

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
«ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ»
Кафедра інформаційних систем в економіці**

галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки та інформаційні технології»

Форма навчання: заочна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ДИПЛОМНИЙ ПРОЄКТ
на тему

**ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ І АНАЛІЗУ ВАКАНСІЙ ДЛЯ
ПРАЦЕВЛАШТУВАННЯ**

студента Олещенка Іллі Дмитровича _____

Науковий керівник: к.е.н., доцент.

_____ Держук О.В.

**Кваліфікаційний бакалаврський
проект допущений до захисту в
Екзаменаційній комісії з атестації
здобувачів вищої освіти
В.о.завідувача кафедри:
к.е.н., доцент.**

_____ Тішков Б.О.

Київ 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА»
навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки та інформаційні технології»
заочна форма навчання

Затверджую:

в.о. Завідувача кафедри

_____ Тішков Б.О.

“ ” 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційний бакалаврський дипломний проєкт
на тему: «Інформаційна система пошуку і аналізу вакансій для
працевлаштування»

Олещенка Іллі Дмитровича

Тему кваліфікаційного бакалаврського дипломного проєкту затверджено наказом ректора від «23» березня 2022 р. № 421-ст.

Випускний кваліфікаційний бакалаврський проєкту виконується на
матеріалах
web-сервісу «Labor»

План випускного бакалаврського проєкту

Розділ 1	Характеристика та аналіз предметної галузі (назва розділу)
Розділ 2	Розробка вимог і моделювання інформаційної системи (назва розділу)
Розділ 3 системи	Проектування та реалізація компонентів інформаційної (назва розділу)

Об'єкт дослідження:	Об'єктом дослідження є процеси, пов'язані з пошуком та аналізом вакансій працевлаштування.
Предмет дослідження:	Предметом дослідження є комп'ютерні технології, за допомогою яких здійснюється пошук вакансій працевлаштування.
Мета випускного кваліфікаційного бакалаврського проєкту:	Мета кваліфікаційного бакалаврського дипломного проєкту є розробка та впровадження системи пошуку вакансій на основі веб-сайту.

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі 1 необхідно надати характерні ознаки предметної галузі, оцінити ризики та конкурентоспроможність, охарактеризувати та навести параметри об'єкта дослідження, що потрібні для побудови веб-сайту.

У розділі 2 необхідно проаналізувати усі вимоги до інформаційної системи такого типу, Обґрунтувати та зпроєктувати алгоритм розв'язання поставленої задачі. Засобами моделювання розробити прототип інформаційної системи пошуку вакансій.

У розділі 3 за допомогою програмного коду, баз даних та іншими технічними засобами провести проєктування модулів ситеми, реалізувати увесь основний функціонал програмних модулів і виконати реалізацію на практиці.

Завдання підготував
науковий керівник

(підпис)

О.В. Держук
(ініціали, прізвище)
«10» березня 2022р.

Завдання одержав студент

(підпис)

І.Д. Олещенко
(ініціали, прізвище)
«10» березня 2022р.

АНОТАЦІЯ

кваліфікаційного бакалаврського дипломного проекту студента 4 курсу
Навчально-наукового інституту «Інститут інформаційних технологій в
економіці»

Олещенка Іллі Дмитровича

виконаного на тему:

«Розробка веб-сайту з пошуку вакансій»

Ідеєю розроблюємого проект є розроблення автоматичної і інформаційної системи пошуку вакансій та подальша її інтеграція до веб сервісу, засобами сучасних веб технологій.

Об'єктом дослідження є процес пошуку вакансій людям, що прагнуть знайти роботу.

Кваліфікаційний бакалаврський дипломний проект містить в собі три розділи, які структуровано та логічно пов'язані.

Перший розділ містить характеристику предметної області й об'єкта дослідження, а також перелік існуючих програмних засобів для вирішення даної задачі.

Аналіз вимог до системи та аспекти її проєктування наведено у другому розділі. А також наведено алгоритм розв'язання задачі та наведено діаграми вимог до системи.

Третій розділ є конструктивним. Нижче наводиться інформація, технічне програмне забезпечення та організаційна підтримка підсистеми прийняття рішень для роботи веб-додатку. Враховані архітектури масивів інформації: розроблена структура носія інформації, форми вхідних та вихідних документів описані та наведені в додатку. Було підтверджено набір технічних засобів, а також програмне забезпечення, що використовується для створення системи організації інтернет торгівлі. Вказано структури інформаційної матриці, яка використовується для розв'язання задачі.

Ключові слова: веб-сайт, аналіз ризику, система, інформаційна система.

ANNOTATION

qualification bachelor's degree project of a 4th year student
Educational and Scientific Institute "Information Technology in Economics"
Oleshenko Iliya Dmitrovich
performed on the topic:
" Development of a website for a search for vacancies "

The idea of the developed project is to develop an automatic and information system for finding vacancies and its further integration into the web service, using modern web technologies.

The object of research is the process of finding vacancies for people looking for a job.

The qualification bachelor's thesis project consists of three sections, which are structured and logically connected.

The first section contains a description of the subject area and object of study, as well as a list of existing software to solve this problem.

The analysis of system requirements and aspects of its design is given in the second section. There is also an algorithm for solving the problem and diagrams of system requirements.

The third section is constructive. The following is information, technical software, and organizational support for the web application decision-making subsystem. The architectures of the arrays of information are taken into account: the developed structure of the information carrier, the forms of input and output documents are described and given in the appendix. A set of hardware and software used to create an e-commerce system was confirmed. The structures of the information matrix used to solve the problem are indicated.

Keywords: website, risk analysis, system, information system.

РЕФЕРАТ

Кваліфікаційний бакалаврський дипломний проєкт містить 94 сторінок, 6 таблиць, 24 рисунка, список літератури з 25 найменувань, 1 додаток.

РОЗРОБКА ВЕБ-САЙТУ З ПОШУКУ ВАКАНСІЙ

Мета кваліфікаційного бакалаврського дипломного проєкту є розробка та впровадження системи с пошуку вакансій працевлаштування на основі веб-сайту.

Об'єктом дослідження є процеси, пов'язані з пошуком та вибором найбільш підходящих для клієнта вакансій працевлаштування.

Предметом дослідження є комп'ютерні технології, принципи проєктування та організації пошуку вакансій.

Результати досягнуті під час проєктування – було розроблено інформаційну систему інформаційну систему для програми «Labor» на основі веб-сайту, яка буде ефективною як для впровадження власного бізнесу в сфері веб-проєктів, так і для клієнтів, що будуть користуватися веб-сайтом для своїх потреб за допомогою оптимізації окремих компонентів і правильного підходу до побудови окремих модулів системи, зручним та зрозумілим інтерфейсом для користувачів, актуальною інформацією та тестуванням, все це доведе, що система працездатна та задовольняє поставлені вимоги.

Результат розробки буде яскравим прикладом для підприємців, котрі бажають розпочати власний бізнес та вивести свою справу у серйозну компанію, що матиме великий потенціал та буде користуватися попитом, а також може виступити у вигляді паттерну для проєктування інформаційних систем на власній базі.

Перелік ключових слів: інтернет додаток, система, інформаційна система, торгівля.

Ключові слова: веб-сайт, аналіз ризику, система, інформаційна система.

Рік виконання і захисту дипломного проєкт – 2022.

ЗМІСТ

АНОТАЦІЯ	4
РЕФЕРАТ	6
ВСТУП	8
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	11
1.1. Характеристика предметної галузі та об'єкта дослідження	11
Сучасні технології допомагають пошукачам отримувати найактуальніші пропозиції щодо роботи без витрат часу на тривалі пошуки.	13
1.2. Аналіз літературних джерел та практичного досвіду використання ІС...	13
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	16
2.1. Аналіз і специфікація вимог до інформаційної системи	16
2.2. Постановка та алгоритм розв'язання задачі	21
2.2.1 Постановка задачі.....	21
2.2.2. Алгоритм розв'язання задачі	28
2.3. Моделювання інформаційної системи.....	30
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ ПІДСИСТЕМИ	36
3.1. Інформаційне забезпечення	36
3.2. Технічне забезпечення.....	40
3.3. Програмне забезпечення	43
3.4. Результати реалізації інформаційної системи/підсистеми	45
ВИСНОВКИ	48
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ.....	53

ВСТУП

Сьогодні майже всі компанії мають власний веб-сайт. Зараз, в умовах використання сучасних інформаційних технологій це необхідний фактор існування кожної сфери діяльності, що допомагає людству у вирішенні багатьох питань, оскільки має певну зручність у використанні і тим самим залучити додаткових клієнтів.

Пошук роботи зараз має багато шляхів вирішення:

- пошук роботи через знайомих та родичів;
- об'яви про вакансії в газетах;
- власна публікація в газету;
- кадрові агентства (ПЗ);
- державна служба зайнятості України;
- Пошук роботи через мережу Internet;

Масове прийняття інтернету створило приклад того, як сьогодні працюють компанії. За останні десять років з'явився новий спосіб пошуку роботи: електронна система, де збираються як роботодавці, що шукають гідних претендентів з великої кількості користувачів і тим самим розвивають свій бізнес, так і люди що потребують знайти собі гідну вакансію що підходила б їм у їхніх критеріях чи своїх поглядів на життя або ж певні персональні обставини. Традиційний пошук роботи у газетах чи знайомих стає дедалі непопулярним, тому багато компаній рухається по всьому світу завдяки електронним системам, тому що вони мають багато переваг і значно розширюють можливості як звичайних користувачів веб-сайтом так і роботодавців своєю зручністю.

Темою проєкт став веб-сайт, як спосіб реалізації пошуку вакансій працевлаштування та пошуку робітників, так як актуальність цієї теми в даний час досить велика, завданням проєкт є полегшення пошуку вакансій та робітників за допомогою сучасного веб-сайту.

Предметом завдання є сукупність теоретичний, методичних і прикладних аспектів розробки веб-додатку з пошуку вакансій.

Об'єкт дослідження – процеси забезпечення розробки програмного забезпечення з електронного пошуку вакансій та його забезпечення через технічні і людські ресурси.

У цій роботі буде спроектована і розроблена автоматизована інформаційна система на основі веб-сайту для зручного пошуку вакансій людьми.

Завдання:

- описати предметну область;
- сформулювати мету створення данної системи (ПЗ);
- писати призначення данної системи (ПЗ);
- визначити хто майбутні споживачі/користувачі системи (ПЗ);
- описати сфери застосування майбутньої системи (ПЗ);
- перерахуйте які проблеми вирішить система (ПЗ),
- описати бізнес- вимоги, та представити діаграму;
- описати функціональні вимоги та не функціональні вимоги, та представити діаграму;
- визначити можливі ризики для проєкт і системи (ПЗ);
- визначити та описати об'єкти системи (ПЗ) представити діаграму прецедентів;
- розробити власну інформаційну систему для реалізації електронної комерції
- розробити програмне забезпечення під підтримки електронної комерції
- визначити технічне забезпечення

Найголовніше для компаній сьогодні – це надання розуміння користувачам, які цілі переслідує та чи інша компанія, можливість якнайшвидше, за вимоги, або якнайкраще підібрати саме те, що ідеально підходить користувачу. Використання веб-сайтів настільки широко поширене, оскільки дозволяє виконувати ці завдання. Діджиталізація працює і робить процес пошуку потрібних вакансій максимально плавним та ефективним. Всесвітня павутина дозволяє зберігати багато часу і давати користувачу свободу вибору, підібрати найбільш влучне. Система також є

ефективною та зручною для компаній оскільки допомагає компаніям набагато ефективніше та швидше знаходити робітників, також має свободу вибору для наймання кращих кандидатів. Інтернет-функціонал значно підвищує ефективність роботи системи.

