

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА**
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем

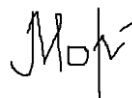
122 «Комп'ютерні науки»
(шифр і назва спеціальності)

«Прикладне програмування»
(назва освітньої програми)

Кваліфікаційна робота бакалавра

на тему: «Веб-застосунок для пошуку робочих місць з ІТ спеціальностей»

Виконав _____
(Підпис)



Морозова Лідія Костянтинівна
(прізвище, ім'я, по батькові)

Керівник Гарко Ірина Ігорівна
(прізвище, ім'я, по батькові)



(Резолюція «До захисту»)

Попередній захист:

До захисту
(Висновок: “До захисту в екзаменаційній комісії”)

Завідувач кафедри _____ Плескач В.Л.

(Підпис) (Прізвище, ініціали)

Засвідчую, що у цій дипломній роботі немає запозичень із праць інших авторів без відповідних посилань.



Унікальність тексту роботи - 88 %

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

№з/п	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	09.10.2021	виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	19.10.2021	виконано
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	21.10.2021	виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	25.10.2022	виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	01.11.2022	виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	21.12.2022	виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	31.01.2022	виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	30.03.2022	виконано
9.	Подання роботи у першому варіанті	28.04.2022	виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	03.05.2022	виконано
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	23.05.2022	виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедрі	27.05.2022	виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботу)	10.06.2022	виконано
14.	Захист кваліфікаційної роботи бакалавра	22.06.2022 23.06.2022 24.06.2022	виконано

Здобувач вищої освіти _____

(підпис)

Керівник _____

ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Складові частини дипломної роботи	Обсяг, арк.
Титульний аркуш	1
Календарний план дипломної роботи	1
Відомість дипломної роботи	1
Анотація	1
Анотація (іноземною мовою-англійською)	1
Зміст	1
Перелік скорочень, умовних позначень, термінів	0
Вступ	3
Розділ 1	13
Розділ 2	8
Розділ 3	5
Висновки	1
Список використаних джерел	2
Додатки	32

				ДП ХХХХ 00.000.00		
	ПІБ	Підп.	Дата	Відомість дипломної роботи	Лист	Листів
Розро бн.	Морозова Л.К.					
Керівн .	Гарко І.І.					
Н/кон тр.	Базилюк А.М.					
Зав.ка ф.	Плескач В.Л.					

АНОТАЦІЯ

Кваліфікаційна робота бакалавра містить: 77 сторінок, 26 рисунків, 17 використаних джерел. Метою роботи є підвищення ефективності пошуку робочих місць з ІТ спеціальностей. Об'єктом дослідження є процеси ефективного пошуку робочих місць з ІТ спеціальностей на основі веб-застосунку як інструмента отримання інформації для клієнта. Предмет дослідження – програмні засади та принципи побудови веб-застосунку за заданою тематикою. Методи дослідження - метод аналізу (пошук теоретичних даних, критичний їх розгляд, виявлення необхідних при реалізації проекту підходів), описовий метод (проекування алгоритмів розроблення, ведення документації та формування звіту до виконаної роботи), метод синтезу (підбір та комбінування сучасних найефективніших програмних і теоретичних рішень з метою отримання системи компонентів, які підходить до виконання поставленого завдання). У результаті роботи було розроблено веб-застосунок для пошуку робочих місць, в даному конкретному випадку – з ІТ спеціальностей. Програмна реалізація виконана мовою програмування Java Script.

Ключові слова: веб-застосунок, пошук робочих місць, ІТ спеціальність, вакансія, користувач.

ABSTRACT

The bachelor's thesis contains: 77 pages, 26 drawings, 17 sources used. The aim of the work is to increase the efficiency of job search in IT specialties. The object of research is the processes of effective job search in IT specialties based on the web application as a tool for obtaining information for the client. The subject of research - software principles and principles of building a web application on a given topic. Research methods - method of analysis (search for theoretical data, critical consideration, identification of necessary approaches in the project), descriptive method (design of algorithms for development, documentation and reporting on the work performed), synthesis method (selection and combination of modern most effective software and theoretical decisions in order to obtain a system of components that are suitable for the task). As a result, the web application was developed to search for jobs, in this case - from IT specialties. The software implementation is made in the Java Script programming language.

Keywords: web application, job search, IT specialty, vacancy, user.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ОСНОВНІ ТЕНДЕНЦІЇ РОЗРОБЛЕННЯ СУЧАСНИХ ВЕБ-ЗАСТОСУНКІВ	10
Аналіз стану ІТ сфери за останні роки	10
1.2 Огляд існуючих веб-застосунків для ефективного пошуку робочих місць	12
1.3 Переваги та недоліки існуючих ІТ рішень поставленої проблеми	17
1.4 Обґрунтування мети вирішення поставленої задачі	23
РОЗДІЛ 2. АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ І ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-ЗАСТОСУНКУ.	24
2.1. Веб-застосунок та його основний принцип роботи	24
2.2. Інструменти розробки проєктованого веб-застосунку	26
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ ПОШУКУ РОБОЧИХ МІСЦЬ З ІТ СПЕЦІАЛЬНОСТЕЙ.	32
3.1 Структурна карта веб-застосунку	32
3.2 Функціональне поле розроблюваного застосунку	34
3.3 Збереження даних	35
3.4 Інструкція користувача	37
ВИСНОВКИ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	45
ДОДАТКИ	47

ВСТУП

Актуальність дослідження. У зв'язку з введенням воєнного стану в Україні ринок праці зазнав значних змін. Понад половина українців, які були працевлаштовані до початку повномасштабної війни, зараз не працюють. Менше чверті працюють у звичайному режимі, частина – віддалено чи частково, і лише два відсотки знайшли нову роботу [1]. Відтак, внаслідок російського вторгнення, потреба в працевлаштуванні громадян України зростає.

Найменш вразливими виявилися працівники ІТ-секторів, бо, у своїй більшості, отримують замовлення з інших країн та західних компаній, та, як правило, мають можливість працювати віддалено, незалежно від місця перебування. Тим не менш, втрата замовлень в українських компаніях націлених на локальний ринок, відсутність бажання співпрацювати з російськими та білоруськими замовниками, хиткість стабільності в економіці, пришвидшений ріст конкуренції – всі ці та інші чинники впливають на потребу ІТ спеціалістів пошуку нових місць роботи.

За статистикою, кількість зареєстрованих безробітних у комп'ютерній сфері у період з січня-квітня 2022 року зменшилась майже в два рази [2]. Як наслідок процесу діджиталізації, основним джерелом пошуку роботи стали веб засоби. Аналітика свідчить, що відвідуваність застосунків даного формату збільшується з кожним днем. Кількість відвідувачів веб-застосунків з пошуку робочих місць станом на зараз в чотири рази більше, ніж на початку року [3].

Веб-застосунки з пошуку робочих місць, на даний момент, є найефективнішим та найперспективнішим інструментом відповіді на безробіття у всіх сферах, включаючи ІТ.

Мета дослідження. Підвищення ефективності пошуку робочих місць з ІТ спеціальностей.

Завдання дослідження. Обґрунтування та досягнення мети кваліфікаційної роботи бакалавра передбачає визначення та вирішення наступних завдань:

- дослідити основні тенденції в розробленні сучасних веб-застосунків;
- здійснити аналіз архітектурних рішень та обрати програмні засоби для реалізації веб-застосунку;
- здійснити програмну реалізацію веб-застосунку підбору для пошуку робочих місць з ІТ спеціальностей.

Об’єкт дослідження. Процеси ефективного пошуку робочих місць з ІТ спеціальностей на основі веб-застосунку як інструмента отримання інформації для клієнта.

Предмет дослідження. Програмні засади та принципи побудови веб-застосунку за заданою тематикою.

Методи дослідження. Для розкриття теми та поставлених задач кваліфікаційної роботи бакалавра, а також для формування звіту про розроблення веб-застосунку було використано такі методи:

- метод аналізу при дослідженні наявних програмних рішень для пошуку робочих місць з ІТ спеціальностей;
- описовий метод при формуванні алгоритму розроблення та кроків моделювання проекту, при описі інструментів і технологій розроблення;
- метод наукового моделювання при формуванні таблиць, моделей та інших компонентів, для відображення необхідних структурних елементів веб-застосунку та іншого допоміжного інструментарію;
- метод наукового порівняння при роботі з існуючими альтернативними програмними рішеннями для створення переліку їхніх переваг та недоліків;
- метод синтезу при вивченні сучасних програмно-технологічних рішень, бібліотек, стеків технологій при виборі оптимального, доцільного та раціонального набору для розробки веб-застосунку.

Практичне значення одержаних результатів. Підвищення ефективності пошуку робочих місць з ІТ спеціальностей за допомогою сучасних технологій, а саме у вигляді веб-застосунку, що позитивно впливе на рівень зайнятості українців.

Структура роботи.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, поділених на підрозділи, висновків, списку використаних джерел та додатків. Загальний обсяг роботи складає 77 сторінок. Список використаних джерел охоплює 17 найменувань.

РОЗДІЛ 1. ОСНОВНІ ТЕНДЕНЦІЇ РОЗРОБЛЕННЯ СУЧАСНИХ ВЕБ-ЗАСТОСУНКІВ

1.1 Аналіз стану ІТ сфери за останні роки

Протягом останніх років в Україні з'являються все більше і більше молодих ІТ спеціалістів. Статистика свідчить: протягом 2018-2020 років кількість ФОПів в українському ІТ збільшилась на 20-22%. За 2020-2021 роки цей відсоток приросту ІТ спеціалістів трохи зменшився до 16%, що досі залишається достатньо хорошим показником. Більш того, за рік кількість спеціалістів, офіційно зареєстрованих як робітників в сфері комп'ютерного програмування, становить 143549 людей [4]. Кількість спеціалістів зростає не тільки в сфері комп'ютерного програмування, а й в таких сферах як консультування з питань інформації, оброблення даних, розміщення інформації в веб-вузлах, діяльність у сфері комп'ютерних систем, видання іншого програмного забезпечення, видання комп'ютерних ігор, тощо (рис. 1.1).



Рисунок 1.1 – Перелік ІТ-КВЕДів зі зміною за 2021 рік [4]

Всі ці факти свідчать про стрімкий розвиток сфери серед українців, що матиме і вже має позитивні наслідки в економіці країни. Варто зазначити, що даний факт тягне за собою підвищення конкуренції та необхідність збільшення кількості робочих місць. З цього випливає потреба в зручних та ефективних інструментах пошуку роботи.

Експерти зазначають, що ІТ-галузь є однією з небагатьох, які несуттєво зазнали впливу війни. Можливість працювати віддалено, наявність закордонних замовників, великого поля роботи як для експертів так і новачків – всі ці чинники допомогли працівникам швидко адаптуватися до нових реалій. Тим не менше, кількість вакансій для ІТ-фахівців знизилася на 27% за даними Djinni [5]. Це майже втричі менше за кандидатів у пошуку роботи. Негативний вплив війни відчули на собі працівники компаній, що зазнали скорочення, кризи через роботу виключно з країнами, на які було накладено санкції, або ж клієнти яких відмовились від співробітництва через страх нестабільності такої угоди. Таким чином, потреба в робочих місцях серед ІТ спеціалістів збільшилась.

1.2 Огляд існуючих веб-застосунків для ефективного пошуку робочих місць

Кожен ІТ фахівець прагне знайти хороше місце роботи, при цьому маючи певний список власних критеріїв до роботодавців. Заробіток, кількість робочих годин, можливість швидкого кар'єрного росту, комфортні умови, великий дружній колектив – кожен працівник шукає місце роботи за власними пріоритетами та критеріями. Це зазвичай уповільнює пошук, але не робить його неможливим. Полегшити пошук робочого місця для ІТ спеціаліста намагалися багато розробників веб-застосунків, які ми розглянемо нижче.

Зараз найбільш актуальні програмні рішення для пошуку робочих місць наступні:

ROBOTA.UA – веб-застосунок, що надає можливість клієнту самостійно безкоштовно розглянути вакансії в Україні (рис. 1.2). ROBOTA.UA надає список з 51365 вакансій з заробітною платою від 3000 до 180000 гривень, постійно оновлюючи список актуальних вакансій.

