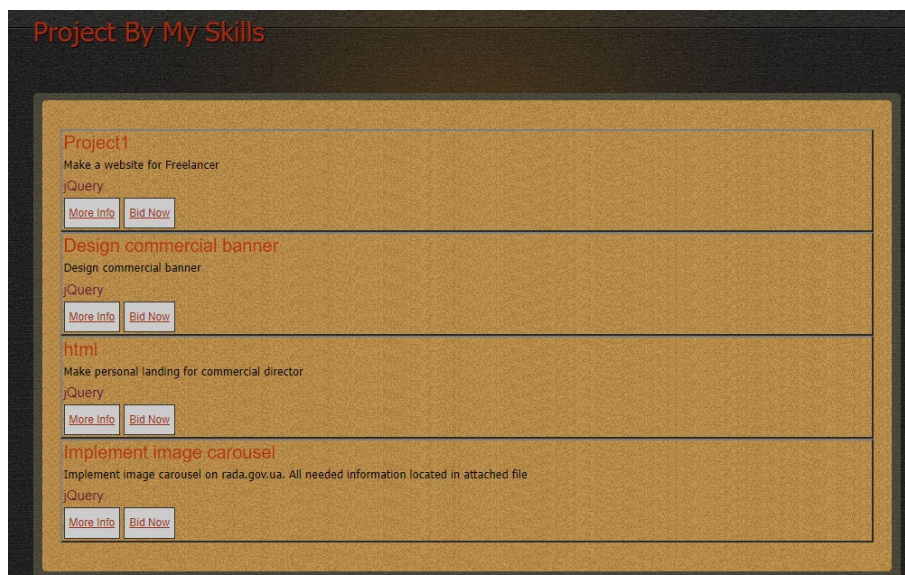


Рис. 3.20 Сторінка відібраних проектів

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

На сьогодні фріланс перебуває у розквіті, особливо на ринку надання ІТ - послуг. Існує чимало спеціальних веб-сайтів, які допомагають фрілансерам отримати чергову роботу. Проаналізувавши деякі з них, а також визначивши їх функціональні можливості та недоліки, було визначена важливість розробки роботи в цьому напрямку, і, як результат, вирішено створити власну веб-орієнтовану інформаційну систему пошуку виконавців ІТ-проектів.

У рамках випускної кваліфікаційної роботи було виконано загальний огляд і аналіз сучасного стану розглянутої проблеми. Зокрема приділено увагу особливостям процесу розробки веборієнтованих ІС, загальному огляду та вимог до їх функціональним можливостям.

Проведено дослідження та аналіз предметної області, формування вимог до створюваної системи; були обрані методи та інструменти реалізації веб-орієнтованої інформаційної системи; змодельовано інформаційну систему, спроектовано її інтерфейс та структуру бази даних.

Основними функціями розробленої системи є реєстрація та авторизація користувачів, розміщення на сайті завдань замовників з можливістю прикріплення файлів, перегляд завдань зареєстрованими користувачами, створення пошуку для формування списку робіт, створення заявок на отримання замовлення, можливість спілкування між замовником та виконавцем, формування оповіщень, формування рейтингу та історії проєкту, блокування користувачів.

Для виконавця система дозволяє виконувати замовлення, створювати портфоліо та спілкуватися з замовниками.

Для клієнта – створити замовлення в інформаційній системі, обрати виконавця та спілкування з виконавцем за допомогою чату. Головною особливістю створеного сервісу є можливість створення аккаунту, як для особистого використання, так й для команди розробників чи компанії. Дана

можливість значно розширює можливості роботи та підвищує можливості користувача. Можна отримати повний пакет послуг, витративши на пошуки менше зусиль.

Структура проєкту добре організована, користувач має змогу легко користуватися додатком, а для розробника з мінімальними затратами часу можливо модернізувати ІС під потреби (розробити мобільні версії додатку; додати систему трекінгу проєкту для робітників та роботодавців).