Актуальність створення автоматизованої системи для сайту пошуку роботи пов'язана з:

- створення зручних умови для завантаження інформації про користувачів
- створення зручних способів додавання вакансій і перегляду резюме студентів з використанням різних фільтрів
- можливість перегляду вакансій користувачами
- переглядати список кандидатів, обирати працівника, що буде відповідати вашим запитам

Завдання мають деталізувати мету роботи:

Відповідно до об'єкта, теми та індивідуального завдання, метою даної роботи є полегшення процесу пошуку роботи для звичайних користувачів, розробка зручного у користуванні сервісу.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Характеристика предметної галузі та об'єкта дослідження

Пошук роботи через мережу Інтернет – найпопулярніший та доступний метод пошуку роботи. Тому що всесвітня мережа є найдемократичнішим засобом спілкування роботодавців із потенційними працівниками. Тут кожен може обрати для себе найбільш оптимальний час та місце роботи, а також пристойну винагороду за власну діяльність.

Сайти з роботи бувають як загальними, так і спеціалізованими. Вакансії всередині певної галузі, вакансії для молодих фахівців тощо. На деяких сайтах з пошуку роботи координати роботодавців є загальнодоступними. Однак на більшості job-сайтів такої можливості немає. Тому претенденти на займану посаду можуть лише відгукнутися на вакансії за допомогою розміщеного на сайті резюме.

Є й нестандартні методи пошуку роботи через мережу Інтернет. Наприклад, форуми з роботи. На таких форумах роботодавці розміщують свої вакансії. А шукачі пропонують свої послуги. Перевагою є те, що на них можуть публікуватися вакансії, яких немає в інших джерелах. Вакансії можна зустріти й на сайтах органів з праці та зайнятості. В інших джерелах ця інформація може не публікуватись.

Сайти кадрових агентств. Найчастіше агентства дублюють свої вакансії в Інтернеті та друкованих виданнях. Але не завжди! Тому ці сайти можуть також виявитись цінним джерелом інформації. Часто пропозиції роботи розміщуються на сайтах компаній. Як правило, на них набагато менше відвідувачів, ніж на сайтах з роботи. Тому, якщо вакансія не дублювалася в інших джерелах і не застаріла, у Вас великі шанси якщо не працевлаштуватися, то, як мінімум, бути поміченим.

Соціальні мережі. Вони бувають як загальними, так і спеціальними. Можна вступити в групу, присвячену пошуку роботи. Можна спробувати вийти прямо на співробітників служб персоналу компаній, які Вас цікавлять. Можна написати про те, що Ви шукаєте роботу. Можна поставити статус «Шукаю роботу».

Серед переваг пошуку роботи в мережі Інтернет:

- 1) На сайтах з працевлаштування великий вибір вакансій.
- 2) Створені умови для швидкого пошуку необхідної інформації.
- 3) Компанія є більш відкритою для користувачів.
- 4) За кількістю вакансій Інтернет значно перекидає паперову пресу.
- 5) Зручність. Проглядати вакансії, відправляти резюме можна у зручний для себе час вдома. Спілкуватися з роботодавцем можна у режимі онлайн.
- 6) Швидкість підбору роботи. Пропозицій щодо працевлаштування на сайтах дуже багато. Вони різноманітні. Тому тут можна швидко знайти постійну роботу або підробіток.
- 7) Актуальний перелік вакансій.
- 8) Компанія є більш відкритою для споживачів.

Для країни та суспільства:

- 1) Надавати людям широкий спектр роботи;
- 2) Створити нові робочі місця;
- 3) Національний економічний розвиток, розвиток науки і техніки та підвищення рівня життя.

Серед недоліків:

- 1) Необхідність регулярного доступу до мережі Інтернет.
- 2) посилення конкуренції;
- 3) Основні інвестиції в постійне оновлення технологій
- 4) Недостовірність відомостей;
- 5) Завдяки відкритості інформації існує інформаційна небезпека, шахрайство;
- 6) Потрібні високоякісні працівники (розробники);
- 7) Деякі сайти з пошуку заповнені сумнівними оголошеннями від фірм, які займаються мережевим маркетингом.
- 8) Роботодавці часто не надають достатньої інформації про фірму, посаду, яку пропонують та оплату праці.

Сучасні технології допомагають пошукачам отримувати найактуальніші пропозиції щодо роботи без витрат часу на тривалі пошуки.

1.2. Аналіз літературних джерел та практичного досвіду використання ІС

Вибрати оптимальну систему управління контентом для створення Інтернет додатку - непросте завдання. Ринок пропонує десятки 20 різноманітних рішень, що відрізняються як за функціоналом, так і за вартістю. Загальні вимоги до CMS Перед тим як перейти до огляду рішень необхідно визначити ряд критеріїв, яким повинна відповідати CMS (Content Management System) організації Інтернет додатку. З підбором вакансій впорається будь-яке рішення, але крім красивого переліку завжди потрібно ряд функцій, без яких уявити сучасний інтернет додаток нереально.

Результати багатьох опитувань, що проводились порталом «CMS Magazine», зайвий раз підтверджують - більшість користувачів не люблять проходити болісний процес реєстрації. Особливо це правило відноситься до так званих одноразових сайтах. Наприклад, коли сайт пропонує передивитись вакансію, то користувачу може знадобитися один, максимум два набори. Отже, непотрібно змушувати користувача витрачати сили на створення облікового запису. Адже він найближчим часом на сайт повернется навряд чи. Сучасні CMS повинні передбачати подібні сценарії і допускати пошуки без реєстрації. Безпека, обрана для створення інтернет додатку CMS може не володіти багатим функціоналом, але відповідати елементарним вимогам до безпеки вона повинна.

Немає нічого гіршого, коли персональні дані користувачів викрадаються через ваш веб-сайт. Найпростіший спосіб визначити безпечність певного продукту - співвіднести коефіцієнт популярності з інтенсивністю знаходження критичних вразливостей. Чим частіше їх знаходять, тим зайвий раз варто задуматися про вибір такого рішення. Популярні рішення по статистиці безпечніші, тому що компанії-розробники в основному приділяють належну увагу тестуванню коду. Продуктивність більшість сучасних рішень для створення інтернет додатків,

написані на популярному мові програмування PHP. Його підтримує будь-який 21 віртуальний хостинг.

Не всі CMS готові пропагувати нову інтеграцію з 1С, тому їх замінює Varto. Важливо пам'ятати, що ви можете розробити аналогічний модуль для існуючої CMS - дорого, і іноді простіше вибрати інше рішення, менш заморочене розробкою. На ринку електронної комерції існує безліч рішень для автоматизації діяльності компаній із багатої сфери. Давайте подивимося на гру програми. "DIAFAN.CMS" - універсальна система управління контентом. Система вдосконалюється на основі створення веб-сайтів, будь то розбірні та віддалені, з практичної точки зору. В "DIAFAN.CMS" ви можете створити: 22 - сайти компаній; - Інтернет магазини; - Інформаційні портали. Система може мати гарні функціональні можливості, які можна використовувати для запуску інтернет-доповнень у різних напрямках. Майже всі елементи системи можна налаштовувати з панелі керування. Візьми, якщо хочеш, таку форму молитви. Легко заповнити потрібні поля, щоб заповнити протягом години. На складі тип можна легко змінити з панелі керування. Інтернет-доповнення може охоплювати все необхідне для онлайн-опитування: інтеграцію з 1С, бази даних, додаткові функції, імпорт ботів та багато іншого. Доступніше звертатися до колег та робіт для спеціальних обговорень, обміну грошей. Вносити зміни до конфігурації 1С не обов'язково. Обмін реалізаціями на базі 1С-подібного формату – SearchML. Керування процесом обміну пива в "DIAFAN.CMS" інших налаштувань на панелі керування немає.

Якщо потрібно змінити логіку або процес формування найменувань, то доведеться лізти в код модуля. Розробники уважно ставляться до нетипових функцій. До таких можна віднести відображення банерів. Невеликі інтернет-додатки практикують обмін банерами і «DIAFAN.CMS» пропонує все необхідне для автоматизації цього процесу. Для кожного банера доступні налаштування, що дозволяють гнучко управляти показами. У «DIAFAN.CMS» є можливість розділяти права керування сайтом.

Якщо у вас більше прав доступу до сторінки, ви можете призначити секретарю привілею, наприклад, лише для публікації вакансій, статистичної інформації чи

читання інтерв'ю. Панель керування Mova може бути російською, українською чи англійською мовами. Перенесення загальнодоступної частини сайту може бути безкоштовним. "DIAFAN.CMS" підходить для створення найпопулярніших доповнень в Інтернеті. Проект знаходиться у активній фазі розробки. Незначні випуски регулярно випускаються та виправляються для інших незначних проблем. "ДІАФАН.CMS" 23 поставляється єдиним релізом, що включає всі модулі за одну ліцензію 3000 грн. "ImageCMS" - платна система управління контентом (Professional - 8 000 грн, Premium - 13 500 грн). Перемога за рахунок створення типового дизайну корпоративного веб-сайту та багатофункціональних інтернет-доповнень. Написаний на PHP, з фреймворком CodeIgniter та інтеграцією з базою даних MySQL.

Для створення інтернет-додатків система представлена у двох версіях: «ImageCMS Web Pro» та «ImageCMS Web Premium». Платформа задовольняє потреби роботодавців великою кількістю вакансій (1000 одиниць). Каталоги продуктів, складені на «Image.CMS», підтримують більшість типових функцій: сортування, фільтрація за ознаками, аjax-пошук тощо. Вибрані вакансії також підтримують технологію аjax, тому користувач не зможе перезавантажити сторінку. Однією з основних проблем цього рішення є продуктивність і помилки. Розробники намагаються виправити їх якомога швидше, але з кожним випуском є скарги. Вимоги до продуктивності системи не малі. Підтримка інтеграції з 1C: Enterprise Edition ще не завершена. Web-Script — популярна комерційна CMS з відкритим кодом, призначена для створення Інтернет-додатків. Написано мовою програмування PHP з використанням фреймворку Webasyst.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Аналіз і специфікація вимог до інформаційної системи

Аналіз вимог є критичним для успішної розробки проєкту. Вимоги мають бути задокументовані, піддані вимірюванню, перевірені на відповідність вимогам бізнесу та описані достатньо детально для проєктування системи. На етапі аналізу вимог і планування користувачі АІС спочатку визначають функції, які вона повинна виконувати, описують вимоги до інформації. Формулювання вимог АІС в основному здійснюється користувачами під керівництвом експертів з розробки. Обмежений масштаб проєкту АІС встановлює часові рамки для кожної з наступних фаз. Крім того, було визначено можливість реалізації проєкту на зазначеному обсязі фінансування, наявному обладнанні тощо. Результатом цього етапу має стати пріоритетний функціональний список для майбутнього АІС, а також попередня функціональна модель АІС.