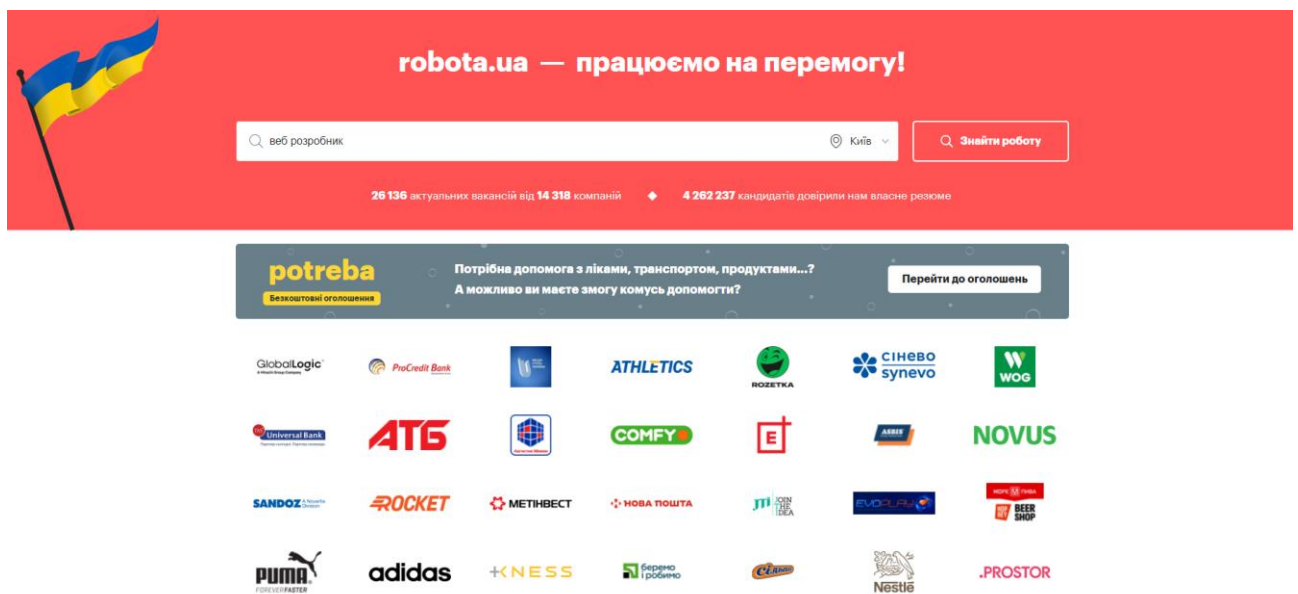


Рисунок 1.2 – Головна сторінка ресурсу ROBOTA.UA

Вакансії розділені за категоріями. Пошук можна здійснювати за найпопулярнішими професіями та розташуванням. Більш того, користувач може

ввести власні параметри пошуку, щоб отримати найвідповіднішу пропозицію. Користувач може зазначити вид зайнятості (кількість робочих годин), бажану заробітну плату та рівень складності. Більш того, можна вказати власну локацію, якщо є потреба знайти офіс недалеко від себе.

Розробники прийняли рішення полегшити пошук за допомогою накладання фільтрів. Бокова спливаюча панель допоможе найякісніше підібрати бажане місце роботи. Варто звернути увагу на можливість вибору декількох фільтрів одразу, а також, можливість скинути всі критерії до базових (рис. 1.3).

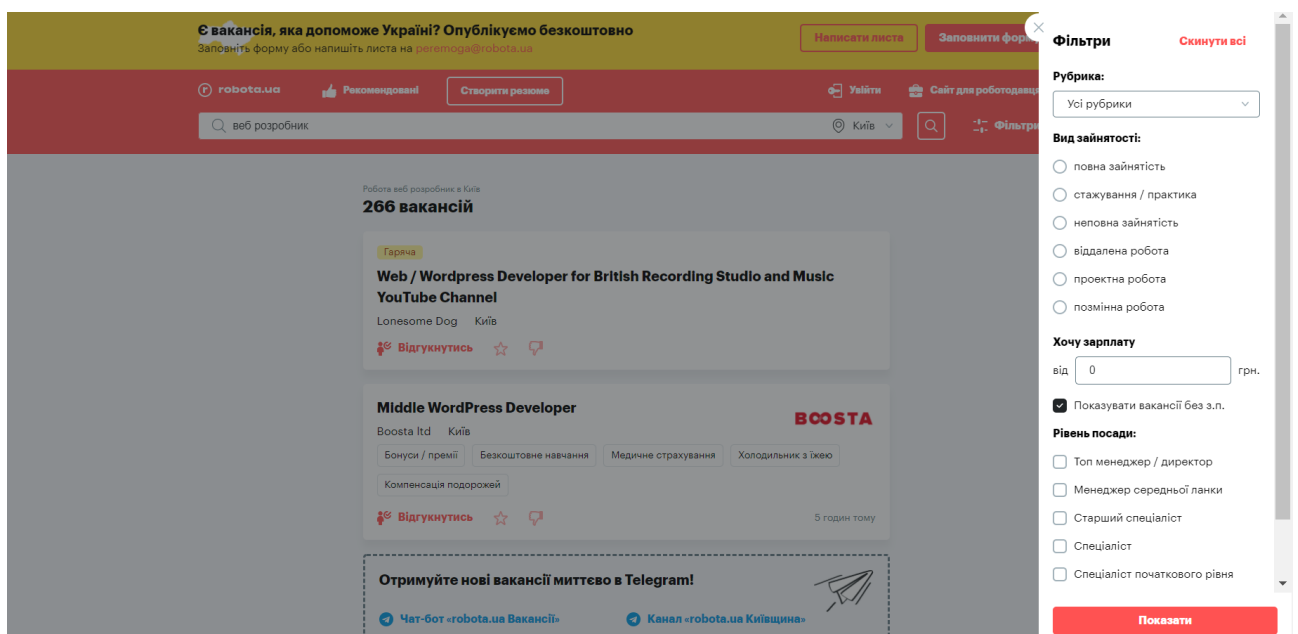


Рисунок 1.3 – Сторінка пошуку з фільтрами ресурсу ROBOTUA.UA

Застосунок WORK.UA має схожий функціонал. Ще однією перевагою цього застосунку є можливість фільтрувати пропозиції за датою їх створення, тобто, знайти найсвіжіші. Також, застосунок має функціонал сортування вакансій по даті, заробітній платі та можливості працювати віддалено. Остання функція є особливо корисною з урахуванням росту тенденції працювати з дому.

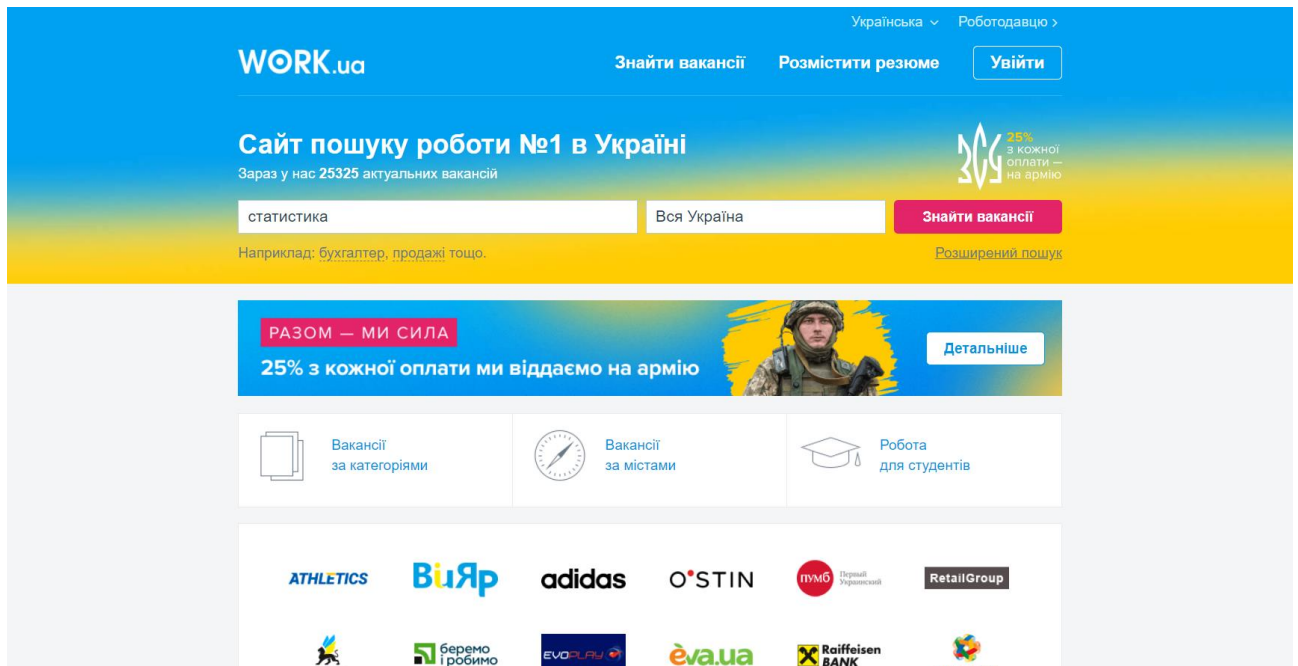


Рисунок 1.4 – Головна сторінка ресурсу WORK.UA

Список вакансій також розділений за категоріями [14]. Шукач роботи може обрати варіант не тільки за видом професії, а також за необхідністю мати досвід, резюме, або певний вік. Пропозиції також не обмежені за розташуванням, компаніями та видами позицій. Варто зазначити, що користувач може вподобати пропозицію, якою зацікавився (рис. 1.5).

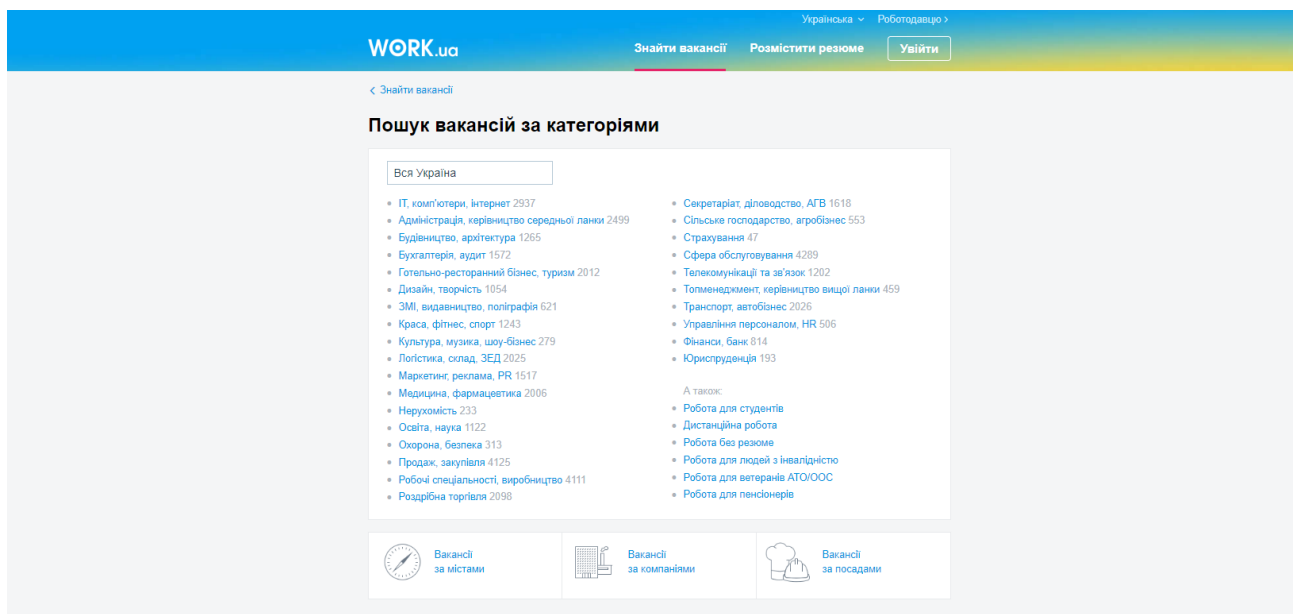


Рисунок 1.5 – Пошук вакансій за категоріями ресурсу WORK.UA

Описані вище програмні рішення поставленої задачі є доволі схожими. Тому розглянемо чим від них відрізняється веб-застосунок TRUD.ua.

TRUD.ua – веб-застосунок, що на даний момент має більше 210 тис. підписників, 54,5 тис. вакансій та 6,6 тис. поданих резюме. Застосунок надає статистику пошуку за містами, категоріями та професіями. Найпопулярнішими та найбільшими за кількістю є пропозиції працівників виробництва, адміністративного персоналу та робітників у сфері інформаційних технологій.