Розроблена пошукова ІС для фріланс біржі може бути використана як відкрита фріланс-платформа, яка повинна полегшити знаходження замовлення та комунікацію між клієнтами та виконавцями. Інтерфейс веб-системи досить зручний, зрозумілий і надійний у використанні, а забезпечення захисту даних дозволяє перехід на відповідні сторінки тільки авторизованим користувачам.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 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/>
2. Freelancehunt. URL: freelancehunt.com. (Date of access: 24.12.2022).
3. David Geary, Cay S. Horstmann. Core JavaServer Faces, 3rd Edition. 2011. 656 p.
4. UML моделирование. URL:<https://habr.com>. (Date of access: 24.12.2022).
5. Посібник з HTML та CSS3. URL: <https://metanit.com/web/html5/>. (Date of access: 24.12.2022).
6. Основи CSS3. URL:<https://metanit.com/web/html5/>. (Date of access: 24.12.2022).
7. Язык программирования Python. URL:<https://metanit.com/python/>. (Date of access: 24.12.2022).
8. Доценко С. І. Людино-машинний інтерфейс: навч.посібник. Харків: УкрДУЗТ, 2022. 135 с.
9. Кевин Мельтцер Разработка CGI-приложений на Perl. Вильямс. 2011. 400с.
10. Philip A. Koryaka. CGI - Simple Common Gateway Interface Class URL: <http://www.xserver.ru/computer/langprogr/cgi/>. (Date of access: 24.12.2022).
11. Введение в поисковую оптимизацию. URL:<https://developers.google.com/search/docs/fundamentals/seo-starter-guide?hl>. (Date of access: 24.12.2022).
12. Кешування та продуктивність веб-додатків. URL: <https://habr.com/company/ruvds/blog/350310/>. (Date of access: 24.12.2022).
13. Адміністрування. URL:<https://workspace.google.com/intl/uk/products/admin/>. (Date of access: 24.12.2022).
14. UpWork. URL: <https://www.upwork.com/>. (Date of access: 24.12.2022).
15. 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).

16. Пол МакФедрис. Web Design Playground (2019). URL:https://vk.com/wall-54530371_296845. (Date of access: 24.12.2022).

17. Учебник HTML с CSS. URL:https://www.schoolsw3.com/html/html_css.php. (Date of access: 24.12.2022).

18. Eric Elliott. Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries – O'Reilly Media, 2014. 254 pages

19. Adriaan de Jonge, Phil Dutson. jQuery, jQuery UI, and jQuery Mobile: Recipes and Examples – Addison-Wesley Professional, 2012. 400 pages.

20. jQuery Overview. URL: <https://www.tutorialspoint.com/jquery/jquery-overview.htm>. (Date of access: 24.12.2022).

21. Why Quill - Quill Rich Text Editor. URL:<https://quilljs.com/guides/why-quill>. (Date of access: 24.12.2022).

22. MySQL Tutorial. URL: <https://www.w3schools.com/MySQL/default.asp>. (Date of access: 24.12.2022).

23. Chris Dawson, Ben Straub. Building Tools with GitHub: Customize Your Workflow – O'Reilly Media, 2016. 302 pages.

24. Deepu K Sasidharan, Sendil Kumar N. Full Stack Development with JHipster – Packt Publishing, 2018. 380 pages.

25. Котеров Д.В., Симдянов И.В. PHP 8. СПб.: БХВ, 2021. – 850с.