Що дасть виявлення вимог:

- узгоджений перелік бізнес-вимог, вимог зацікавлених осіб і вимог до рішення;
- можливість визначити пріоритети, за якими система буде розроблятися і впроваджуватися;
- базу для оцінки трудомісткості і загальних витрат по проєкті.

Що дасть розробка концептуальних документів:

- високорівневий опис рішення відповідно до загальновизнаних стандартів розробки бізнес-аналітичних документів;
- розуміння зацікавленими особами кордонів рішення (що рішення має забезпечувати і що знаходиться поза рамками рішення);
- можливість планування робіт і виконання подальших дій з розробки рішення.

Що дасть створення специфікацій вимог:

- закінчений опис поведінки інформаційної системи, яку потрібно розробити;

- наявність повністю сформульованих функціональних і нефункціональних вимог до ПЗ;
- можливість точної оцінки етапів і планів, термінів проект;
- можливість початку робіт з розробки програмного забезпечення.

На етапі проектування частина користувачів бере участь в технічному проектуванні системи під керівництвом фахівців-розробників. Інструмент CASE використовується для швидкого створення прототипу робочої програми. Користувачі безпосередньо взаємодіють з ними, уточнюючи та доповнюючи потреби системи, які не були визначені на попередньому етапі. Більш детально розглянуто процес роботи системи. За потреби аналізуйте та коригуйте функціональні моделі. Кожен процес розглядається детально. Визначає вимоги для розмежування доступу до даних. На цьому ж етапі визначаються необхідні документи. Результатами цього етапу повинні бути: загальна інформаційна модель системи; функціональна модель всієї системи та підсистем, реалізована кожною командою розробників; інтерфейси, незалежно визначені інструментами CASE між самостійно розробленими підсистемами; побудова екранів , повідомляє Прототип діалогу. Усі моделі та прототипи необхідно отримати з використанням тих інструментів CASE, які будуть використовуватися для побудови системи в майбутньому. Дане вимога викликана тим, що в традиційному підході при передачі інформації про проект з етапу на етап нерідко відбувається неконтрольоване спотворення даних. Застосування єдиної середовища зберігання даних про проект дозволяє цього уникнути. На відміну від традиційних підходів, при яких використовуються специфічні засоби, не призначені для побудови реальних додатків, а прототипи викидаються після усунення неясностей у проекті АІС, в підході RAD кожен прототип передається майбутній системі. Таким чином, на наступну фазу передається більш повна і корисна інформація.

На етапі побудови здійснюється безпосередньо сама швидка підготовка програми. При цьому розробники виконують ітеративное побудова реальної АСУ на основі отриманих в попередній фазі моделей, а також вимог нефункціонального характеру. Програмний код частково формується CASE-засобами автоматично.

Кінцеві користувачі на цій фазі оцінюють одержувані результати і вносять корективи, якщо в процесі розробки система перестає задовольняти вказаним вимогам. Тестування автоматизованої системи здійснюється в процесі розробки. Після закінчення робіт кожної окремої команди розробників проводиться поступова інтеграція даної частини системи з іншими, формується повний програмний код, виконується тестування спільної роботи даної частини програми, а потім тестування АІС в цілому. Завершується фізичне проєктування АІС, що включає: визначення необхідності розподілу даних; аналіз використання даних; фізичне проєктування бази даних; визначення вимог до апаратних ресурсів і способів збільшення продуктивності, завершення розробки документації проєкт. Результатом даного етапу є готова автоматизована система, яка задовольнить всіх узгоджених вимог

Бізнес-вимоги до системи:

Інформаційна система управління підприємством повинна забезпечувати: - Цілісність інформації в усіх аспектах системи управління. Цілісність визначається як відношення отриманої інформації до інформації, яка вимагається або запитується керівництвом. Оскільки наші знання відносні, неможливо досягти 100% повноти інформації. Крім того, слід мати на увазі, що бажання покращити цілісність інформації може призвести до збільшення адміністративних витрат та зниження її ефективності;

- Корисність і цінність інформації. Як згадувалося раніше, дані є цінними для керівників лише тоді, коли інформація використовується для прийняття адміністративних рішень. Тому інформаційний потік в системі управління має направлятися за конкретними адресами, тобто конкретним керівникам, спеціалістам і працівникам керівників;

- Точність і достовірність інформації. Прийняття рішень на основі неточних або недостовірних даних збільшує ризик помилитися та прийняти неправильні рішення;

- своєчасність надходження інформації. Якщо інформація не надходить вчасно, то орган управління не діятиме як раз в той момент, коли об'єкт управління має особливу потребу в керуючому впливі;

- агрегуємий інформації. Під агрегуємий розуміється раціональний розподіл інформації за рівнями ієрархії управління. На вищі рівні управління повинна надходити більш узагальнена інформація, на нижні - більш деталізована. Прикладом агрегуємий може служити система оперативного, бухгалтерського і статистичного обліку. Для прийняття рішень на республіканському рівні управління важливі дані статистики, на рівні начальника ділянки підприємства - оперативного обліку;

- актуальність інформації. В умовах, безперервного техніко-технологічного оновлення інформація застаріває все більш швидкими темпами. Тому при прийнятті рішень слід враховувати вік інформації та її актуальність для конкретних управлінських завдань;

- економічність і ефективність обробки інформації. Ефективність інформаційної підсистеми можна оцінити, зіставляючи результати управління з витратами на збір, накопичення, зберігання, обробку, перетворення і передачу інформації.

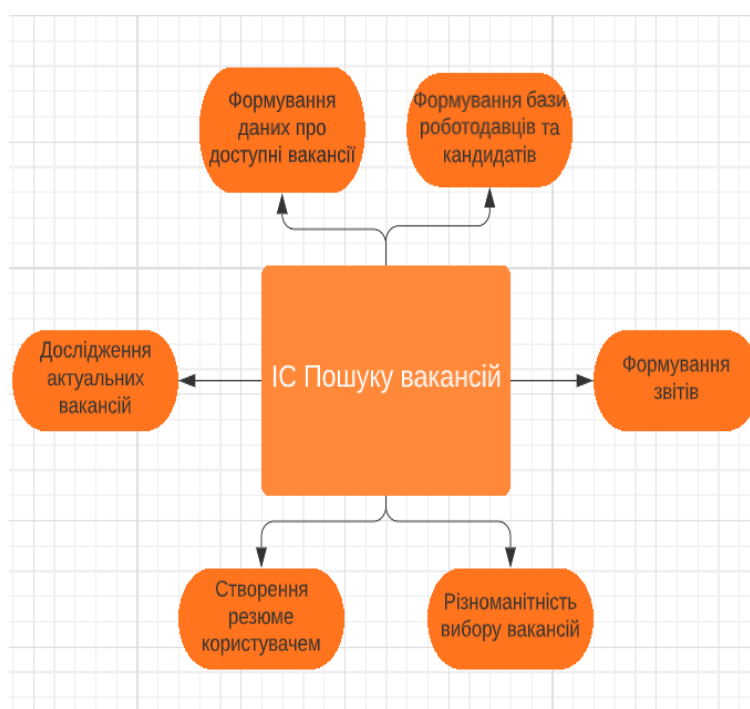


Рисунок 2.1 – Діаграма бізнес-вимог до системи

Функціональні вимоги до системи:

Функціональні вимоги описують внутрішню функціональність системи, її поведінку: обчислення даних, обмін та керування даними, вивчення даних та інших конкретних функцій, які система повинна робити. Як правило, функціональні вимоги визначають, що система повинна робити, а нефункціональні — як система повинна виглядати.

Крім того, автоматизована інформаційна система повинна задовольняти ряд таких технічних вимог, як:

- швидкодія - швидкість при введенні, пошуку, обробки інформації;
- надійний захист від несанкціонованого доступу до даних;
- зручний призначений для користувача інтерфейс робочих місць;
- можливість розвитку системи;
- інтеграція з модулями, використовуваними в системі передачі даних;
- можливість проведення конвертації даних з використовувалися раніше в нову систему;
- висока надійність роботи. Інформаційна система підприємства включає суб'єктів комунікації, канали і носії інформації, а також технічні засоби інформаційної роботи.

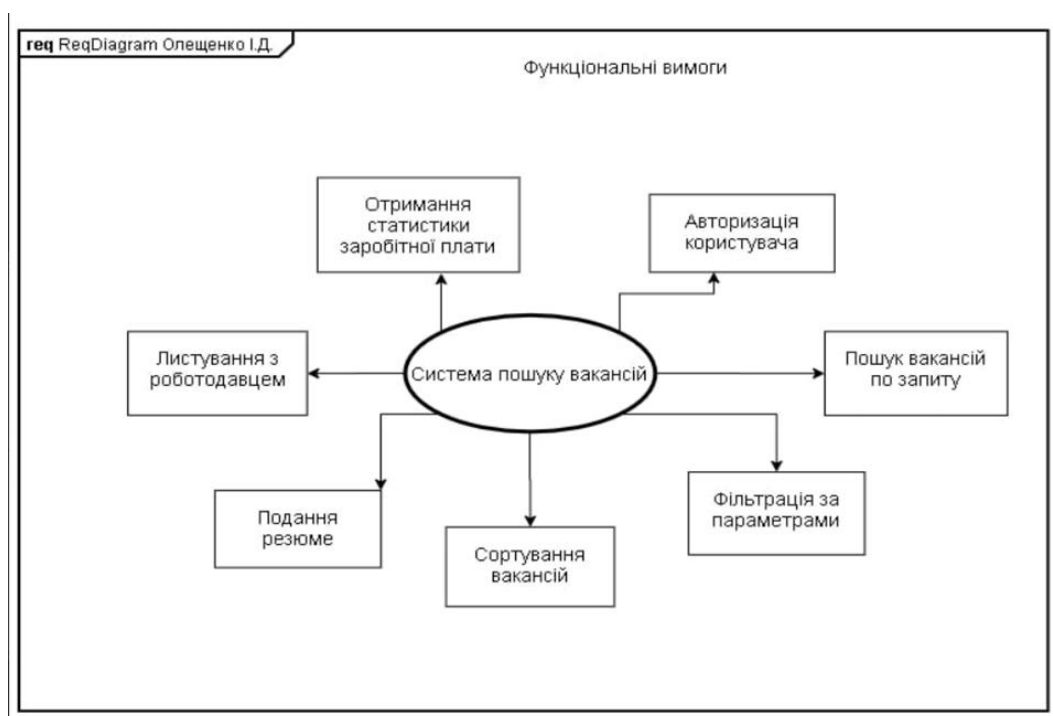


Рисунок 2.2 – Діаграма функціональних вимог до системи

Нефункціональні вимоги до системи:

Нефункціональні вимоги описують не те, що програмне за буде робити, а те, як він буде робити це, наприклад, вимоги до продуктивності, до зовнішнього інтерфейсу, до обмежень дизайну та атрибутів якості. Нефункціональні вимоги важкі для тестування; тому, вони, як правило, оцінюються суб'єктивно.