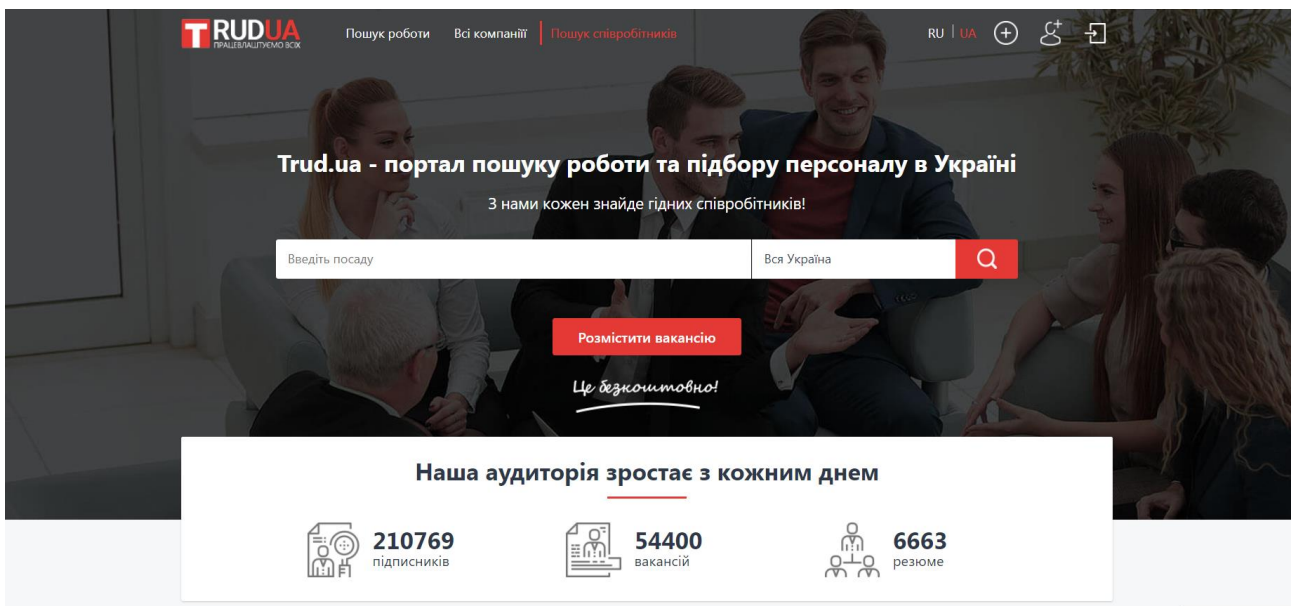


Рисунок – 1.6 Головна сторінка ресурсу TRUD.UA

Варто зазначити, пошук роботи тут може здійснюватися за компаніями – користувач шукатиме посаду, якою зацікавився, саме в конкретній українській компанії (рис 1.7). TRUD.ua – не менш корисний і для компаній, які шукають співробітників. Таким чином, даний портал збільшує коло зацікавлених користувачів від звичайного жителя країни до HR-фахівців великих організацій.

The image shows a screenshot of the TRUD.UA website's job search results for the company ERAM in Ukraine. The search bar at the top contains 'ERAM' and the location filter is set to 'Вся Україна'. The results are displayed in a list format. The first two results are highlighted with red text and links to external sites: 'Удалённая работа на дому, 800\$ - Стажировка оплачивается, 700\$' and 'Дистанционная работа на дому - Высокооплачиваемая вакансия'. Below these, there are four job listings for ERAM Systems Ukraine, including 'Бухгалтер', 'Personnel Officer', 'Senior JavaScript Software Engineer', and 'SAP CO (FI) Consultant'. The right sidebar contains filters for 'Период поиска' and 'Компания', and a 'Подписка в 1 клик!' button. The bottom of the page shows a 'Объявление закрыто Google' message.

Рисунок – 1.7 Сторінка пошуку вакансій за компанією ресурсу TRUD.UA

1.3 Переваги та недоліки існуючих ІТ рішень поставленої проблеми

Розглянуті веб-ресурси мають свої переваги та недоліки, які варто проаналізувати перед початком роботи над власним проектом.

Однією з найважливіших складових веб-застосунку є зовнішній вигляд, бо саме він створює перше враження користувача про створений ресурс. Тим не менш, перше враження може бути зіпсоване повільною швидкістю реакції сайту, або ж недостатнім функціоналом. Тому, проаналізувавши основні важливі елементи веб-застосунків, створимо порівняльну таблицю розглянутих ресурсів і опишемо їхні переваги та недоліки за наведеними параметрами. (табл. 1.1).

Таблиця 1.1 – Порівняльна таблиця застосунків з пошуку роботи.

Параметр	РАБОТА.ua	Work.ua	Trud.ua
Адаптивність	Адаптивний	Адаптивний	Неадаптивний дизайн. При зміні розширення вікна браузера фонові картинка змінює свій розмір та «їде» у правий верхній кут екрану.
Клієнтоорієнтованість	+ Орієнтовано на пошукача і роботодавця. + Можливість створення особистого кабінету.	Відсутня можливість переглянути вартість без реєстрації.	+ Орієнтовано на пошукача та роботодавця. + хороша навігація та інтуїтивність інтерфейсу. - Неможливо викласти резюме без реєстрації.

Продовження таблиці 1.1.

Параметр	РАБОТА.ua	Work.ua	Trud.ua
Швидкість дія	Від 1 до 475ms на запит.	Від 18 до 81ms на запит.	Від 1 до 194ms на запит.
Вартість функціоналу	<p>Інформаційно-рекламні послуги: 3000-16790.</p> <p>Опублікувати вакансію: безкоштовно.</p> <p>Викласти резюме: безкоштовно.</p>	<p>Набори послуг на місяць для розміщення вакансій та ефективного пошуку кандидатів: від 490 грн на місяць.</p> <p>Гаряча вакансія: від 3190 грн за шт.</p> <p>Доступ до бази кандидатів: від 1290 грн на тиждень.</p> <p>Викласти резюме або вакансію: безкоштовно (обмеження в 1 вакансію на до 10 контактів у базі кандидатів на місяць).</p>	<p>Опублікувати вакансію: безкоштовно.</p> <p>Знайти вакансію: безкоштовно</p>

Продовження таблиці 1.1

Параметр	РАБОТА.ua	Work.ua	Trud.ua
Інтуїтивність інтерфейсу	Інтуїтивний інтерфейс	Недостатньо інтуїтивний інтерфейс: Фільтри пошуку одночасно знаходяться в декількох місцях на сторінці, що ускладнює розуміння користувача як їх одночасно використати. Не на всіх сторінках присутня кнопка повернення до головної сторінки. Деякі функції блокуються відсутністю реєстрації, в той же час, якщо відкрити їх за допомогою адресної строки, реєстрація не потрібна.	Інтерфейс інтуїтивний. Полегшує пошук наявності досконалої навігації.

Продовження таблиці 1.1

Параметр	РАБОТА.ua	Work.ua	Trud.ua
Інформативність	<p>Пошукачу:</p> <ul style="list-style-type: none"> - знайти роботу - розмістити резюме - як скласти резюме - поради пошукачу <p>Роботодавцю:</p> <ul style="list-style-type: none"> - додати вакансію - знайти резюме - наші послуги - поради роботодавцю 	<p>Пошукачу:</p> <ul style="list-style-type: none"> - можливість викласти резюме безкоштовно - пошук з фільтрацією - статті та новини <p>Роботодавцю:</p> <ul style="list-style-type: none"> - знайти кандидатів - статті та новини 	<p>trud.ua:</p> <ul style="list-style-type: none"> - Блог - Права та обов'язки - Каталог вакансій - Universal Commerce Group - Trud.com <p>Претендентів:</p> <ul style="list-style-type: none"> - Знайти роботу - Всі компанії - Розмістити резюме

Продовження таблиці 1.1

Параметр	РАБОТА.ua	Work.ua	Trud.ua
Інформативність (продовження)	<p>robota.ua:</p> <ul style="list-style-type: none"> - про компанію - конфіденційність - написати в підтримку - умови користування - договір публічної оферти <p>budni:</p> <ul style="list-style-type: none"> - корисні статті - про проект - реклама 		<p>Роботодавцю:</p> <ul style="list-style-type: none"> - Пошук резюме - Додати вакансію - Послуги та сервіси
Навігація	Зручна та інтуїтивна навігація.	Навігація ускладнена недостатньою інтуїтивністю інтерфейсу.	Інтуїтивна навігація. Полегшена навігація за допомогою бокового меню, верхнього меню та нижньої панелі навігації.

Продовження таблиці 1.1

Параметр	РАБОТА.ua	Work.ua	Trud.ua
Кольорова гама	Переважає червоний колір, що вважається дратівливим для людського ока.	Основний колір – голубий(синій). Колір, що заспокоює і асоціюється зі стабільністю, на думку психологів ідеально підходять для людей, що знаходяться в пошуку роботи[17].	Переважає сірий колір з елементами червоного, для створення акценту на важливих елементах.

1.4 Обґрунтування мети вирішення поставленої задачі

Проведений аналіз наявних програмних рішень дає змогу звернути увагу на особливості та передбачити можливі недоліки, які варто враховувати при побудові архітектури власного застосунку. Тому, поставивши перед собою задачу щодо розроблення веб-застосунку для пошуку робочих місць з ІТ спеціальностей, визначено, що його рішення складається з таких підпунктів:

- інтуїтивний інтерфейс;
- швидке оброблення запитів;
- не дратівливий, якісно розпланований дизайн;
- можливість переглядати відкриті вакансії без реєстрації;
- наявність можливості додавати вакансії в розділ «вподобане» за особистим інтересом користувача;
- наявність можливості видаляти вакансії зі списку «вподобаних»;
- додавання вакансій;
- видалення вакансій;
- сортування вакансій за різними параметрами;
- інший функціонал для зручності роботи та створення інтуїтивного інтерфейсу.

Реалізація програмного рішення з заявленими функціональними можливостями є основною метою кваліфікаційної роботи бакалавра. Домінуючим фактором, що впливає на розроблення та якість застосунку, варто вважати створення безкоштовного інструментарію, що сприятиме підвищенню ефективності пошуку робочих місць з ІТ спеціальностей.

РОЗДІЛ 2. АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ І ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-ЗАСТОСУНКУ.

2.1. Веб-застосунок та його основний принцип роботи

Перед початком розроблення практичної частини кваліфікаційної роботи необхідно якісно дослідити існуючі архітектурні рішення для підбору програмних засобів для реалізації програмного продукту найбільш відповідними інструментами.

Веб-застосунок — це розподілений застосунок, в якому клієнтом виступає браузер, а сервером — веб-сервер. Браузер може бути реалізацією клієнтів, в яких логіка застосунку зосереджується на сервері, а функція браузера полягає переважно у зображенні інформації, завантаженої мережею з сервера, і передачі назад даних користувача. Варто зазначити найбільшу перевагу такого підходу — клієнти не залежать від конкретної операційної системи користувача. Завдяки цьому веб-застосунки є міжплатформними сервісами. Як наслідок цієї універсальності й відносної простоти розробки, веб-застосунки стали широко популярними в кінці 1990-х та активно використовуються і сьогодні.

Веб-застосунки як програмні продукти мають перевагу, бо при побудові використовують стандартні функції браузерів, незалежно від операційної системи клієнта. Таким чином, застосунок створюється один раз для довільно обраної платформи та розгортається на ній. Не дивлячись на цю істотну перевагу, треба пам'ятати, що користувачі мають можливість змінювати налаштування браузерів, що може вплинути на зовнішній вид створюваних веб-застосунків. Тим не менш, на функціонал це значного впливу не має.

Веб-застосунок отримує запит від клієнта і виконує обчислення, після цього формує веб-сторінку і відправляє її клієнтові мережею з використанням протоколу HTTP. В той же час, веб-застосунок може бути клієнтом інших співпрацюючих служб, наприклад, іншого стороннього веб-застосунку чи бази

зберігання даних. Для роботи з веб-застосунком немає різниці вид операційної системи як розробника так і клієнта.

Для інтерактивності та продуктивності розробки веб-застосунків розроблений новий підхід, названий AJAX, який нині є стандартним. Значною перевагою є використання Ajax, бо, з таким підходом, сторінки здатні відправляти веб-запити до сервера у фоновому режимі, і не перезавантажуються цілком, а лише довантажують необхідні дані з сервера. Це значно пришвидшує роботу, робить її зручнішою та комфортнішою для самого користувача.

Для створення веб-застосунків використовуються різноманітні серверні технології та мови програмування.

Таблиця 2.1 – Серверні технології та мови програмування для веб-застосунків.

Назва	Ліцензія	Веб-сервер
ASP	власницька	спеціалізований
ASP.NET	власницька	спеціалізований
Java	вільна	безліч, зокрема вільних
Groovy	вільна	практично будь-який
Perl	вільна	практично будь-який
PHP	вільна	практично будь-який
Python	вільна	практично будь-який
Ruby	вільна	практично будь-який

Клієнтська частина може використовувати:

- JavaScript;
- Flash;
- Java / JavaFX;
- ActiveX;
- Silverlight.

2.2. Інструменти розробки проектованого веб-застосунку

Розроблення та реалізація якісного програмного продукту передбачає використання стеку сучасних технологій, таких як мова розмітки HTML, CSS, Bootstrap, JavaScript, JS-бібліотеки, такі як JQuery, збирач модулів Webpack, LocalStorage, система контролю версій Git та інші (рис. 2.1). Перераховані та інші сучасні технології забезпечують можливість реалізації технічного завдання кваліфікаційної роботи у вигляді якісного веб-застосунку.