26. Browser Support | jQuery. URL:<https://jquery.com/browser-support/>.(Date of access: 24.12.2022).

27. Free-lance.ua. URL: free-lance.ua. (Date of access: 24.12.2022).

28. Weblancer.net. URL: weblancer.net. (Date of access: 24.12.2022).

29. Kabanchik.ua. URL: kabanchik.ua. (Date of access: 24.12.2022).

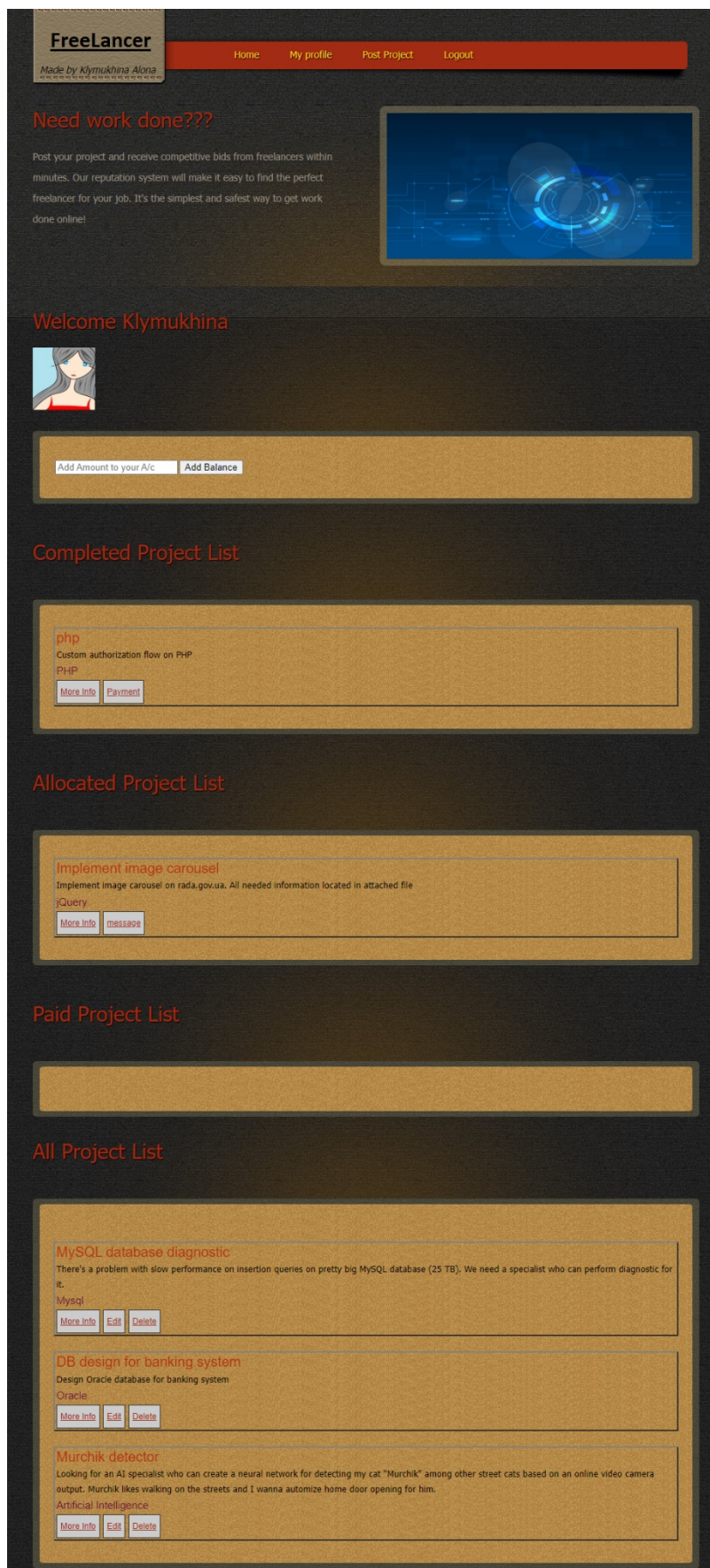
30. Jason Price, Oracle Database 11g SQL. McGraw-Hill Osborne Media, 2007. 656с.

31. Дронов В. А. HTML5, CSS3 и Web 2.0. Разработка современных Web-сайтов. СПб. : БХВ, 2011. 416 с.

32. Хеник Б. HTML и CSS: путь к совершенству. СПб. : Питер, 2011. 336 с.
33. Джон Дакетт - HTML и CSS. Разработка и дизайн веб-сайтов [пер. С англ. М.А. Райтмана]. М. : Эскимо, 2013. 480 с.
34. Matt Frisbie. Professional JavaScript for Web Developers. 4th Edition. Wrox. 2019. 1200p.