Вимоги до зовнішнього інтерфейсу: інтерфейс користувача повинен бути зручним та зрозумілим для кожної людини будь-якого віку, можливість налаштування, російська/українська мови.

Система має забезпечити захист введених даних за допомогою авторизації користувача та безпомилкову роботу програми.

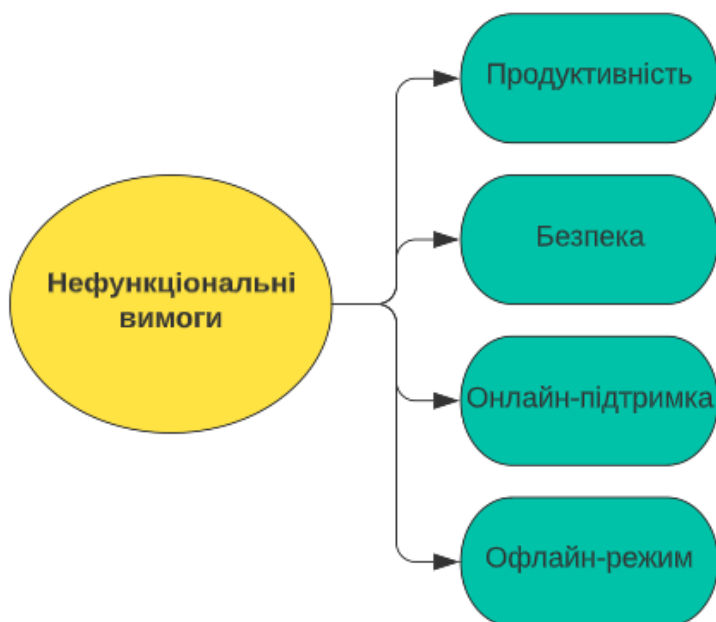


Рисунок 2.3 – Діаграма нефункціональних вимог до системи

2.2. Постановка та алгоритм розв'язання задачі

2.2.1 Постановка задачі

Відповідно до завдання необхідно розробити Інтернет-додаток, призначений для пошуку вакансій працевлущування. Додаток повинен бути інтерактивним, тобто відповідати таким основним вимогам:

1.Більшість сторінок Web-сайту генеруються серверними програмами. Статичні сторінки хоч і присутні, однак складають вкрай незначний відсоток і містять відомості, що не змінюються з плином часу, наприклад, дані про власника Web-сайту або посилальні відомості про використанні Web-сайту.

2.Відомості з метою генерації сторінок, як правило, беруться з баз даних. Баз даних майже постійно функціонують під управлінням сервера і вельми найчастіше - на окремому серверному комп'ютері. Найчастіше також застосовується збереження даних в текстових файлах, однак це об'єктивно, в основному, для фрагментів Web-сторінок (серверні введення).

3.Застосування розмежування доступу до Web-сайту. розмежування доступу передбачає, що різні відвідувачі Web-сайту мають у своєму розпорядженні різноманітними повноваженнями на доступ до тих чи інших даних, опублікованих на Web-сайті. Таким чином, прості користувачі можуть тільки переглядати інформацію, в той час як адміністратор Web-сайту має можливість керуванням сайтом.

4.У разі якщо Web-сайт використовує розмежування доступу, то він обов'язково повинен враховувати можливість входу. Для того щоб увійти на Web-сайт, відвідувач повинен зайти на сторінку авторизації і набрати свій логін і пароль. У разі якщо даний логін і пароль внесені в список відвідувачів Web-сайту, серверна програма дозволяє відвідувачеві увійти на Web-сайт. В Інакше відвідувачеві пропонується зареєструватися на Web-сайті.

5.Web-сайт, який використовує розмежування доступу, зобов'язаний також враховувати можливість реєстрації відвідувача. Як правило на Web-сайтах інтернет-додатків, відвідувач реєструється на ньому, вводячи дані про себе на особливій сторінці реєстрації, після чого вони поміщаються в список відвідувачів Web-сайту.

6.Web-сайт, який використовує розмежування доступу, зобов'язаний крім того враховувати можливість виходу з Web-сайту. При цьому серверна програма «Забуває», то, що цей відвідувач увійшов на Web-сайт, щоб звільнити пам'ять,

відведену для зберігання даних, хоча відповідний запис в списку відвідувачів, звичайно ж, зберігається.

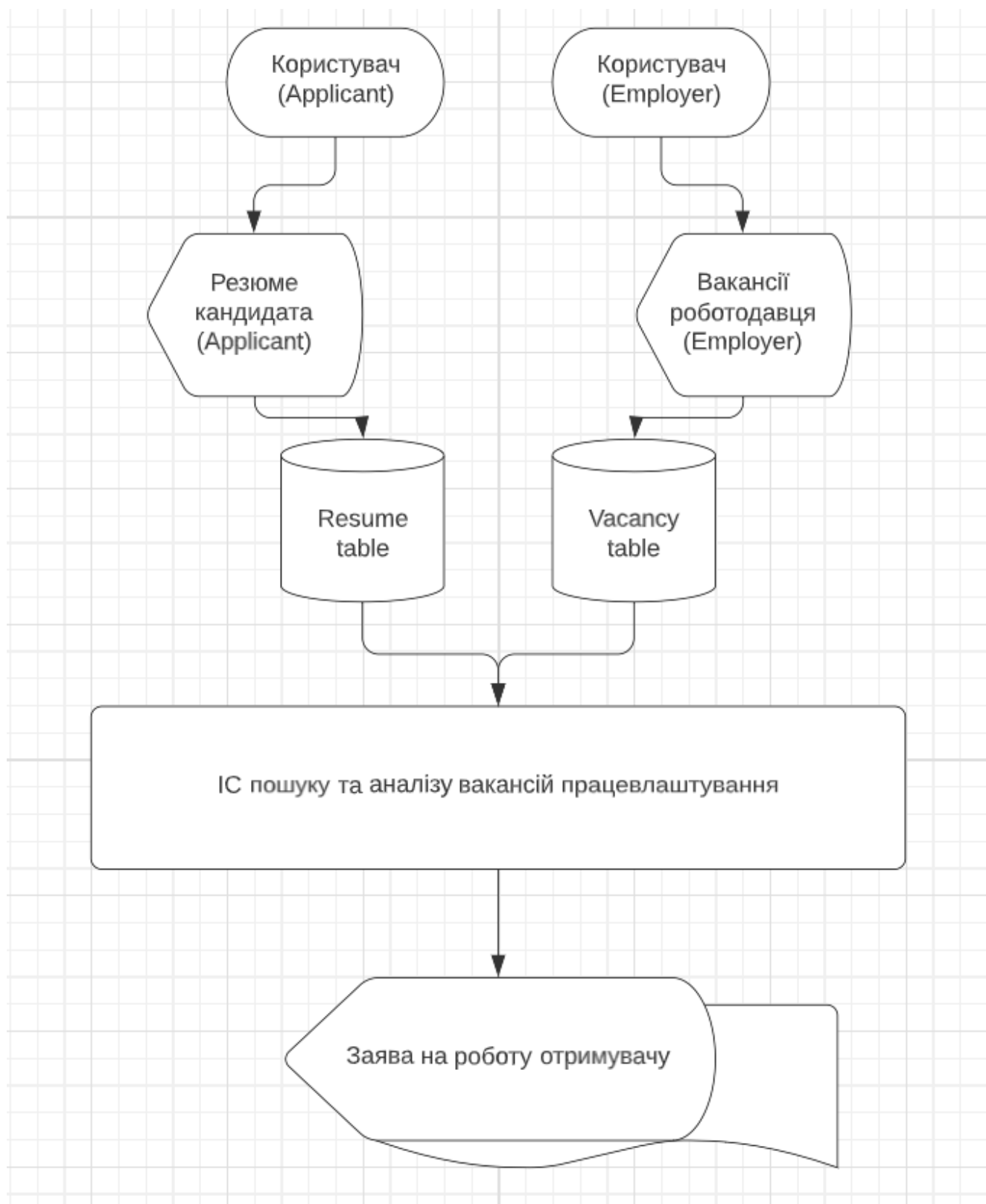


Рис. 2.4 Інформаційна модель задачі

Вихідна та вихідна інформація

Перелік і опис вихідних повідомлень наведений в таблиці 2.1

ПЕРЕЛІК І ОПИС ВИХІДНИХ ПОВІДОМЛЕНЬ

Таблиця 2.1

№ з/п	Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність	Термін видання і допустимий час затримки	Користувачі інформації
1	2	3	4	5	6	7
1	Вакансія	VC	Повинен містити всю необхідну інформацію про вакансію, доступ до контактів (Екранна, документ)	При надходженні	Негайно	Клієнт, власник вакансії
2	Звітність про кількість вакансій що були переглянуті користувачем	SC	Історія перегляду користувачем (Екранна)	Автоматичне оновлення	Негайно	Адміністратор сайту, роботодавець

Перелік і опис вхідних повідомлень наведений в таблиці 2.2 та 2.3

ПЕРЕЛІК І ОПИС ВХІДНИХ ПОВІДОМЛЕНЬ

Таблиця 2.2

Вхідне повідомлення «Вакансія(Vacancy)»

№ з/П	Назва вхідного повідомлення	Ідентифік атор	Форма подання	Джерело
1	2	3	4	6
1	Вакансія	VC	Екранна	Користувач
2	Кількість розміщених (доступних)	CD	Екранна	Розрахунок системи
3	Посилання вакансії	LC	Екранна	Користувач

Продовження таблиці 2.2

4	Спосіб зв'язку с роботодавцем	SR	Екранна	Користувач
5	Дата та час	DT	Екранна	Користувач
6	Інформація про клієнта	CI	Екранна	Користувач

Таблиця 2.3

Вхідне повідомлення «Користувач(Applicant)»

№ з/П	Назва вхідного повідомлення	Ідентифік атор	Форма подання	Джерело
1	2	3	4	6
1	ПІБ	NM	Данні з форми, Екранна	Користувач
2	Номер мобільного	PH	Данні обраних, Екранна	Користувач
3	Адреса	AD	Данні обраних, Екранна	Користувач
4	E-mail	EM	Данні обраних, Екранна	Користувач
5	Дата народження	BD	Данні серверу, Екранна	Користувач

Машинограма звітності пошуків

Таблиця 2.4

ID вакансії	ID клієнта	Вакансії			Дата та час	Інформація по запиту
		Актуальність	Час	Клієнт		
-	-	-	-	-	-	-
		-	-	-		
Запит			_*			
-	-	-	-	-	-	-
		-	-	-		
Запитів за день			_**			

Показники ефективності інтернет додатку

Відвідуваність

Один з найважливіших показників ефективності. Намагайтеся відстежувати статистику відвідувань вашого інтернет-ресурсу. З її допомогою ви зможете побачити сплески і спади відвідуваності сайту, а отже, виявити їх причини та закономірності.

Така статистика - відмінний інструмент для аналізу своїх помилок. Ви можете тестувати різні види залучення клієнтів і при цьому відстежувати які з них найбільш дієві.