Рисунок 2.1 – Стек використаних технологій

Розмітка. Зазвичай основою, з якої варто почати збирати проект, вважається розроблення макету сторінки за допомогою стандартної мови розмітки HTML (англ. HyperText Markup Language – мова розмітки гіпертексту). HTML – це мова тегів, засобами якої описується структура будь-якої веб-сторінки. Мова розмітки HTML складається з ряду елементів, що підказують браузеру, як саме відображати вміст на сторінці. Браузери отримують HTML-документи з веб-сервера або з локальної пам'яті та передають документи в мультимедійні веб-сторінки [6].

HTML надає інструменти для створення структурованого документу шляхом позначення структурного складу тексту, а саме заголовків, абзаців, списків, таблиць, цитат та іншого. Варто зазначити, що мова розмітки надає можливості отримання інформації з мережі через гіперпосилання, створення інтерактивних форм, а також включення до структури зображень, відео, аудіо та інших об'єктів.

За допомогою HTML розробник декларує те, що подається глядачеві. При цьому він не описує у якому вигляді воно подається. Візуальне уявлення визначається каскадними таблицями стилів (CSS), що реалізуються браузером чи компіляторами.

HTML відіграє центральну роль в побудові DOM (з англ. Document Object Model – “об’єктна модель документа”). DOM – це незалежний від платформи та мови програмний інтерфейс, що дозволяє програмам і скриптам отримати доступ до вмісту HTML-, XHTML- і XML-документів, а також змінювати вміст, структуру та оформлення таких документів.

Модель DOM не накладає обмежень на структуру документа. Будь-який документ відомої структури за допомогою DOM може бути представлений у вигляді дерева вузлів, кожен вузол якого є елементом, атрибутом, текстовим, графічним або будь-яким іншим об’єктом. Вузли пов’язані між собою відносинами «батьківсько-дочірній».

Відповідно до об’єктної моделі документа, кожен HTML-тег є об’єктом, теги є дітьми батьківського елемента, і навіть текст, що знаходиться всередині тега, також є об’єктом. За допомогою JavaScript, ми можемо працювати з цими об’єктами, додавати функціональність веб-застосунку, активувати реакції сторінок на дії користувача.

DOM – це представлення HTML-документу як дерева тегів, що має вигляд та структуру, зображені на рис. 2.2.

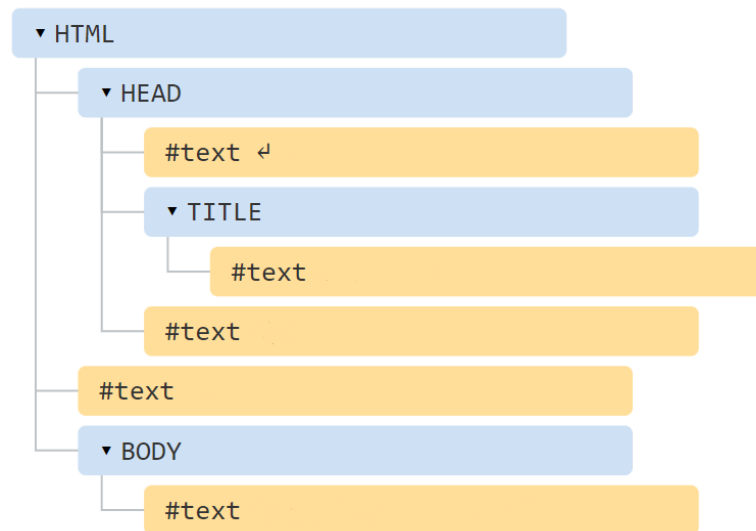


Рисунок 2.2 – Дерево тегів

Стилізація. CSS (англ. Cascading Style Sheets) – це технологія описування зовнішнього вигляду документу, що використовується для опису зовнішнього вигляду на основі мови розмітки HTML. CSS визначає стиль документа – кольори, шрифти, розміри медіафайлів, оздоблення кнопок, розташування об’єктів, зовнішні реакції на натискання, наведення та багато іншого [7].

У даному веб-застосунку мова розмітки HTML допомагає створити основну структуру сторінки, блоки для виведення інформації, поля для введення, кнопки, навігацію та інше. Стили CSS оздоблюють створений проект, забезпечують реакції кнопок на кліки, розташування та переміщення створених об’єктів згідно з розмірами браузерного вікна, стиль сторінки та інтуїтивність інтерфейсу.

Функціонал. Основні дії веб-застосунку були реалізовані за допомогою мови програмування JavaScript. JavaScript (JS) – динамічна, об’єктно-орієнтована прототипна мова програмування. JavaScript є реалізацією стандарту ECMAScript. Найчастіше використовується для створення сценаріїв динамічних веб-сайтів, що надає можливість керувати браузером з боку клієнта, обмінюватися даними з сервером, змінювати зовнішній вигляд та структуру веб-сторінки [8]. Для нашого проекту JavaScript є найзручнішою мовою, бо вона створена саме для написання сценаріїв односторінкових прогресивних веб-застосунків, надання їм

інтерактивності, програмування з боку сервера та ще багато іншого. Більш того, варто зазначити, що розробниками створено чималий масив інструментів додатково до основної мови JavaScript. Вони роблять легкодоступним велику кількість додаткового функціоналу. Саме мова програмування JavaScript змушує розроблюваний веб-застосунок виконувати свої основні функції правильно.

Збереження даних. Збереження інформації у веб-застосунках може реалізовуватись багатьма способами. Одним з найзручніших є збереження даних у LocalStorage. LocalStorage має такі переваги над альтернативними способами збереження Cookies та sessionStorage:

- термін дії зберігання не закінчується автоматично;
- підтримка усіма браузерами;
- немає необхідності передавати дані при кожному HTTP-запиті;
- дані не видаляються автоматично без відома користувача.

Feature	Cookies	localStorage	sessionStorage
Maximum data size –	4 kB	5 MB	5 MB
Blockable by users –	yes	yes	yes
Auto-expire option –	yes	no	yes
Supported data types –	string only	string only	string only
Browser support –	very high	very high	very high
Accessible server-side –	yes	no	no
Data transferred on every HTTP request –	yes	no	no
Editable by users –	yes	yes	yes
Supported on SSL –	yes	n/a	n/a
Can be accessed on –	server-side & client-side	client-side only	client-side only
Clearing / deleting –	PHP, JS & automatic	JS only	JS & automatic
Lifetime –	as specified	till deleted	till tab is closed
Secure data storage –	no	no	no

Рисунок 2.3 – Порівняння способів локального збереження даних у браузері з альтернативними

Вміст LocalStorage не передається на сервер напряму. З цього випливає декілька додаткових переваг [9].

По-перше, неможливо спровокувати сховище на передачу за допомогою CSRF. CSRF (англ. Cross-Site Request Forgery) – небезпечна атака, яка призводить до того, що хакер може виконати на веб-сайті безліч різних дій, не передбачених розробником для виконання. Використання LocalStorage допомагає попередити цю загрозу, так як, як було зазначено вище, не передає дані напряму на сервер, що попереджає виконання загрозливих дій.

Також, у LocalStorage можна зберігати більше інформації ніж у відомих cookies, а також не хвилюватись, що ця інформація буде передаватися з кожним запитом на сервер, не раціонально використовуючи вихідний трафік.

Більш того, за останніми даними, LocalStorage підтримується усіма браузерами, як на персональних комп'ютерах, так і на мобільних пристроях.

	🖥️						📱					
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	WebView Android	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet
localStorage	4	12	3.5	8	10.5	4	37	18	4	11	3.2	1.0

Рисунок 2.4 – Підтримка LocalStorage браузерами

Додаткові інструменти. Також, у даному проекті була використана бібліотека jQuery. jQuery – популярна JavaScript-бібліотека з відкритим кодом.

Синтаксис jQuery ретельно розроблений так, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, обробці подій, створенню анімації та розробці Аjax-застосунків. Розробники можуть утворювати абстракції для низькорівневої взаємодії та анімацію для ефектів високого рівня. Це сприяє створенню потужних динамічних веб-застосунків.

Бібліотека jQuery є JavaScript файлом, яка включає всю його DOM, події, ефекти та Ajax-функції. Вона може бути додана до веб-застосунку посиланням на локальну копію, або на одну з копій доступних на публічному сервері. Але кращим рішенням є завантаження файлу бібліотеки, щоб попередити непередбачувані проблеми.

Ще однією перевагою є те, що дана бібліотека підтримується всіма браузерами як на мобільних пристроях, так і на персональних комп'ютерах.

Розроблення проекту передбачає використання додаткових JavaScript інструментів, таких як збирач модулів Webpack, Jsdом та інші. Також, для покращення роботи самого інтерфейсу був застосований вільний набір інструментів для створення веб-застосунків Bootstrap. Bootstrap включає в себе HTML- і CSS-шаблони оформлення для веб-форм, кнопок, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ ПОШУКУ РОБОЧИХ МІСЦЬ З ІТ СПЕЦІАЛЬНОСТЕЙ.

3.1 Структурна карта веб-застосунку

Правильна та якісна побудова веб-застосунку забезпечує послідовну роботу та правильну взаємодію користувача з системою. Розроблюваний застосунок має надавати користувачу можливість інтуїтивно взаємодіяти з ним за допомогою зручного інтерфейсу та безперешкодно отримувати результати пошуку. Не менш важливо забезпечити зручний перехід від одного розділу до іншого.

Отже, як висновок, карта системи має складатись з наступних складових:

- верхня навігаційна панель;
- перелік усіх вакансій;
- поп-ап вакансій;
- статистика з фільтрацією;
- перелік персональної інформації користувача;
- перелік вподобаних пропозицій;
- нижня навігаційна панель;
- форма надання персональної інформації.

СТРУКТУРНА КАРТА ВЕБ- ЗАСТОСУНКУ



Рисунок 3.1 – Структурна карта веб-застосунку

3.2 Функціональне поле розроблюваного застосунку

Задля якісного створення проекту, що відповідатиме поставленим задачам кваліфікаційної роботи, був проведений ґрунтовний аналіз існуючих аналогів програмного забезпечення. Результати дослідження було описано вище в попередніх розділах.

Базуючись на висновках аналізу було розроблено функціонал, що не буде ускладнювати взаємодію користувача із застосунком та зможе задовольнити мету його створення.

Основний функціонал застосунку подано у вигляді таблиці (табл. 3.1).

Таблиця 3.1 – Функціонал веб-застосунку

Назва функціональної можливості програмного продукту	Загальний опис функціональних можливостей
Перегляд відкритих вакансій	Загальний список вакансій з бази даних. Завантажується автоматично при завантаженні сторінки браузера. Не змінюється після оновлення сторінки. Не піддається зміні даних від користувачів, як захист від хакерських атак.
Рор-уп вакансій	Користувач має можливість по кліку на пропозицію відкрити рор-уп з її детальною інформацією.
Додавання вакансії в розділ «вподобане» за особистим інтересом користувача	До кожного об'єкту додана кнопка «Вподобати». Після її натиснення вакансія додається в список «Вподобаного» та відображається в полі для обраних вакансій нижче.
Прибирання «вподобаного»	По повторному кліку на кнопку «Вподобати», одиниця прибирається зі списку. Також, можна перейти в розділ «Вподобані» та натиснути кнопку видалення потрібної пропозиції.

Сортування вакансій за різними параметрами	По кліку на назву стовпчика дані таблиці сортуються.
Пошук	У розділі «Статистика» надана можливість пошуку позицій по назві та заробітній платі.
Фільтрація вакансій	Веб-застосунок пропонує користувачу здійснювати фільтрацію пропозицій за різними категоріями для полегшення пошуку в розділі «Статистика».
	+ інший функціонал для зручності роботи та створення інтуїтивного інтерфейсу.

Також додано функціонал, що полегшує пересування по веб-застосунку такий як, кнопка «Вгору», що повертає користувача до навігаційної панелі, та поява прихованого меню при зміні розмірів вікна браузера.

3.3 Збереження даних

Зберігання даних веб-застосунку – одне із ключових питань, що варто уважно дослідити перед створенням продукту. Місце зберігання даних (база даних) – інструмент зберігання великих обсягів інформації у вигляді текстових структур.

Відповідно до поставлених задач в нашому випадку, інформація про вакансії та посилання на зображення, зберігається в файлі формату json. Кожна вакансія зберігається як неповторний об'єкт. Це полегшує роботу з даними, їх аналіз та виведення на сторінку. Варто зазначити, що кожен об'єкт має свій індивідуальний id номер, що і дає змогу маніпулювати всіма об'єктами одночасно, швидко, не витрачаючи час на перебір всіх елементів під час пошуку чи сортування. Саме з цієї структури дані передаються далі, аналізуються скриптами, що здійснюють управління цією інформацією.