ДОДАТКИ

Додаток А. Головна сторінка роботодавця



Додаток Б. Код employee_home.php

```

<?php
session_start();
ob_start();
include("header.php");
include("Config.php");
$eid = $_SESSION['employer_id'];
$em = $_SESSION['em'];
$sql = "select * from employer_regitbl where em='$em'";
$res = mysql_query($sql);
$row = mysql_fetch_array($res);
if (isset($_POST['submit'])) {
    mysql_query("update employer_regitbl set balance='" . $_POST['addbal'] .
"'where employer_id=" . $eid) or die("Error");
}
?>
<style type="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="tooplate_middle">
    <div id="mid_title">Welcome <?php echo $_SESSION['unm']; ?></div>
    
    <div class="cleaner"></div>
</div> <!-- end of middle -->
<div id="tooplate_main_top"></div>
<div id="tooplate_main">
    <div id="gallery">
        <form method="POST">
            <table>
                <tr>
                    <td><input type="text" name="addbal"
placeholder="Add Amount to your A/c"></td>
                    <td><input type="submit" name="submit" value="Add
Balance"></td>
                </tr>
            </table>
        </form>
    <div class="cleaner"></div>
</div>
<div class="cleaner"></div>
</div> <!-- end of main -->
<div id="tooplate_main_bot"></div>
<div id="tooplate_middle">
    <div id="mid_title"></div>

```

```

        <div id="mid_title">Completed Project List</div>
</div> <!-- end of middle -->
<div id="tooplate_main_top"></div>
<div id="tooplate_main">
    <div id="gallery">
        <?php
            $result1 = mysql_query("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("select * from prjtbl where p_id="
                    . $r1[1], $con) or die("Error in Select Query2");
                    if (mysql_num_rows($result2) > 0) {
                        while ($r2 = mysql_fetch_array($result2)) {
                            ?>
                                <table width="100%" border="2">
                                    <tr>
                                        <td align="left" style="font-
family:Arial;font-size:20px;color:#B6300E">
                                            <?php echo $r2[2]; ?></td>
                                        <tr>
                                            <td align="left"><?php echo
$ r2[5]; ?></td>
                                        </tr>
                                        <?php
$ result10 = mysql_query("select * from
skilltbl where skill_id=" . $r2[4]);
                                        if (mysql_num_rows($result10) > 0) {
                                            while ($r10 =
mysql_fetch_array($result10)) {
                                                ?>
                                                    <tr>
                                                        <td align="left"
style="font-family:Arial;font-size:15px;color:#6E103C"><?php echo $r10[1]; ?></td>
                                                    </tr>
                                                    <?php }
                                                    } ?>
                                                    <tr>
                                                        <td><a
href="employercompleteprjview.php?pid=<?php echo $r2[0]; ?>& cid=<?php echo
$ r1[0]; ?>" class='link'>More Info</a>
                                                        <a
href="employerpayment.php?pid=<?php echo $r2[0]; ?>& cid=<?php echo $r1[0]; ?>"
class='link'>Payment</a>
                                                    </td>
                                                    </tr>
                                                </table>
                                            <?php
                                                }
                                            }
                                        }
                                    ?>
                                <div class="cleaner"></div>
                            </div>
                        <div class="cleaner"></div>
                    </div> <!-- end of main -->
                <div id="tooplate_main_bot"></div>
                <div id="tooplate_middle">
                    <div id="mid_title"></div>
                    <div id="mid_title">Allocated Project List</div>
                </div> <!-- end of middle -->

```

```

<div id="tooplate_main_top"></div>
<div id="tooplate_main">
    <div id="gallery">
        <?php
            $result11 = mysql_query("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("select * from prjtbl where p_id="
. $r11[1], $con) or die("Error in Select Query2");
                    if (mysql_num_rows($result12) > 0) {
                        while ($r12 = mysql_fetch_array($result12)) {
                            ?>
                                <table width="100%" border="2">
                                    <tr>
                                        <td align="left" style="font-
family:Arial;font-size:20px;color:#B6300E">
                                            <?php echo $r12[2];
                                        </td>
                                    <tr>
                                        <td align="left"><?php echo
$r12[5]; ?></td>
                                    </tr>
                                <?php
                                    $result1 = mysql_query("select * from
skilltbl where skill_id=" . $r12[4]);
                                    if (mysql_num_rows($result1) > 0) {
                                        while ($r1 =
mysql_fetch_array($result1)) {
                                            ?>
                                                <tr>
                                                    <td align="left"
style="font-family:Arial;font-size:15px;color:#6E103C"><?php echo $r1[1]; ?></td>
                                                </tr>
                                            <?php }
                                            } ?>
                                        <tr>
                                            <td><a
href="prj_allocate.php?pid=<?php echo $r12[0]; ?>" class='link'>More Info</a>
                                            <a
href="message.php?pid=<?php echo $r12[0]; ?>" class='link'>message</a>
                                            </td>
                                        </tr>
                                    </table>
                                <?php
                                    }
                                }
                            }
                        }
                    }
                }
            }
        </div>
        <div class="cleaner"></div>
        <div class="cleaner"></div>
    </div> <!-- end of main -->
<div id="tooplate_main_bot"></div>
<div id="tooplate_middle">
    <div id="mid_title"></div>
    <div id="mid_title">Paid Project List</div>
</div> <!-- end of middle -->
<div id="tooplate_main_top"></div>
<div id="tooplate_main">
    <div id="gallery">