Перегляди актуальності вакансій:

Варто відстежувати які саме сторінки і які вакансії набирають найбільшу кількість переглядів. Таким чином можна зрозуміти:

- які вакансії найактуальніші;
- можливо варто змінити щось, якщо її переглядають часто, але рідко реагують на неї;
- до яких пропозицій варто залучати більше уваги.
- сторінки виходу

Аналіз цього пункту допоможе дізнатися, скільки разів відвідувачі залишають ваш інтернет-ресурс. Якщо користувачі виходять з сайту і не формують запити, то це свідчить про те, що:

- процедура зв'язку клієнта с роботодавцем не задовольняє вашого користувача;
- відвідувачі стикаються з проблемою обов'язково залишати свої дані;
- неактуальність даних.

Джерела залучення користувачів:

Важливо знати, з яких каналів просування відвідувачі потрапляють до вас на сайт. Так ви зможете зрозуміти які методи просування найбільш ефективні. Можливо від деяких з варто відмовитися. Зібрані дані допоможуть зрозуміти як краще розділити бюджет на рекламу.

2.2.2. Алгоритм розв'язання задачі

Використовувана інформація.

ПЕРЕЛІК МАСИВІВ ВИКОРИСТОВУВАНОЇ ІНФОРМАЦІЇ

Таблиця 2.5

Масив	Ідентифікатор	Максимальна кількість записів
1	2	3
Клієнти	Clients	1024
Вакансії	Vacancy	1024

Використовувана інформація.

ПЕРЕЛІК МАСИВІВ РЕЗУЛЬТАТНОЇ ІНФОРМАЦІЇ

Таблиця 2.6

Масив	Ідентифікатор	Максимальна кількість записів
1	2	3
Звіт переглянутих вакансій	LookedVC	1024

АЛГОРИТМ РОЗВ'ЯЗАННЯ ЗАДАЧІ

Алгоритм розв'язання задачі подано у вигляді схеми на рис. 2.5.

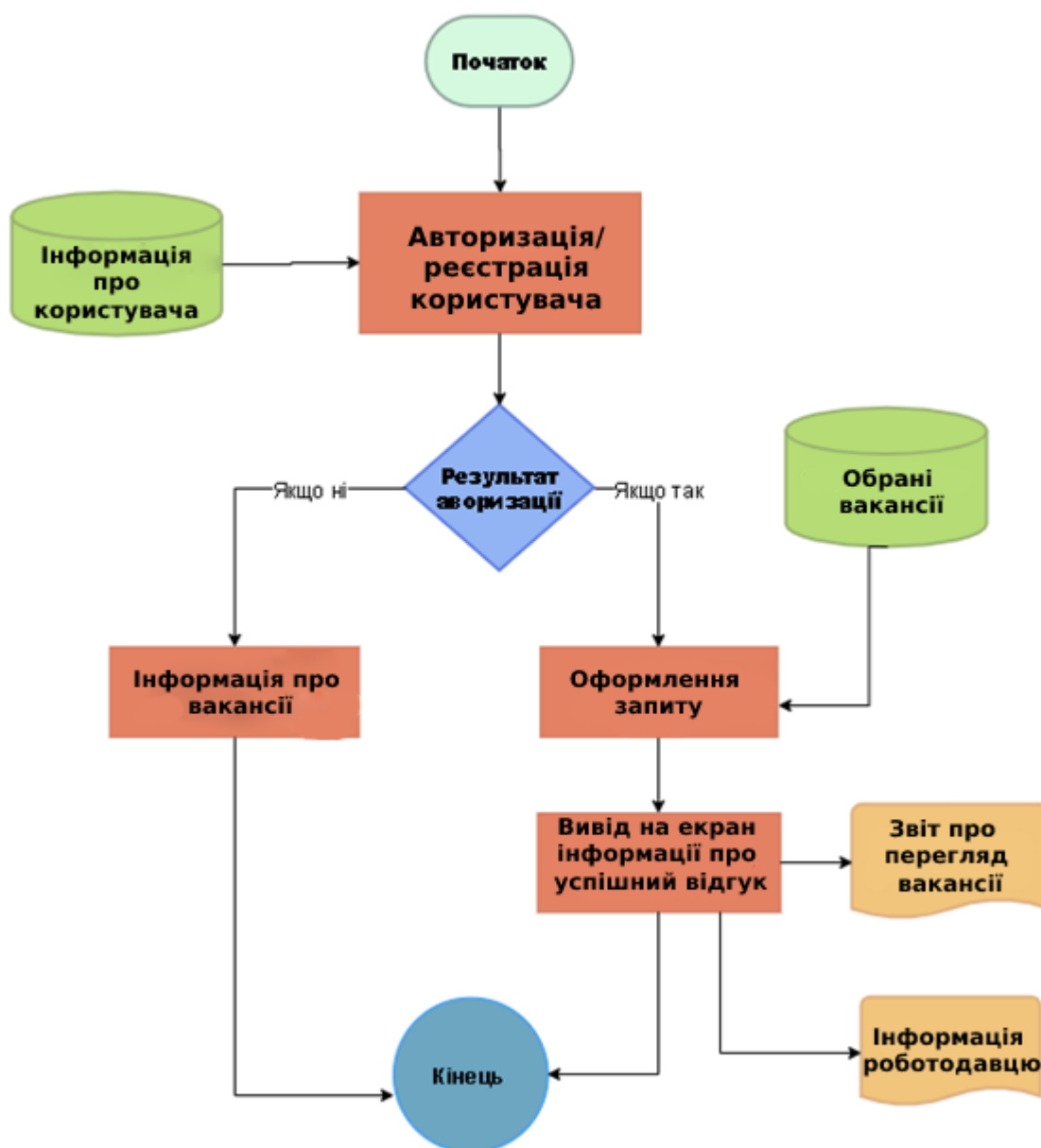


Рис.2.5. Схема алгоритму розв'язання задачі

Алгоритм роботи програмного продукту наведено на рис. 2.6.

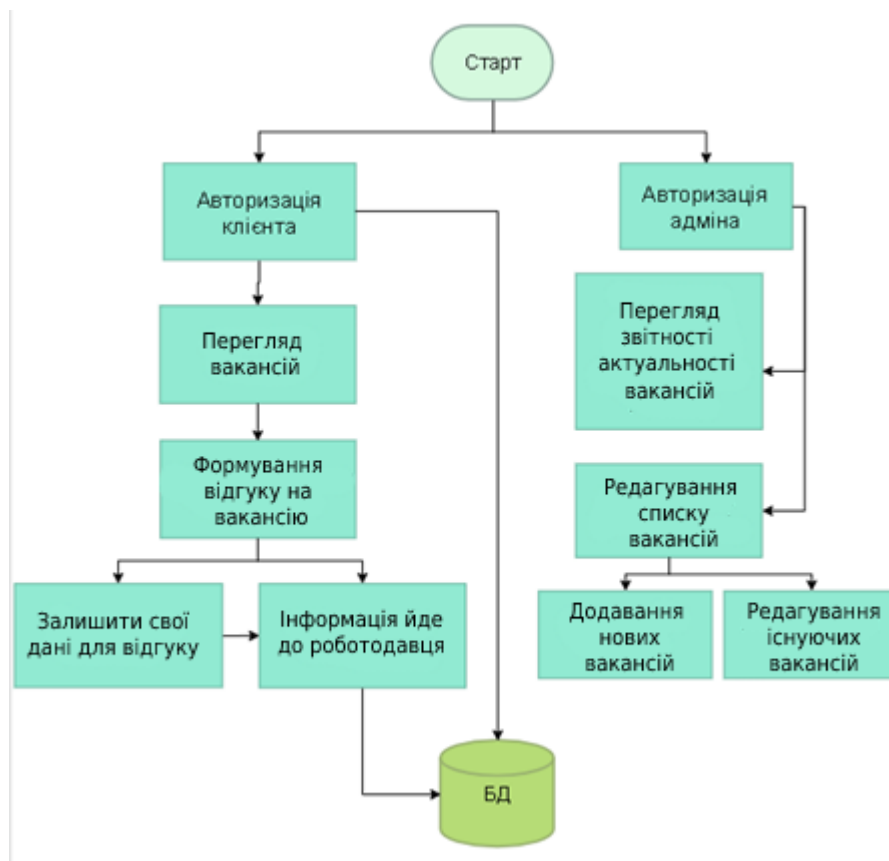


Рис.2.6. Схема алгоритму роботи інтернет додатку

2.3. Моделювання інформаційної системи

Моделювання предметної області

Моделювання предметної області є основою статичної частини моделі UML. Побудова моделі предметної області починається з виявлення абстракцій, що існують в реальному світі, тобто тих основних концептуальних об'єктів, які зустрічаються в системі. При проєктуванні об'єктно-орієнтованого програмного забезпечення ви прагнете структурувати програму так, щоб в центрі виявилися саме ці об'єкти з простору завдання. Це робиться тому, що вимоги до програми змінюються набагато швидше, ніж реальний світ. Основою об'єктного моделювання взагалі і статичного моделювання зокрема якраз і є створення моделі ЦИХ абстракцій з предметної області.

У процесі виявлення об'єктів з предметних областей необхідно визначити, які зв'язки існують між ними. Особливий інтерес представляють два типи зв'язків: узагальнення (зв'язок між підкласами і надкласами) і агрегація (зв'язок між цілими

і частинами). Між класами можуть бути й інші відносини, включаючи найпростіші асоціації, але ці два надзвичайно важливі. Ми базуємо статичну модель на діаграмі класів, що представляє модель предметної області. Клас в UML — це місце, де розміщуються властивості (тобто дані члена) та операції (тобто функції, що виконуються об'єктом). Однак, починаючи моделювати предметну область, ми, як правило, не хочемо витратити багато часу на визначення атрибутів та операцій; ми розберемося з цим пізніше, коли уточнюємо та заповнюємо статичні частини моделі. Тепер ми повинні зосередитися на ідентифікації фактичних об'єктів і зв'язків між ними.

Однією з головних цілей побудови програм на основі реальних абстракцій є повторне використання. Пам'ятайте, що якщо ви збираєтеся використовувати щось повторно, вам потрібно дуже ретельно моделювати предметну область, оскільки на цьому етапі проявляється потенціал повторного використання. Таким чином, модель предметної області стає основою для статичної частини моделі. При моделюванні предметної області ми будемо дотримуватися підходу ОМТ (об'єктного моделювання), який характеризується орієнтацією навиворіт. Це означає, що ми починаємо з ключових об'єктів в системі, а потім переходимо назовні, щоб вивчити, як вони взаємодіють з іншими об'єктами. Отже, при виявленні прецедентної або динамічної частини системи ми рухаємося ззовні всередину, а при створенні статичної моделі - зсередини. Секрет полягає в тому, щоб рухатися в обох напрямках одночасно, зустрічаючись посередині, не залишаючи проміжків. Коли мова піде про аналізі придатності і діаграмах послідовності, ми побачимо, як це робиться. А поки запам'ятайте, що моделювання предметної області і статичну моделювання - це погляд на систему зсередини назовні.

Діаграма прецедентів (рис. 2.7) - це діаграма поведінки UML, яка зазвичай використовується для аналізу різних систем. Вони дозволяють вам уявити різні типи ролей у системі та те, як ці ролі взаємодіють із системою.