З метою оптимізації роботи застосунку використовується інструмент зберігання даних на локальному сервері у вигляді LocalStorage (рис. 3.2). У

застосунку реалізується запис вподобаних вакансій у вигляді їх id та кількості кліків на кнопку «Додати», щоб попередити повторне додавання вакансії у список. При перезавантаженні сторінки дані будуть виводитись саме з цього місця зберігання. Це, як і було зазначено вище, значно прискорює роботу застосунку та допомагає забезпечити безпеку даних клієнта від витягування їх зловмисником, так як доступ до вподобаних пропозицій має виключно користувач персонального комп'ютера. Саме так, на даному етапі зберігається і інформація клієнта, що буде передаватись на сервер у випадку необхідності додавання інструментарію збору статистики та інформації користувачів. Така інформація легко піддається змінам і не має обов'язково повторно витягатись з бази після кожного оновлення сторінки. Якщо інформація на сервері змінам не підлягала, вона може братися з локального сторедржа, економлячи час і ресурси комп'ютера. Таким чином, навіть якщо якість техніки в користувача погана, це не завадить йому отримувати дані швидко без додаткових очікувань. А як було доведено в попередніх розділах, час очікування може зіпсувати досвід користувача і ставлення до всього продукту в цілому.

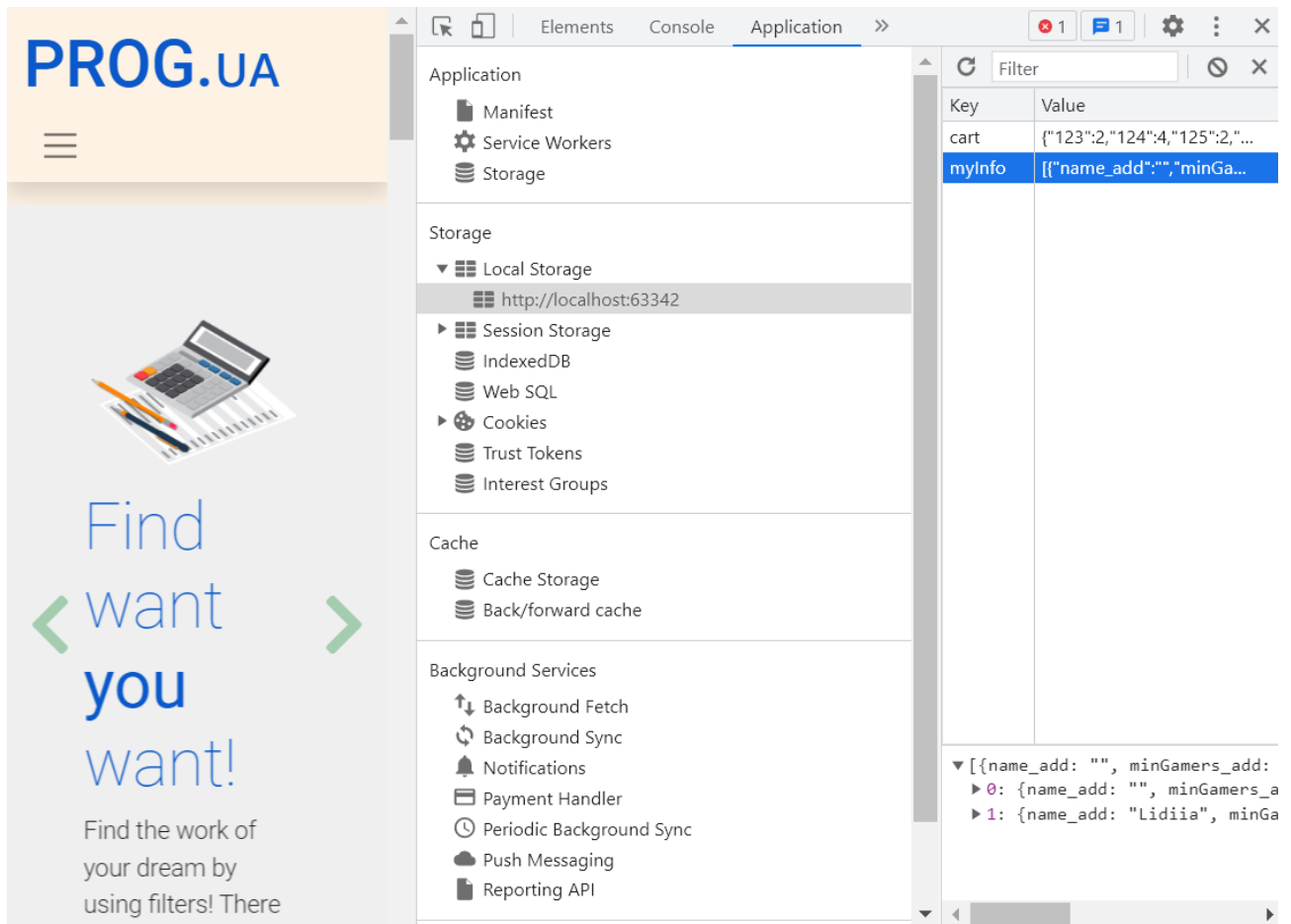


Рисунок 3.2 – Приклад зберігання інформації про вакансію у Local Storage.

Більш того, завдяки гнучкості цього способу зберігання інформації, користувач може легко здійснювати їхнє сортування, видалення, пошук та інші необхідні дії.

3.4 Інструкція користувача

Веб-застосунок був створений з метою підвищення ефективності пошуку у робочих місць з ІТ спеціальностей.

Інтерфейс даного проекту був розроблений відповідно до сучасних вимог та рекомендацій, основною метою яких є створення інтуїтивних, зручних для користувачів систем.

Основна кольорова гамма була підібрана згідно з останніми дослідженнями за відсутності дратівливих кольорів, що допомагає позбутися від

тривожності, хвилювань, гарантуючи комфорт користувачам у роботі з веб-застосунком [10]. Вибір професії є непростою задачею, саме тому, створений продукт має не тільки не дратувати, але, більш того, гарантувати користувачу почуття безпеки і легкості.

Стартовий вигляд сторінки представлено на рис. 3.3.

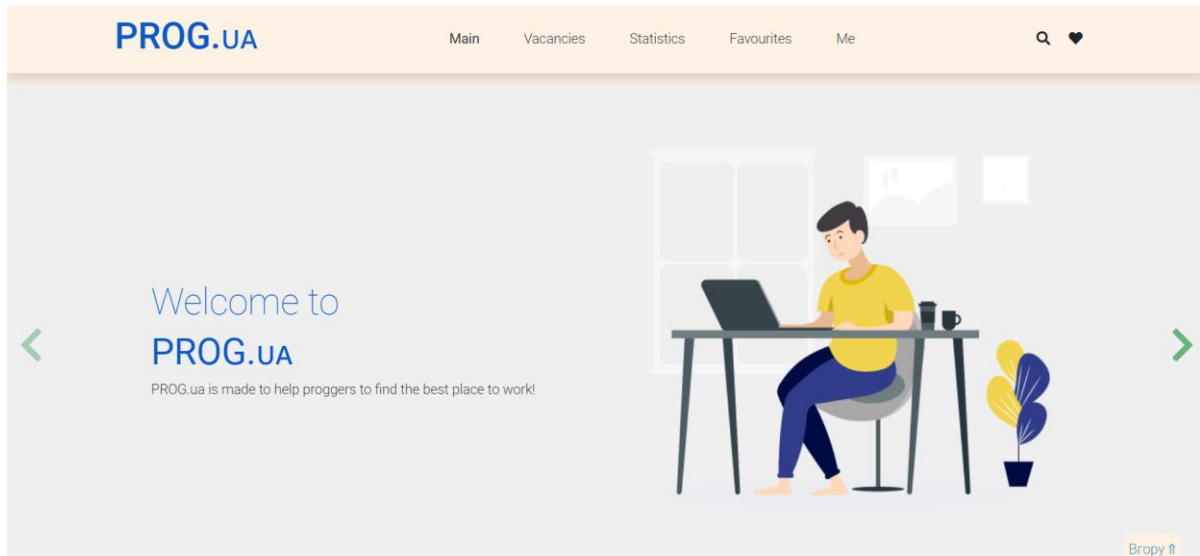


Рисунок 3.3 – Стартова сторінка веб-застосунку

Сторінка починається з верхньої навігаційної панелі (рис. 3.4). Користувач одразу може перейти до необхідного розділу, якщо має таку мету.



Рисунок 3.4 – Навігаційна панель

При натисканні на емблеми соціальних мереж над навігаційною панеллю користувач переходить на рекламну сторінку проекту в обраній соціальній інтернет мережі (рис. 3.5).

За необхідності, при зменшенні екрану, дана панель зникає, таким чином забезпечуючи простір для більш важливої інформації. Тим не менш, дані посилання залишаються на нижній панелі, тому, при необхідності, користувач в будь-якому випадку зможе зв'язатись з розробником або власником компанії.



Рисунок 3.5 – Емблеми соціальних мереж

Під навігаційною панеллю знаходиться інформаційна слайдер-панель з корисною інформацією для користувача (рис. 3.6-3.7). Така панель виконує роль оповіщення користувачів про новини, правила користування, цікаву інформацію або заклики до дій.

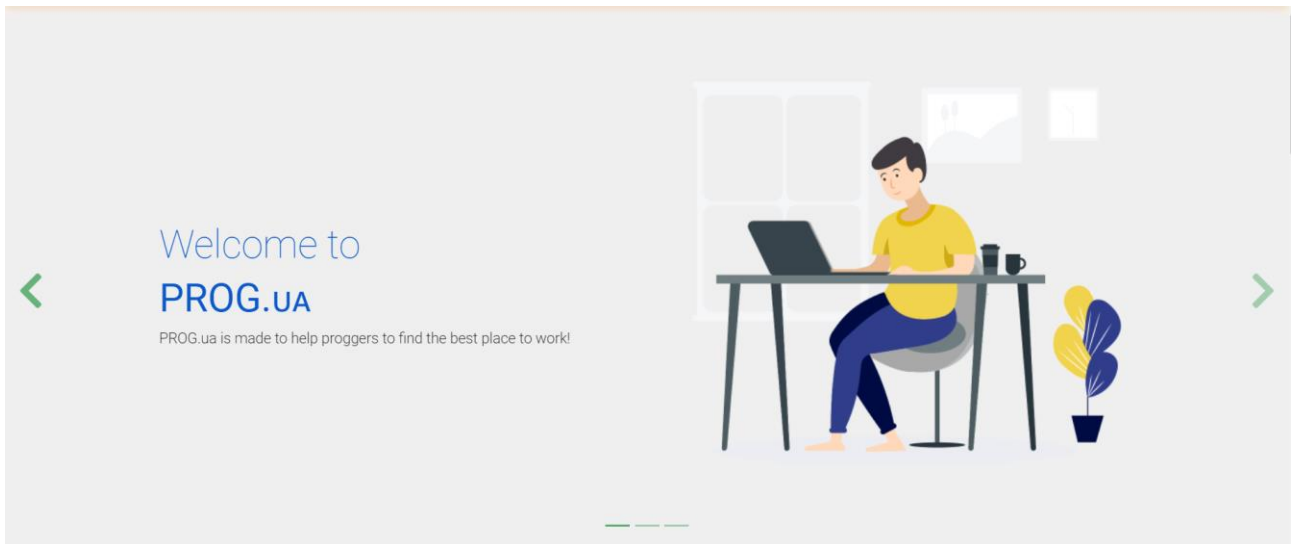


Рисунок 3.6 – Інформаційний слайдер (Слайд 1)

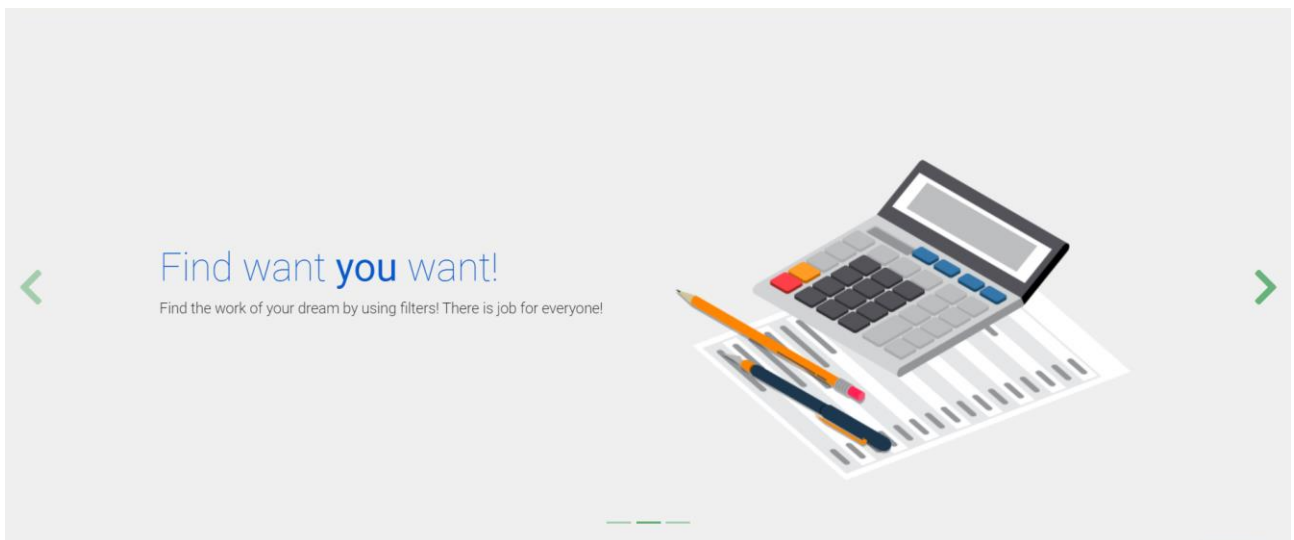


Рисунок 3.7 – Інформаційний слайдер (Слайд 2)

При необхідності зміни інформації, слайди легко можуть бути додані розробником програми або технічною підтримкою.