```

```

        <?php
        $result10 = mysql_query("select employerpay_id,p_id from employerpaytbl
where employer_id=" . $eid, $con) or die("Error in Select Query1");
        if (mysql_num_rows($result10) > 0) {
            while ($r10 = mysql_fetch_array($result10)) {
                $result20 = mysql_query("select * from prjtbl where p_id="
. $r10[1], $con) or die("Error in Select Query2");
                if (mysql_num_rows($result20) > 0) {
                    while ($r20 = mysql_fetch_array($result20)) {
                        ?>


|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <td \$r1[1];="" \$r20[5];="" \$result='mysql_query("select' \$result1='mysql_query("select' &gt;="" &gt;&lt;="" &gt;&lt;?php="" &gt;all="" &gt;more="" &lt;="" &lt;!--="" &lt;?php="" &lt;div="" &lt;tr&gt;="" (select="" *="" --&gt;="" <="" <td="" <td&gt;&lt;a="" <tr&gt;="" ?&gt;="" ?&gt;&lt;="" a&gt;="" align="left" and="" class="cleaner" completeprj)="" div&gt;="" echo="" end="" from="" href="employercompleteprjview.php?pid=&lt;?php echo \$r20[0]; ?&gt;&amp;cid=&lt;?php echo \$r10[0]; ?&gt;" id="gallery" in="" info&lt;="" left"="" list&lt;="" main="" middle="" not="" of="" p_id="" pre="" prj_allocated)="" prjtbl="" project="" skill_id=" . \$r20[4]);         if (mysql_num_rows(\$result1) &gt; 0) {             while (\$r1 = mysql_fetch_array(\$result1)) {                 ?&gt; &lt;tr&gt; &lt;td align=" skilltbl="" style="font-family:Arial;font-size:15px;color:#6E103C" table&gt;="" td&gt;="" tr&gt;="" where="" }=""> </td> |  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|


```

```

IN (select p_id from employerpaytbl) AND employer_id=" . $eid, $con) or die("Error in
Select Query");
    if (mysql_num_rows($result) > 0) {
        while ($r = mysql_fetch_array($result)) {
            ?> <br />
                <table width="100%" border="2">
                    <tr>
                        <td align="left" style="font-
family:Arial;font-size:20px;color:#B6300E"><?php echo $r[2]; ?></td>
                    </tr>
                    <tr>
                        <td align="left"><?php echo $r[5]; ?></td>
                    </tr>
                    <?php
$result1 = mysql_query("select * from skilltbl where
skill_id=" . $r[4]);
                    if (mysql_num_rows($result1) > 0) {
                        while ($r1 = mysql_fetch_array($result1)) {
                            ?>
                                <tr>
                                    <td align="left" style="font-
family:Arial;font-size:15px;color:#6E103C"><?php echo $r1[1]; ?></td>
                                </tr>
                                <?php }
                                } ?>
                                <tr>
                                    <td><a href="employerviewbid.php?pid=<?php
echo $r[0]; ?>" class='link'>More Info</a>
                                    <a href="employereditprj.php?pid=<?php
echo $r[0]; ?>" class='link'>Edit</a>
                                    <a href="employerdelprj.php?pid=<?php
echo $r[0]; ?>" class='link'>Delete</a>
                                </td>
                                </tr>
                            </table>
                        <?php }
                        }
                        ?>
                        <?php
include("footer.php");
                        ?>

```