Основна мета цієї діаграми — описати функціональні можливості та поведінку, які дозволяють клієнтам, кінцевим користувачам і розробникам обговорювати разом проект або існуючу систему. При моделюванні систем з використанням прецедентних діаграм системні аналітики прагнуть:

- визначити дійових осіб (акторів), їх взаємодія з системою і очікувану функціональність системи;
- визначити в глосарії предметної області поняття, які стосуються детального опису функціональності системи (тобто прецедентів).

Робота над діаграмою може початися з текстового опису, отриманого при роботі із замовником. При цьому нефункціональні вимоги при складанні моделі прецедентів опускаються .

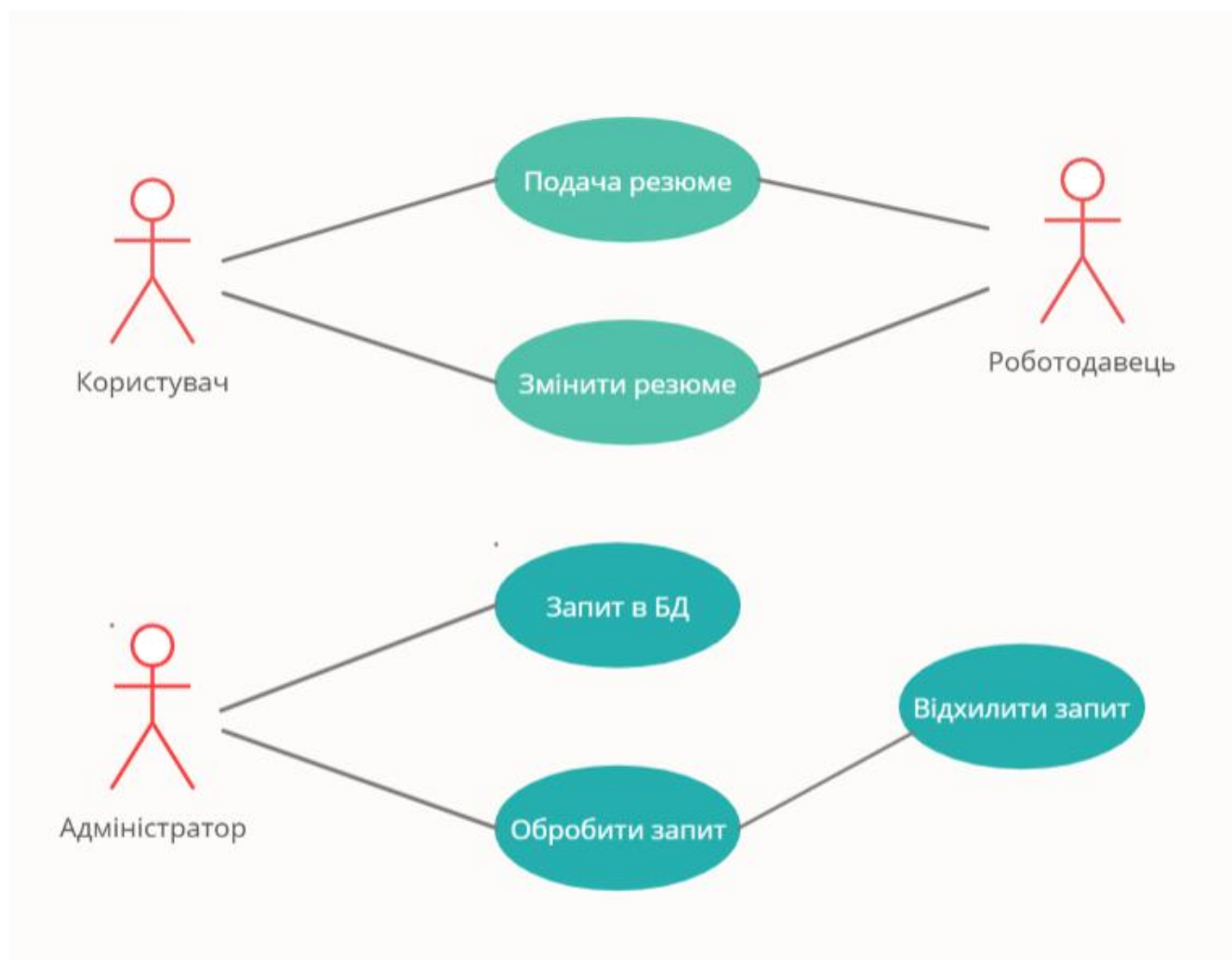


Рисунок 2.7 – Діаграма прецедентів

Робота з прототипом починається з авторизації та входу користувача на головну сторінку з переліком вакансій. Користувач має можливість натиснути на

один із них для щоб помітити її як обрану. Після натиснення на обрані користувач може перейти до збережених вакансій.

Моделювання структури системи

Архітектура програмного забезпечення — це структура програми або комп'ютерної системи, включаючи компоненти програмного забезпечення, видимі ззовні властивості цих компонентів і зв'язки між ними. Цей термін також застосовується до документування архітектури програмного забезпечення. Документування архітектури програмного забезпечення спрощує процес спілкування між зацікавленими сторонами, дозволяє задокументувати ранні етапи проектних рішень щодо проектування системи високого рівня та дозволяє повторно використовувати компоненти та шаблони цього дизайну в інших проєктах.

Розроблюваний веб-сайт запускається на комп'ютері, який має деяку апаратну конфігурацію і працює під управлінням певної операційної системи. Веб-орієнтовані програми часто вимагають для своєї роботи деякої ІТ-інфраструктури, зберігають інформацію в базах даних, розташованих на серверах, викликають веб-сервіси, використовують загальні ресурси.

Діаграма розгортання (англ. Deployment diagram) – діаграми в UML, які показують обчислювальні вузли під час виконання програм, компонентів та об'єктів на цих вузлах. Ці компоненти відповідають представленням робочих копій одиниць коду. Компоненти, які не представлені під час виконання програми на таких діаграмах, не відображаються, однак їх можна відобразити на діаграмах компонентів. Діаграми розгортання показують робочі екземпляри компонентів, а діаграми компонентів показують зв'язки між типами компонентів.

Розроблюваний веб-сайт запускається на комп'ютері, який має деяку апаратну конфігурацію і працює під управлінням певної операційної системи. Веб-орієнтовані програми часто вимагають для своєї роботи деякої ІТ-інфраструктури, зберігають інформацію в базах даних, розташованих на серверах, викликають веб-сервіси, використовують загальні ресурси. Ось для цього і потрібні діаграми розгортання, які іноді називають діаграмами розміщення. Такі діаграми є сенс

будувати тільки для апаратно-програмних систем, тоді як UML дозволяє будувати моделі будь-яких систем, не обов'язково комп'ютерних. По-перше, графічне представлення ІТ-інфраструктури може допомогти більш раціонально розподілити компоненти системи по вузлах мережі, від чого, як відомо, залежить в тому числі і продуктивність системи. По-друге, така діаграма може допомогти вирішити безліч допоміжних завдань, пов'язаних, наприклад, із забезпеченням безпеки. Діаграма розгортання показує топологію системи і розподіл компонентів системи за її вузлів, а також сполуки–маршрути передачі інформації між апаратними вузлами.

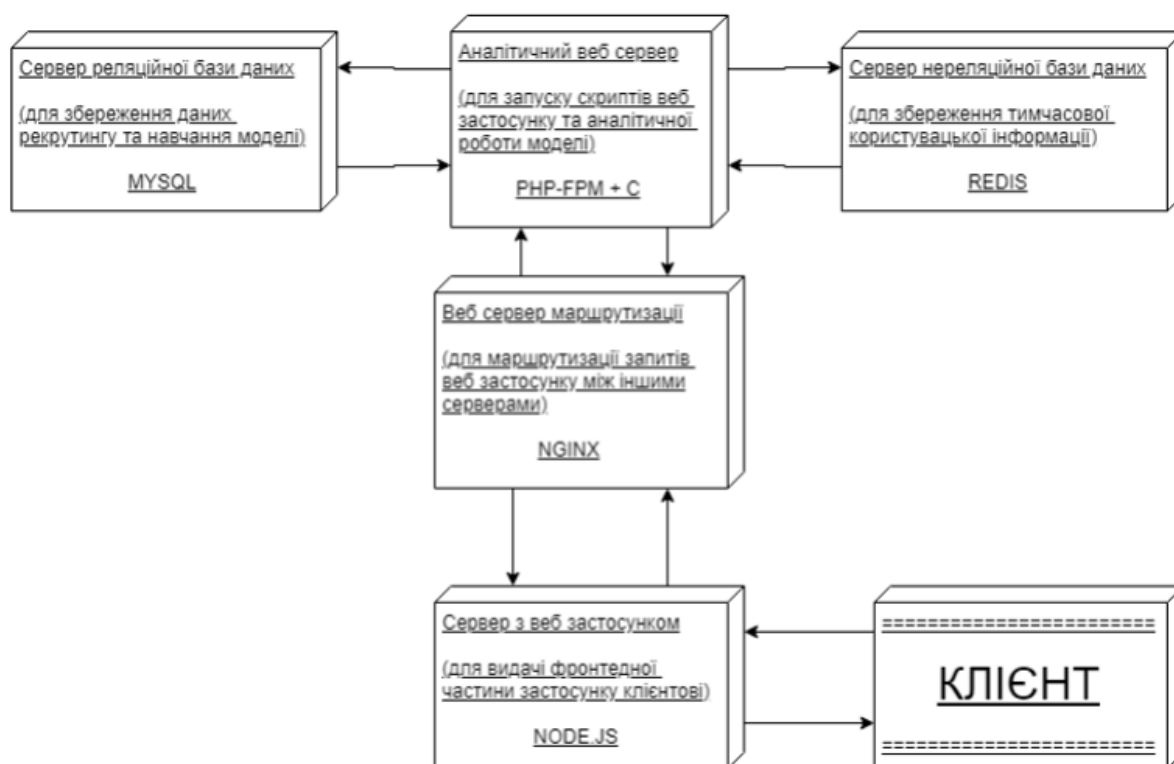


Рисунок 2.8 – Діаграма розгортання

Призначенням системи підтримка процесу рекрутингу в інформаційних системах з використанням нейронних мереж є спрощення пошуку вакансій і кандидатів користувачами системи, а також мінімізація людського фактору на підбір найкомпетентніших кандидатів. Цілі створення системи: – підвищення ефективності підбору кадрів з мінімізацією людського фактору і урахуванням імовірної подальшої успішності кандидатів. Для реалізації поставлених цілей система має вирішувати такі задачі:

- пошук найперспективніших кандидатів для вибраної вакансії;

- визначення найбільш підходящих вакансій для кандидата;
- ведення історії індивідуальної для кожного користувача, для накопичення «словника» для машинного навчання.