Під інформаційною панеллю користувач має змогу переглянути список наявних вакансій. Кожна вакансія містить свою емблему, назву, короткий опис. Більш того, користувач вже на даному етапі може вподобати вакансію, що перенесе її у список вподобаних, або ж переглянути додаткову інформацію натиснувши на кнопку під обраною пропозицією. Поп-ап легко закривається кліком на вільне місце на екрані, що робить такий функціонал інтуїтивно зрозумілим.

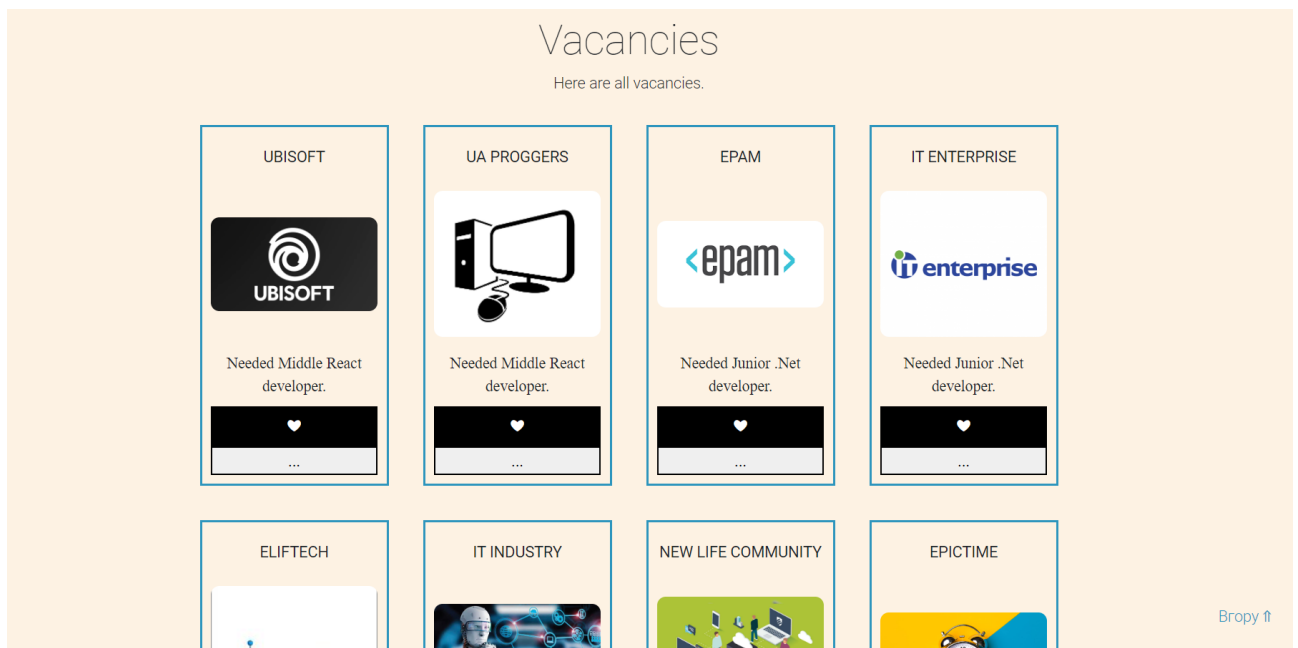


Рисунок 3.8 – Список вакансій

Наступним блоком ми маємо можливість переглянути поле статистики по вакансіям (рис. 3.9). Даний функціонал допоможе користувачу швидко орієнтуватися серед великої кількості вакансій переглядаючи їх за допомогою наданих фільтрів.

Тут існує фільтрація за назвою компанії, за обсягом обіцяної зарплати, за необхідністю мати досвід, пошук за назвою. Даний функціонал є основою будь-якого пошукового застосунку.

Statistics

Find appropriate using filters.


Company	Salary	Description	Experience
Ubisoft	400	Needed Middle React developer.	2.5+ years
UA Progers	800	Needed Middle React developer.	2+ years
EPAM	200	Needed Junior .Net developer.	Is not needed
IT Enterprise	400	Needed Junior .Net developer.	Is not needed
Eliftech	200	Needed Junior .Net developer.	Is not needed
It Industry	800	Needed Middle React developer.	Is not needed
New Life Community	600	Needed Middle React developer.	3+ years
EpicTime	900	Needed Middle React developer.	1.5+ years

Рисунок 3.9 – Блок статистики

Наступна панель відображає список вподобаних вакансій (рис. 3.10). Користувач може видалити вакансію з улюблених кліком на кнопку.

Favourites


There are your favourites vacancies.



NEW LIFE
COMMUNITY

Salary: 600

Remove from favs



ЕPICTIME

Salary: 900

Remove from favs

Рисунок 3.10 – Блок улюблених

Цікавим для користувача функціоналом є вкладка «Я» (рис. 3.11).

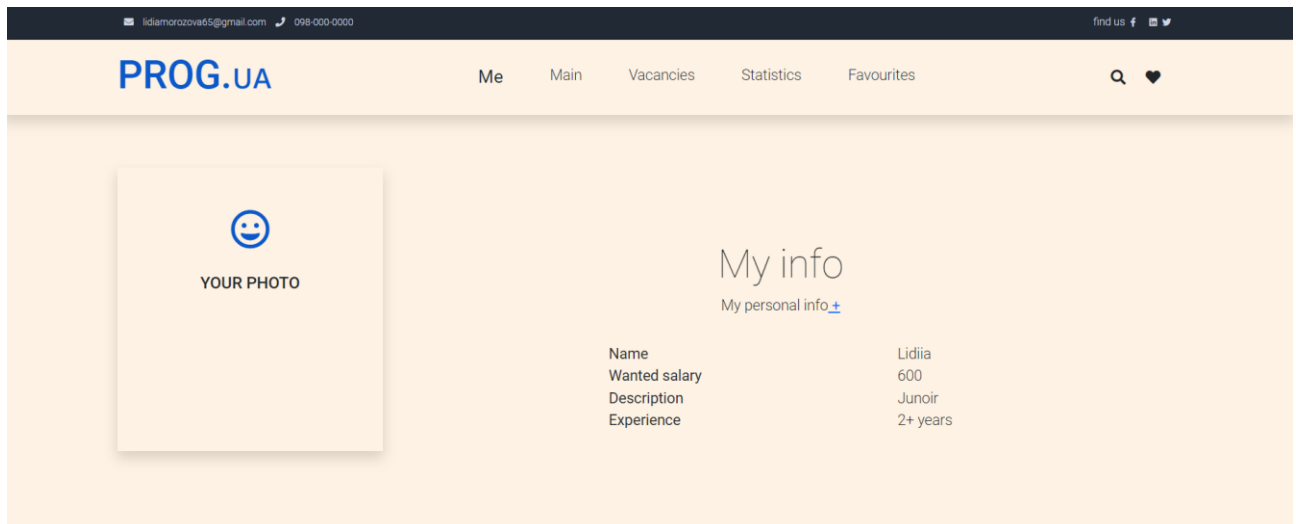


Рисунок 3.11 – Сторінка з персональною інформацією

Розглянувши дану сторінку ми можемо побачити присутню панель з контактами. Також, користувач може скористатися описаною вище верхньою навігаційною панеллю.

Користувач може додати своє фото на сторінку та персональну інформацію. У випадку відсутності інформації з'являється кнопка з цією можливістю (рис. 3.12).

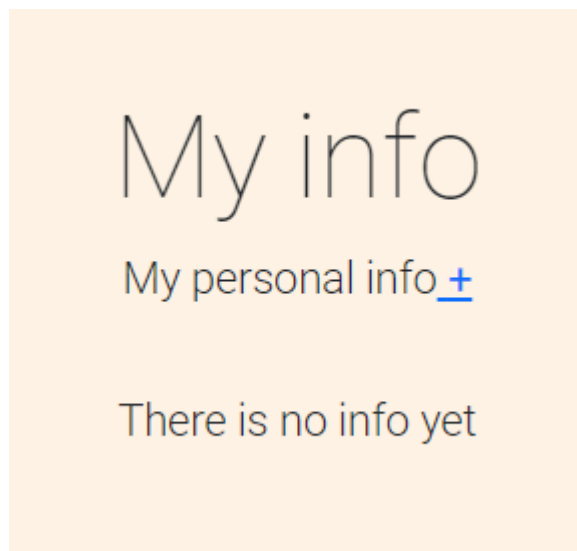


Рисунок 3.12 – Сторінка з відсутньою персональною інформацією

Нижче користувач бачить слайдер панель зі стекком технологій, що змінюють свій колір при наведенні (рис. 3.13).



Рисунок 3.13 – Слайдер – стек технологій

У правому нижньому кутку застосунку при прокручуванні вниз з'являється кнопка «Вгору», що дозволяє користувачу повернутись на стартову позицію.

У самому низу веб-застосунку розташована нижня навігаційна панель, створена з метою полегшення пересування користувача (рис. 3.14). Нижня панель містить навігацію, додаткову інформацію та посилання на соціальні мережі, описані вище.

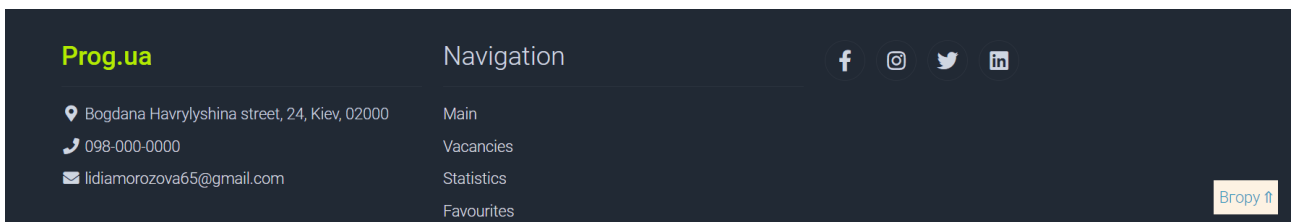


Рисунок 3.14 – Нижня навігаційна панель з кнопкою «Вгору»

Насамкінець, необхідно зазначити, що емблема сторінки має посилання на головну сторінку. Таким чином, користувач матиме змогу повернутись на початок у будь-який момент (рис. 3.15).



Рисунок 3.15 – Емблема з посиланням

ВИСНОВКИ

На основі огляду та аналізу трьох систем пошуку робочих місць в онлайн-режимі, що були знайдені за допомогою пошукового сервісу Google, було визначено низку переваг та недоліків у веб-системах такого формату за чітко виділеними основними критеріями. Проаналізувавши усі знайдені недоліки, було описано шляхи удосконалення таких систем. Дане розуміння дало змогу створити список критеріїв та очікувань від реалізованої в ході написання кваліфікаційної роботи програми.

У ході роботи було досліджено теоретичні основи побудови веб-застосунків, проведено аналіз програмно-технологічного інструментарію забезпечення програмного продукту, здійснено проектування та реалізацію планованого веб-застосунку.

В результаті ретельного огляду літератури за темою «Побудова веб-застосунків для пошуку робочих місць з ІТ спеціальностей» було оцінено сучасний стан веб-сайтів для працевлаштування. Таким чином були сформовані нові, відносно існуючих рішень, вимоги до системи, яка розробляється.