Для реалізації поставлених цілей система повинна вирішувати такі задачі:

1. Ведення бази кандидатів:

- a. Додавання нових кандидатів
- b. Редагування інформації про вже існуючих кандидатів
- c. Видалення інформації про кандидатів

2. Ведення бази вакансій

- a. Додавання нових вакансій
- b. Редагування інформації про вже існуючі вакансії
- c. Видалення інформації про вакансії

3. Автоматичний підбір вакансій

- a. Навчання та перенавчання моделі
- b. Визначення найкращої вакансії для кожного окремого кандидата
- c. Використання підходу дистиляції знань до навчання нейромережі.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ ПІДСИСТЕМИ

3.1. Інформаційне забезпечення

Інформаційне забезпечення - форма комплексу документів, нормативних актів та впроваджених рішень щодо обсягу, місця розташування та існування інформації, яка використовується в інформаційній системі.

Вхідні дані:

- інформація про користувача
- інформація про вакансію

Вихідні дані:

- резюме
- відповідь роботодавця

Інформаційне забезпечення організоване у вигляді бази даних, яка має розмішуватися на сервері. Для доступу до серверу з персонального комп'ютеру була використана система керування базами даних Microsoft SQL Server 2014, яка здатна обслуговувати системи з високим рівнем навантаження, що підкріплюється результатами стандартних галузевих тестів продуктивності.

Опис бази даних MySQL

В даний час MySQL є однією з найпопулярніших систем управління базами даних в Інтернеті. Ця система використовується для обробки великих обсягів інформації. Однак MySQL чудово підходить для невеликих проєктів та Інтернету. Важливою особливістю цієї системи є її ступені свободи.

Основні переваги MySQL це надійність, швидкість та гнучкість - основні особливості MySQL. Робота з цією системою не є складною, і підтримка сервера MySQL автоматично включається в доставку PHP. MySQL випускається за умовами Загальної публічної ліцензії GNU (GPL) .

Проте професійні методи роботи з файлами є складними та трудомісткими, оскільки потрібно ретельно вводити інформацію у файл, сортувати дані та робити такі заяви. Однак майте на увазі, що всі ці дії будуть виконуватися на сервері

провайдера хостингу, на якому може бути встановлена одна з установок Unix. Тому ви також повинні забезпечити безпечний доступ до файлів. У цьому випадку обсяг коду значно збільшується, а ймовірність програмних помилок висока.

На рисунку (рис. 3.1) зображено схему бази даних.

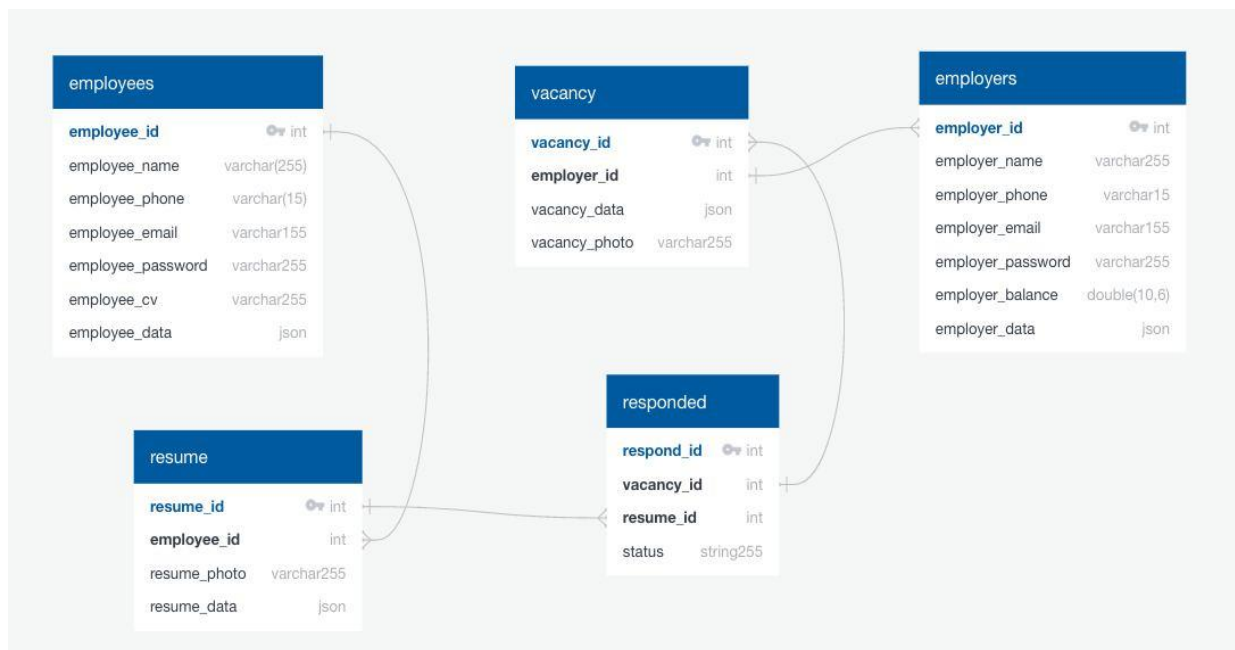


Рисунок 3.1 – Схема бази даних

База даних складається з таблиць які відображують інформацію для пошуку вакансій працевлаштування та ефективності інтренет додатку вцілому.

Опис таблиць бази даних

Таблиця employees (рис. 3.2) містить в собі такі поля, що описують кандидата:

- Первинний ключ – employee_id з типом даних (int)
- employee_name – ім'я кандидата з типом даних (varchar(255))
- employee_phone – телефон кандидата з типом даних (varchar(255))
- employee_email – пошта кандидата з типом даних (varchar(155))
- employee_password – пароль кандидата з типом даних (varchar(255))
- employee_cv – резюме кандидата з типом даних (varchar(255))
- employee_data – дані кандидата з типом даних (json)

employee_id	employee_name	employee_phone	employee_email	employee_password	employee_cv	employee_data
135	Петро	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
454	Петро	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
523	Слава	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
734	Андрей	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
1234	Слава	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
2423	Слава	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
3333	Петро	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
4564	Петро	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
5233	Андрей	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
5745	Слава	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
7567	Петро	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...
9767	Слава	38066246345	715qy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	was born.....	{ "age": 23, "exp": true, "name": "Андрей", "drive_...

Рисунок 3.2 – Таблица employees

Таблица resume (рис. 3.3) описує резюме та має поля:

- Первинний ключ – resume_id з типом даних (int)
- employee_id – з типом даних (int)
- resume_photo – фото кандидата з типом даних (varchar(255))
- resume_data – опис резюме (json)

resume_id	employee_id	resume_photo	resume_data
111	111	photo	{ "age": 25, "name": "Иван" }
223	322	photo	{ "age": 25, "name": "Иван" }
232	128	photo	{ "age": 25, "name": "Иван" }
345	143	photo	{ "age": 25, "name": "Иван" }
385	578	photo	{ "age": 25, "name": "Иван" }
462	134	photo	{ "age": 25, "name": "Иван" }
555	656	photo	{ "age": 25, "name": "Иван" }
634	978	photo	{ "age": 25, "name": "Иван" }
764	345	photo	{ "age": 25, "name": "Иван" }
777	777	photo	{ "age": 25, "name": "Иван" }
967	567	photo	{ "age": 25, "name": "Иван" }
998	674	photo	{ "age": 25, "name": "Иван" }

Рисунок 3.3 – Таблица resume

Таблица vacancy (рис. 3.4) має наступні поля:

- Первинний ключ – vacancy_id з типом даних (int)
- employer_id – з типом даних (int)
- vacancy_data – опис вакансії з типом даних (json)

- vacancy_photo – фото в вакансії з типом даних (varchar(255))

vacancy_id	employer_id	vacancy_data	vacancy_photo
12	653	{"cash": "30 000", "name": "architector", "remotel...	photo url
34	65	{"cash": "30 000", "name": "architector", "remotel...	photo url
36	78	{"cash": "30 000", "name": "architector", "remotel...	photo url
89	74	{"cash": "30 000", "name": "architector", "remotel...	photo url
344	553	{"cash": "30 000", "name": "architector", "remotel...	photo url
423	78	{"cash": "30 000", "name": "architector", "remotel...	photo url
444	555	{"cash": "30 000", "name": "architector", "remotel...	photo url
563	342	{"cash": "30 000", "name": "architector", "remotel...	photo url
567	234	{"cash": "30 000", "name": "architector", "remotel...	photo url
666	777	{"cash": "30 000", "name": "architector", "remotel...	photo url
884	774	{"cash": "30 000", "name": "architector", "remotel...	photo url
889	778	{"cash": "30 000", "name": "architector", "remotel...	photo url

Рисунок 3.4 – Таблиця vacancy

Таблиця responded (рис. 3.5) має такі поля:

- Первинний ключ – respond_id з типом даних (int)
- vacancy_id – з типом даних (int)
- resume_id – з типом даних (int)
- status – статус відповідей користувачу (refusing/approved/considered)

respond_id	vacancy_id	resume_id	status
123	452	546	considered
232	136	234	approved
233	678	456	considered
245	234	565	refusing
345	790	578	approved
443	859	232	refusing
454	123	543	considered
534	345	236	refusing
561	354	345	approved
567	345	987	considered
644	230	675	refusing
876	564	967	refusing

Рисунок 3.5 – Таблиця responded

Таблиця employers (рис. 3.6):

- Первинний ключ – employer_id з типом даних (int)

- `employer_name` – ім'я роботодавця з типом даних (`varchar(255)`)
- `employer_phone` – телефон роботодавця з типом даних (`varchar(255)`)
- `employer_email` – пошта роботодавця з типом даних (`varchar(155)`)
- `employer_password` – пароль роботодавця з типом даних (`varchar(255)`)
- `employer_balance` – зарплатня яку може запропонувати роботодавець з типом даних (`double (10,6)`)
- `employer_data` – дані роботодавця з типом даних (`json`)

employer_id	employer_name	employer_phone	employer_email	employer_password	employer_balance	employer_data
11	Анастасия	380444618061	rv7bp@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	9.000000	("type": "organization", "commercy": true, "employ...
45	Сергей	380444618061	yxunv@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	5.000000	("type": "organization", "commercy": true, "employ...
57	Сергей	380444618061	2shlaq@outlook.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	6.000000	("type": "organization", "commercy": true, "employ...
333	Сергей	380444618061	19dn@outlook.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	8.000000	("type": "organization", "commercy": true, "employ...
342		380444618061	281av0@gmail.com8edmfh@outlook.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	6.000000	("type": "organization", "commercy": true, "employ...
676	Анастасия	380444618061	hr6zd@yandex.ru	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	5.000000	("type": "organization", "commercy": true, "employ...
754	Сергей	380444618061	715cy08@gmail.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	4.000000	("type": "organization", "commercy": true, "employ...
756	Сергей	380444618061	wnhborq@outlook.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	8.000000	("type": "organization", "commercy": true, "employ...
784	Анастасия	380444618061	sfn13i@mail.ru	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	5.000000	("type": "organization", "commercy": true, "employ...
879	Сергей	380444618061	dcu@yandex.ru	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	7.000000	("type": "organization", "commercy": true, "employ...
999	Анастасия	380444618061	kaf93x@outlook.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	7.000000	("type": "organization", "commercy": true, "employ...
5642	Сергей	380444618061	g0orc3xt@outlook.com	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73...	7.000000	("type": "organization", "commercy": true, "employ...