В подальшому застосунок може бути покращено та впроваджено на ресурс загального доступу користувачів завдяки можливостям платформи Git.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ринок праці: статистика, перспективи та можливості воєнного часу [Електронний ресурс] // Суспільство. – 2022. – Режим доступу до ресурсу: <http://www.golos.com.ua/article/357720>.
2. Аналітична та статистична інформація [Електронний ресурс] // Державний центр зайнятості. – 2022. – Режим доступу до ресурсу: <https://www.dcz.gov.ua/analytics/69>.
3. Статистика сайту [Електронний ресурс] // work.ua. – 2022. – Режим доступу до ресурсу: <https://www.work.ua/ru/about-us/stat/>.
4. Скільки IT-спеціалістів в Україні: +29 тисяч за рік згідно з Мін'юстом [Електронний ресурс] // DOU. – 2021. – Режим доступу до ресурсу: <https://dou.ua/lenta/articles/how-many-devs-in-ukraine-2020/>.
5. Як змінився ринок праці в IT через війну [Електронний ресурс] // Українська правда. – 2022. – Режим доступу до ресурсу: <https://www.epravda.com.ua/columns/2022/06/8/687956/>.
6. HTML [Електронний ресурс] // Вільна енциклопедія Wikipedia. – 2021. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML>.
7. CSS [Електронний ресурс] // Вільна енциклопедія Wikipedia. – 2020. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/CSS>.
8. JavaScript [Електронний ресурс] // Вільна енциклопедія Wikipedia – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/JavaScript>.
9. Local Storage [Електронний ресурс] // Get revising – Режим доступу до ресурсу: https://getrevising.co.uk/grids/local_storage.
10. How to build a good website [Електронний ресурс] // w3school – Режим доступу до ресурсу: https://www.w3schools.com/howto/howto_website.asp.
11. Top 10 Tips to increase the performance of your JavaScript app in 2021 [Електронний ресурс] // TechMagic. – 2021. – Режим доступу до ресурсу: <https://blog.techmagic.co/top-10-tips-to-increase-the-performance-of-your-javascript-app-in-2020/>.

12. Прогресивні веб застосунки [Електронний ресурс] // - 2022р. - Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps.
13. 15 провідних тенденцій веб розробки у 2020 році [Електронний ресурс] // - 2020р. - Режим доступу до ресурсу: <https://web4u.in.ua/blog/15-prov-dnih-tendenc-y-veb-rozrobki-u-2020-roc-31> .
14. Пошук вакансій за категоріями [Електронний ресурс] // Work.ua. – 2022. – Режим доступу до ресурсу: <https://www.work.ua/ru/jobs/by-category/>.
15. Веб-безпека. Різновиди веб-систем та їх основні вразливості [Електронний ресурс]. Режим доступу до ресурсу: <http://websecurity.com.ua/security/chapter1/>
16. 100 сочетаний цветов, которые привлекут нужную аудиторию [Електронний ресурс] // Canva – Режим доступу до ресурсу: https://www.canva.com/ru_ru/obuchenie/100-cvetovyx-sochetnij/.
17. Блакитний колір в психології: значення, що символізує, характеристика [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.kozaky.org.ua/blakitnij-kolir-v-psixologi%D1%97-znachennya-shho-simvolizuye-harakteristika/>.

ДОДАТКИ
ДОДАТОК А

```
{
  '111': {
    'name': 'EPAM',
    'salary': 200,
    "description": "Needed Junior .Net developer with Upper Intermediate level of
English. Communication with colleges is needed.",
    "experience": "Is not needed",
    "img" : "epam.png"
  },

  '112': {
    'name': 'Eliftech',
    'salary': 300,
    "description": "Needed Junior .Net developer. We offer you quick growth of your
technical skills and salary.",
    "experience": "Is not needed",
    "img" : "epam.png"
  },

  '113': {
    'name': 'NewLife',
    'salary': 500,
    "description": "Needed Middle React developer.",
    "experience": "Is not needed",
    "img" : "epam.png"
  },

  '114': {
```

```

'name': 'IT Enterprise',
'salary': 400,
'description': "Needed Junior .Net developer with Upper Intermediate level of
English. Communication with colleges is needed.",
'experience': "Is not needed",
'img' : "epam.png"
},

"115": {
'name': "UA Proppers",
'salary': 800,
'description': "Needed Middle React developer.",
'equipment': "2+ years",
'img' : "comp.png"
}
}

```

Додаток А1 – Зберігання даних про компанії у вигляді об’єктів.

```

<nav class="navbar navbar-expand-lg bg-dark navbar-light d-none d-lg-block"
id="templatemo_nav_top">
  <div class="container text-light">
    <div class="w-100 d-flex justify-content-between">
      <div>
        <i class="fa fa-envelope mx-2"></i>
        <a class="navbar-sm-brand text-light text-decoration-none"
href="mailto:lidiamorozova65@gmail.com">lidiamorozova65@gmail.com</a>
        <i class="fa fa-phone mx-2"></i>
        <a class="navbar-sm-brand text-light text-decoration-none" href="tel:098-
000-0000">098-000-0000</a>

```

```

</div>
<div>
  find us
  <a class="text-light" href="https://www.facebook.com/kyiv.university/"
target="_blank" rel="sponsored"><i class="fab fa-facebook-f fa-sm fa-fw me-
2"></i></a>
  <a class="text-light" href="https://www.linkedin.com/" target="_blank"><i
class="fab fa-linkedin fa-sm fa-fw"></i></a>
  <a class="text-light" href="https://twitter.com/" target="_blank"><i
class="fab fa-twitter fa-sm fa-fw me-2"></i></a>
  <div class="shareButtons">
  </div>
</div>
</div>
</div>
</nav>

<nav class="navbar navbar-expand-lg navbar-light shadow">
  <div class="container d-flex justify-content-between align-items-center">

    <a class="navbar-brand text-success logo h1 align-self-center"
href="index.html">
      PROG.&#x1F1FA;&#x1F1E6;
    </a>

    <button class="navbar-toggler border-0" type="button" data-bs-
toggle="collapse" data-bs-target="#templatemo_main_nav" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>

```

```
</button>
```

```
<div class="align-self-center collapse navbar-collapse flex-fill d-lg-flex justify-content-lg-between" id="templatemo_main_nav">
```

```
<div class="flex-fill">
```

```
<ul class="nav navbar-nav d-flex justify-content-between mx-lg-auto">
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="#main"><b>Main</b></a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="#games">Vacancies</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="#stats">Statistics</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="#favs">Favourites</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="../about.html">Me</a>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
<div class="navbar align-self-center d-flex">
```

```
<a class="nav-icon position-relative text-decoration-none" href="#stats">
```

```
<i class="fa fa-fw fa-search text-dark mr-1"></i>
```

```
<span class="position-absolute top-0 left-100 translate-middle badge rounded-pill bg-light text-dark"></span>
```

```
</a>
```

```

    <a class="nav-icon position-relative text-decoration-none" href="#favs">
      <i class="fa fa-fw fa-heart text-dark mr-1"></i>
      <span class="position-absolute top-0 left-100 translate-middle badge
rounded-pill bg-light text-dark"></span>
    </a>
  </div>
</div>
</nav>

```

Додаток А2 – Верхня навігаційна панель.

```

<footer class="bg-dark" id="tempaltemo_footer">
  <div class="container">
    <div class="row">
      <div class="col-md-4 pt-5">
        <h2 class="h2 text-success border-bottom pb-3 border-light logo"
style="color: #60a2ff !important;">Prog.ua</h2>
        <ul class="list-unstyled text-light footer-link-list">
          <li>
            <i class="fas fa-map-marker-alt fa-fw"></i>
            Bogdana Havrylyshina street, 24, Kiev, 02000
          </li>
          <li>
            <i class="fa fa-phone fa-fw"></i>
            <a class="text-decoration-none" href="tel:098-000-0000">098-000-
0000</a>
          </li>

```

```

    <li>
      <i class="fa fa-envelope fa-fw"></i>
      <a class="text-decoration-none"
href="mailto:lidiamorozova65@gmail.com">lidiamorozova65@gmail.com</a>
    </li>
  </ul>
</div>

<div class="col-md-4 pt-5">
  <h2 class="h2 text-light border-bottom pb-3 border-
light">Navigation</h2>
  <ul class="list-unstyled text-light footer-link-list">
    <li><a class="text-decoration-none" href="#main">Main</a></li>
    <li><a class="text-decoration-none" href="#games">Vacancies</a></li>
    <li><a class="text-decoration-none" href="#stats">Statistics</a></li>
    <li><a class="text-decoration-none" href="#favs">Favourites</a></li>
    <li><a class="text-decoration-none" href="#myGames">Me</a></li>
  </ul>
</div>

<div class="col-md-4 pt-5">
  <ul class="list-inline text-left footer-icons">
    <li class="list-inline-item border border-light rounded-circle text-
center">
      <a class="text-light text-decoration-none" target="_blank"
href="http://facebook.com/"><i class="fab fa-facebook-f fa-lg fa-fw"></i></a>
    </li>
    <li class="list-inline-item border border-light rounded-circle text-
center">
      <a class="text-light text-decoration-none" target="_blank"
href="https://www.instagram.com/"><i class="fab fa-instagram fa-lg fa-

```

```

fw"></i></a>
    </li>
    <li class="list-inline-item border border-light rounded-circle text-
center">
        <a class="text-light text-decoration-none" target="_blank"
href="https://twitter.com/"><i class="fab fa-twitter fa-lg fa-fw"></i></a>
    </li>
    <li class="list-inline-item border border-light rounded-circle text-
center">
        <a class="text-light text-decoration-none" target="_blank"
href="https://www.linkedin.com/"><i class="fab fa-linkedin fa-lg fa-fw"></i></a>
    </li>
</ul>
</div>
</div>
</div>
</div>
</footer>

```

Додаток А3 – Нижня навігаційна панель.

```

<div class="addModal" id="add">
    <a href=""> <span class="close" id="close_add" >&times;</span></a>
    <div class="popupContainer">
        <h1 class="h1">ADD YOUR INFO</h1>
        <label for="name_add">Name</label><br>
        <input class="formInput" id="name_add" type="text" name="name_add"
autofocus required><br>

        <label for="add">Wanted salary</label><br>
        <input class="formInput" id="minGamers_add" type="number"

```

```
name="minGamers_add" min="1" max="100" required><br>
```

```
<label for="desc_add">About me</label><br>
```

```
<input class="formInput" id="desc_add" type="text" name="desc_add"
required><br>
```

```
<label for="equipment_add">Experience</label><br>
```

```
<input class="formInput" id="equipment_add" type="text"
name="equipment_add" required><br>
```

```
<div class="submitButton">
```

```
<input type="submit" value="SAVE" class="submitButton add button"
id="add">
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="modal-overlay"></div>
```

Додаток А4 – Модальне вікно для введення особистої інформації.

```
<section class="bg-light py-5">
```

```
<div class="container my-4">
```

```
<div class="row text-center py-3">
```

```
<div class="col-lg-6 m-auto">
```

```
<p>
```

Choose your knowledge stack

```
</p>
```

```
</div>
```

```
<div class="col-lg-11 m-auto tempaltemo-carousel">
```

```
<div class="row d-flex flex-row">
```

```

<div class="col-1 align-self-center">
  <a class="h1" href="#templatemo-slide-brand" role="button" data-bs-
slide="prev">
    <i class="text-light fas fa-chevron-left"></i>
  </a>
</div>

```

```

<div class="col">
  <div class="carousel slide carousel-multi-item pt-2 pt-md-0"
id="templatemo-slide-brand" data-bs-ride="carousel">
  <div class="carousel-inner product-links-wap" role="listbox">
    <div class="carousel-item active">
      <div class="row" id="carouselLogos">
        <div class="col-3 p-md-5">
          <a href="#"></a>
        </div>
        <div class="col-3 p-md-5">
          <a href="#"></a>
        </div>
        <div class="col-3 p-md-5">
          <a href="#"></a>
        </div>
        <div class="col-3 p-md-5">
          <a href="#"></a>
        </div>
      </div>
    </div>
  </div>

```



```
var allAvailables = [];
```

```
function init() {
    $.getJSON('vacans.json', vacansOut);
    $.getJSON('vacans.json', function (json) {
        for (var key in json) {
            if (json.hasOwnProperty(key)) {
                var item = json[key];
                allAvailables.push({
                    name: item.name,
                    salary: item.minGamers,
                    description: item.description,
                    experience: item.equipment,
                    img: item.img
                });
            }
        }
    });
    console.log(allAvailables);
}
```

```
function vacansOut(data) { // ВИВІД НА СТОРІНКУ
    var out="";

    for (var key in data) {
        out += '<div class="cart">';
        out += `<p class="name"><b>${data[key].name}</b></p>`;
        out += '<div class="cartImg">';
        out += ``;
    out += '</div>';
    out += '<div class="blockBtns">';
    out += '<div class="description">${data[key].description}</div>`;

    out += '<button class="add-to-cart" id="${key}">#10084;</button>`;
    out += '<button class="myBtn" id="${key}">...</button>`;
    out += '</div>';
    out += '</div>';
}
$('.goods-out').html(out);
$('.add-to-cart').on('click', addToCart);
$('.myBtn').on('click', showPopUp)

getStatistics(data);
load();
}

```

Додаток А6 – Вивід усіх доступних вакансій.

```

function showPopUp(){
    var full_id = $(this).attr('id');
    let id = full_id.split("").pop();
    let final_id;

    if (id === '4')
        final_id = '0';
    else if (id === '5')
        final_id = '1';
    else if (id === '6')

```

```

    final_id = '2';
else if (id === '7')
    final_id = '3';
else if (id === '8')
    final_id = '4';
else if (id === '9')
    final_id = '5';
else if (id === '0')
    final_id = '6';
else if (id === '1')
    final_id = '7';

var out="";
out +=`<div class='modalOpenedCard'>`;
out +=`<div class='h1'>${allAvaliables[final_id].name}</div>`;
    out +=`<div class="description">Salary:
${allAvaliables[final_id].minGamers}</div>`;
    out +=`<div class="description">Experience:
${allAvaliables[final_id].equipment}</div>`;
    out +=`<div class="description">Description:
${allAvaliables[final_id].description}</div>`;
    out +=`</div>`;
    out +=``;
    out +=`<span class="close" id="closeModalContent">&times;</span>`;

$('.modal-content').html(out);

```

```
var modal = document.getElementById("myModal");
modal.style.display = "block";
```

```
var span = document.getElementById("closeModalContent");
```

```
span.onclick = function() {
    modal.style.display = "none";
}
```

```
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
}
```

Додаток А7 – Поп-ап вакансій.

```
function getStatistics(data){
    var out="";

    out +=`<tr>`;
    out +=`<th onclick="sortTable(0)">Company</th>`;
    out +=`<th onclick="sortTable(1)">Salary</th>`;
    out +=`<th>Description</th>`;
    out +=`<th>Experience</th>`;
    out +=`</tr>`
```

```

for (var key in data) {
    out +=`<tr>`;
    out +=`<td>${data[key].name}</td>`;
    out +=`<td>${data[key].salary}</td>`;
    out +=`<td>${data[key].experience}</td>`;
    out +=`<td>${data[key].equipment}</td>`;
    out +=`</tr>`;
}
$('.table_statistics').html(out);
}

function sortTable(n){
    var table, rows, switching, i, x, y, shouldSwitch, dir, switchcount = 0;
    table = document.getElementById("table_statistics");
    switching = true;

    dir = "asc";
    while (switching) {
        switching = false;
        rows = table.getElementsByTagName(`tr`);
        for (i = 1; i < rows.length - 1; i++) {
            shouldSwitch = false;
            x = rows[i].getElementsByTagName(`td`)[n];
            y = rows[i + 1].getElementsByTagName(`td`)[n];

            if (dir == "asc") {
                if (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase()) {
                    shouldSwitch = true;
                    break;
                }
            }
        }
    }
}

```

```

    } else if (dir == "desc") {
        if (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {
            shouldSwitch = true;
            break;
        }
    }
}
if (shouldSwitch) {
    rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
    switching = true;
    switchcount++;
} else {
    if (switchcount == 0 && dir == "asc") {
        dir = "desc";
        switching = true;
    }
}
}
}

```

Додаток А8 – Сортування статистики.

```

function showMyInfo(){
    if (localStorage.getItem('myInfo')) {
        myInfo = JSON.parse(localStorage.getItem('myInfo'));
        console.log(myInfo)
    }
    var out = "";
    if(myInfo[1]!=null){
        out +=`<div class="namesMyInfo">`;
    }
}

```

```

    out += `

<b>Name</b></div>`;
    out += `

<b>Wanted salary</b></div>`;
    out += `

<b>Description</b></div>`;
    out += `

<b>Experience</b></div>`;
    out += `</div>`;
    out += `

Додаток А9 – Вивід персональної інформації.



```

<div id="carousel" class="carousel slide" data-bs-ride="carousel">
 <ol class="carousel-indicators">
 <li data-bs-target="#template-mo-zay-hero-carousel" data-bs-slide-to="0"
class="active">
 <li data-bs-target="#template-mo-zay-hero-carousel" data-bs-slide-to="1">
 <li data-bs-target="#template-mo-zay-hero-carousel" data-bs-slide-to="2">

 <div class="carousel-inner">

```


```

```

<div class="carousel-item active">
  <div class="container">
    <div class="row p-5">
      <div class="mx-auto col-md-8 col-lg-6 order-lg-last">
        
      </div>
      <div class="col-lg-6 mb-0 d-flex align-items-center">
        <div class="text-align-left align-self-center">
          <h1 class="h1 text-success">Welcome to</h1>
          <h1 class="h1 text-
success"><b>PROG.&#x1F1FA;&#x1F1E6;</b></h1>
          <p>
            PROG.ua is made to help proggers to find the best place to work!
          </p>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="carousel-item">
  <div class="container">
    <div class="row p-5">
      <div class="mx-auto col-md-8 col-lg-6 order-lg-last">
        
      </div>
      <div class="col-lg-6 mb-0 d-flex align-items-center">
        <div class="text-align-left">
          <h1 class="h1 text-success">Find want <b>you</b> want!</h1>
          <p>
            Find the work of your dream by using filters! There is job for

```

everyone!

```

        </p>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>

</div>
<a class="carousel-control-prev text-decoration-none w-auto ps-3"
href="#template-mo-zay-hero-carousel" role="button" data-bs-slide="prev">
    <i class="fas fa-chevron-left"></i>
</a>
<a class="carousel-control-next text-decoration-none w-auto pe-3"
href="#template-mo-zay-hero-carousel" role="button" data-bs-slide="next">
    <i class="fas fa-chevron-right"></i>
</a>
</div>

```

Додаток А10 – Інформаційна карусель.

```

body{
    background-color: #fdf2e3;
}
#toTop {
    max-width: 150px;
    max-height: 130px;
    border: 1px solid #fdf2e3;
    border-block-style: dotted;
    color: #2d95bd;

```

```
background-color: #fdf2e3;
text-decoration: none;
text-align: center;
padding: 5px;
position: fixed;
bottom: 25px;
right: 35px;
cursor: pointer;
display: none;
}
```

```
.filterGrid{
  display: grid;
  grid-auto-flow: column;
  justify-content: stretch;
  width: 100%;
  border: 1px solid #ddd;
  margin-bottom: 12px;
}
```

```
.cart {
  width: 220px;
  height: 420px;
  border: 3px solid #2d95bd;
  margin: 20px;
  float: left;
  text-align: center;
  align-content: center;
  padding: 10px;
```

```
display: flex;
flex-direction: column;
justify-content: space-between;
}
```

```
.cart .name {
  font-size: 1.7rem;
  font-weight: bold;
  text-transform: uppercase;
  margin-bottom: 10px;
}
```

```
.cart .description {
  font-size: 16px;
  font-family: "Times New Roman", cursive;
  margin: 10px 0;
  align-content: center;
}
```

```
.cartImg{
  display: flex;
  align-items: center;
  justify-content: center;
}
```

```
.cart .add-to-cart {
  font-size: 1.1rem;
  background: black;
  color: white;
  padding: 10px 20px;
  border: none;
```

```
    cursor: pointer;
}
.cart .add-to-cart:hover {
    background: #b0cdf6;
}

.table_statistics {
    border-collapse: collapse;
    text-align: center;
    width: 100%;
    font-size: 13px;
}

.table_statistics td,
.table_statistics th {
    padding: 10px;
}

.table_statistics th {
    font-weight: normal;
    font-size: 15px;
    background: white;
}

.table_statistics tr:nth-child(odd) {
    background: #fdf2e3;
}

.table_statistics td:nth-child(1),
.table_statistics th:nth-child(1) {
```

```
text-align: left;
}

.table_statistics th:hover {
border-bottom: 3px solid black;
cursor: pointer;
background-color: #b0cdf6;
}

.myCart{
width: 180px;
height: 380px;
border: 3px solid #2d95bd;
margin: 20px;
float: left;
text-align: center;
align-content: center;
padding: 10px;

display: flex;
flex-direction: column;
justify-content: space-between;
}

.myCart .name {
font-size: 1rem;
font-weight: bold;
text-transform: uppercase;
}
```

```
.myCart .description {  
    font-size: 10px;  
    font-family: "Times New Roman", cursive;  
    margin: 10px 0;  
    align-content: center;  
}
```

```
.del-goods{  
    font-size: 0.8rem;  
    background: black;  
    color: white;  
    padding: 10px 20px;  
    border: none;  
    cursor: pointer;  
}
```

```
.del-goods:hover {  
    background: #b0cdf6;  
}
```

```
.addTeacherModal {  
    display: none;  
    position: fixed;  
    background-color: white;  
    height: 600px;  
    max-width: 400px;  
    width: 100%;  
    top: 50%;  
    left: 50%;
```

```
right: 0;
bottom: 0;
transform: translate(-50%, -50%);
border: 1px solid black;
z-index: 10;
}

.addTeacherModal:target,
.addGameButton{
  display: block;
  background-color: white;
}

.teacherInfoModal {
  display: none;
  position: fixed;
  background-color: white;
  max-height: 600px;
  max-width: 1000px;
  width: 100%;
  top: 50%;
  left: 50%;
  right: 0;
  transform: translate(-50%, -50%);
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
  z-index: 10;
}

.addTeacherModal{
  display: none;
```

```
position: fixed;
background-color: white;
height: 600px;
max-width: 400px;
width: 100%;
top: 50%;
left: 50%;
right: 0;
bottom: 0;
transform: translate(-50%, -50%);
border: 1px solid black;
z-index: 10;
}
```

```
.formInput {
  width: 100%;
  height: 40px;
  margin-top: 10px;
  margin-bottom: 20px;
}
```

```
.submitButton input {
  width: 100%;
  margin-top: 0.1px;
}
```

```
.formSelectContainer {
  width: 100%;
  display: flex;
  justify-content: space-between;
```

```
}  
  
.teacherInfoModal:target,  
.addTeacherModal:target,  
.addGameButton{  
    display: block;  
}  
.userInfoModal:target {  
    display: block;  
}  
  
.popupContainer {  
    padding-right: 10%;  
    padding-left: 10%;  
    width: 100%;  
    height: 100%;  
    margin: auto;  
}  
  
.photoAndInfo {  
    display: flex;  
    max-height: 350px;  
    width: 100%;  
}  
  
.photoAndInfo .descr {  
    width: 100%;  
    flex-shrink: 1;  
    padding-left: 10px;  
    margin-left: 50px;
```

```
}
```

```
#close_add {  
  color: #2d95bd;  
  float: right;  
  font-size: 25px;  
  position: absolute;  
  right: 10px;  
  top: 10px;  
}
```

```
#close_add:hover {  
  color: #000;  
  text-decoration: none;  
  cursor: pointer;  
}
```

```
.userInfoModal {  
  display: none;  
  position: fixed;  
  background-color: white;  
  max-height: 900px;  
  max-width: 920px;  
  width: 100%;  
  top: 50%;  
  left: 50%;  
  right: 0;  
  transform: translate(-50%, -50%);  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);  
  z-index: 9999;
```

```
}
```

```
.block {  
  display: inline-block;  
  vertical-align: top;  
  cursor: pointer;  
  box-shadow: 1px 1px 5px #adbdb7;  
  margin: 10px;  
}
```

```
.icon-block {  
  position: relative;  
  line-height: 0;  
}
```

```
.icon-block * {  
  line-height: normal;  
}
```

```
.name {  
  text-align: center;  
  font-size: 18px;  
  padding: 10px 0;  
  transition: 0.3s linear;  
}
```

```
.modal {  
  display: none;  
  position: fixed;  
  z-index: 1;
```

```
padding-top: 100px;
left: 0;
top: 0;
width: 100%;
height: 100%;
overflow: auto;
background-color: rgb(0,0,0);
background-color: #adbdb7;
}
```

```
.modal-content {
background-color: #fefefe;
margin: auto;
padding: 20px;
border: 1px solid #fdf2e3;
width: 80%;
text-align: justify;
display: flex;
flex-direction: row;
justify-content: space-around;
align-items: center;
}
```

```
.modal-content img{
width: 100%;
height: 100%;
object-fit: cover;
max-height: 170px;
border-radius: 10px;
max-width: 400px;
}
```

```
.modalOpenedCard{  
  display: flex;  
  flex-direction: column;  
  justify-content: space-evenly;  
  margin: 30px;  
}
```

```
.close {  
  color: #aaaaaa;  
  float: right;  
  font-size: 28px;  
  font-weight: bold;  
}
```

```
.close:hover,  
.close:focus {  
  color: #000;  
  text-decoration: none;  
  cursor: pointer;  
}
```

```
.goods-out{  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  justify-content: center;  
  align-content: center;  
}
```

```
.cartImg img{  
  width: 100%;
```

```
height: 100%;  
object-fit: cover;  
max-height: 170px;  
border-radius: 10px;  
}
```

```
.blockBtns{  
  display: flex;  
  flex-direction: column;  
}
```

```
.personalInfoClass{  
  display: flex;  
}
```

```
#myInfo{  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  margin: 25px;  
}
```

```
.myGames_grid{  
  display: flex;  
  flex-direction: row;  
  align-items: flex-start;  
  justify-content: space-evenly;  
}
```

```
.namesMyInfo{  
  display: flex;
```

```
flex-direction: column;  
}
```

Додаток А11 – Стилiзацiя.