Рисунок 3.6 – Таблица employers

3.2. Технічне забезпечення

Реалізація веб-додатка для організації пошукової діяльності включає розробку архітектури програми, створення бази даних, впровадження серверної частини, створення веб-частини та тестування. Для користування інформаційною системою користувач повинен мати пристрій з браузером та доступом до Інтернету. Нижче наведено мінімальні вимоги для використання Chrome на пристроях з різними операційними ядрами. Можливо, ви зможете встановити Chrome на інших платформах або попередніх операційних системах, але служба підтримки Google Enterprise відповідатиме лише на запити щодо систем, які відповідають мінімальним вимогам.

Windows

Вимоги до системи:

Windows 7, Windows 8, Windows 8.1, Windows 10 або більш пізньої версії. Процесор Intel Pentium 4 або новішої версії з підтримкою SSE3. На серверах повинна бути встановлена ОС Windows® Server 2008 R2, Windows® Server 2012 Windows® Server 2012 R2 або Windows® Server 2016.

MacOS

Вимоги до системи:

OS X El Capitan (10.11) або більш пізньої версії.

Linux

Вимоги до системи:

Ubuntu 14.04 (64-розрядної версії) або більш пізньої версії, Debian 8 або пізнішої версії, openSUSE 13.3 або більш пізньої версії, Fedora Linux 24 або пізнішої версії. Процесор Intel Pentium 4 або новішої версії з підтримкою SSE3.

Android

Вимоги до системи:

Android 5.0 (Lollipop)

Вибір серверу

Надійний сервер є ключовим фактором у визначенні продуктивності будь-якого веб-додатка. Це особливо важливо для власників інтернет-додатків: щоб полегшити пошук, користувачам важливо швидко завантажувати сторінки. Крім того, добре підібране серверне середовище може підвищити безпеку Інтернет-додатків. На що звернути увагу при виборі хостингу По-перше, вам потрібно зрозуміти, які вимоги до продуктивності програми потрібні. Скільки відвідувачів має веб-сайт? Яке навантаження на користувачів очікується? Який обсяг даних можна обробляти одночасно? Скільки вакансій є на сайті? Отримавши відповіді на ці питання, ви зможете визначити якісні характеристики вашого майбутнього хостинг-провайдера:

Продуктивність: потужність процесора і кількість процесів, що запускаються одночасно. Хорошим умовою вважається, якщо процесори серверів хостинг-провайдера завантажені не більше, ніж на 40%. Це дає необхідний запас міцності.

Пропускна здатність: обсяг переданих даних за обрану одиницю часу. Не менш 100 Мбіт / с.

Оперативна пам'ять: не менше 2 Гб, якщо великий обсяг баз даних, то більше.

Продуктивність дискової підсистеми: по типу надаються сервера з механічними дисками (HDD) і на флеш-накопичувачах (SSD). Другі краще за швидкістю, але обсяг коштує дорожче.

Варто також взяти до уваги вимоги до стека технологій Magento 2.3. Це важливо при виборі характеристик серверів хостинг-провайдера:

- Операційна система (Linux x86-64)
- Вимоги до пам'яті: не менше 2 Гб оперативної пам'яті
- Composer (остання стабільна версія)
- Веб-сервер: Apache 2.2 або 2.4, nginx 1.x
- База даних: MySQL 5.6, 5.7
- PHP: 7.1.3, 7.2.0 (Magento 2.3.1 сертифікований і протестований на PHP 7.2.11)

Крім того, вкрай рекомендується:

- підтримка Elasticsearch версії 6.x
- Varnish версії 4.x або 5.2
- Redis 3.2, 4.0, 5.0 (сумісні з 2.4+), краще 5.0.

3.3. Програмне забезпечення

Так як рішення задачі є кросбраузерним та кросплатформенним для роботи користувача з сайтом потрібно мати пристрій з ядром одної з операційних систем (Windows, Linux, IOS, Android, MacOS та ін.) з можливістю мати вихід в Інтернет та лише встановити один з браузерів (Opera, Chrome, Firefox, Edge, IE або ін.).

Рекомендовано використовувати Chrome як найбільш популярний з них та підтримує багато крутих фіч.

Для розробки проєкт була використана IDE – VS Code.

Visual Studio Code — це редактор вихідного коду Microsoft для Windows, Linux і macOS. Позиціонується як «простий» редактор коду для розробки кросплатформних веб-додатків. Включає налагоджувач, інструменти для роботи з Git, підсвічування синтаксису, IntelliSense та інструменти рефакторингу. Він має широкі можливості налаштування: власні теми, комбінації клавіш і профілі. Безкоштовний дистрибутив, розроблений як програмне забезпечення з відкритим кодом, але готові збірки поширюються за ліцензією.

Вибір мов програмування

JavaScript - — це мова, яка дозволяє застосовувати складний вміст на веб-сторінках, одночасно відображаючи на них нестатичний вміст — відображення регулярно оновлюваного вмісту, інтерактивних карт, 2D/3D графічної анімації або прокручування відео в програвачі тощо. - Ви можете бути впевнені, мабуть, без JavaScript. Це третій рівень стандартних веб-технологій, їх два (HTML і CSS). Ядро JavaScript містить багато поширених функцій, які дозволяють робити наступне: Зберігайте дані у змінних. Наприклад, у наведеному вище прикладі ми запитали нове ім'я, а потім зберегли це ім'я у змінній імені. Операції над фрагментами тексту (так звані «рядками» в програмуванні). У наведеному вище прикладі ми додаємо рядок «Player 1:» до значення імені змінної, щоб отримати повний текст, наприклад: «Player 1: Chris».

Запускати код відповідно до визначених подійми що відбуваються на web сторінці. У нашому прикладі вище, ми використовували click (en-US) подія, для

визначення моменту, коли кнопка була клікнута, відповідно до цього запускався код, який оновлював текст.

Ще цікавішим є функціональність, створена поверх основної мови JavaScript. Так звані інтерфейси прикладного програмування (API) надають вам додаткову функціональність для використання у вашому коді JavaScript. API — це готові набори кодових блоків, які дозволяють розробникам реалізовувати програми, які в іншому випадку були б складними або неможливими. Вони роблять те саме з програмуванням, що і готовий набір меблів у домашньому будинку - купити готові панелі та прикрутити їх разом, щоб зробити книжкову полицю, набагато простіше, ніж розробити дизайн самостійно, знайти потрібну деревину, розрізати всі панелі розміру та форми, знайдіть потрібні шурупи та зберіть їх разом, щоб зробити книжкову полицю.

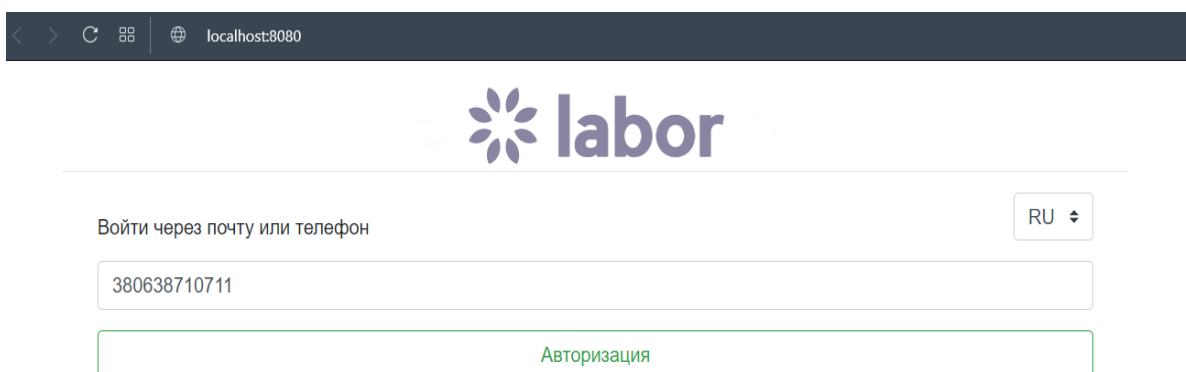
PHP (рекурсивна аббревіатура PHP: Hypertext Preprocessor) — поширена мова програмування загального призначення з відкритим кодом. PHP розроблений для веб-розробки, і його код можна реалізувати безпосередньо в HTML. PHP відрізняється від JavaScript тим, що PHP-скрипти запускаються на сервері і генерують HTML, який надсилається клієнту. Ви навіть можете налаштувати свій сервер так, щоб процесор PHP обробляв звичайні файли HTML, щоб клієнти навіть не знали, чи отримують вони звичайні файли HTML чи результати сценаріїв.

Вибір СКБД

Microsoft SQL Server — це система керування базами даних, спільно розроблена Microsoft і Sybase, іменована СУБД. Це дозволяє ефективно керувати базами даних будь-якого розміру - від персональних баз даних до будь-якої інфраструктури баз даних підприємства. За роки розробки цієї системи Microsoft продовжувала вдосконалювати свої продукти, впроваджуючи нові технології, підвищуючи продуктивність, запроваджуючи нові сценарії переходу та інтегруючи систему з іншими програмами. Результатом постійного вдосконалення є останній продукт Microsoft SQL Server 2014, запущений 1 квітня 2014 року. Після багатьох років удосконалень система управління базою даних допоможе ефективно організувати та обробити дані та транзакції, а також дозволить вашій компанії досягти нових рівнів продуктивності та перевершити конкурентів.

3.4. Результати реалізації інформаційної системи/підсистеми

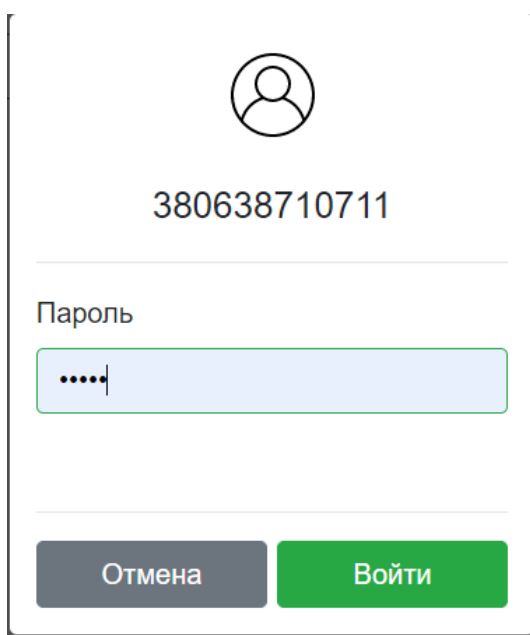
На першому кроці користувач попадає на сторінку авторизації (рис 3.2)



The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The page features the 'labor' logo at the top center. Below the logo, there is a text input field labeled 'Войти через почту или телефон' with the value '380638710711'. To the right of this field is a language selector dropdown showing 'RU'. Below the input field is a green button labeled 'Авторизация'.

Рисунок 3.2 – Сторінка авторизації.

Потім треба ввести пароль (рис 3.3)



The screenshot shows a modal form for login. At the top is a user icon placeholder. Below it is the phone number '380638710711'. Underneath is a label 'Пароль' followed by a password input field containing five dots. At the bottom are two buttons: 'Отмена' (grey) and 'Войти' (green).

Рисунок 3.2 – Модальна сторінка входу.

Якщо вхід успішний, попадаємо на головну сторінку з вакансіями. (рисунк 3.3)

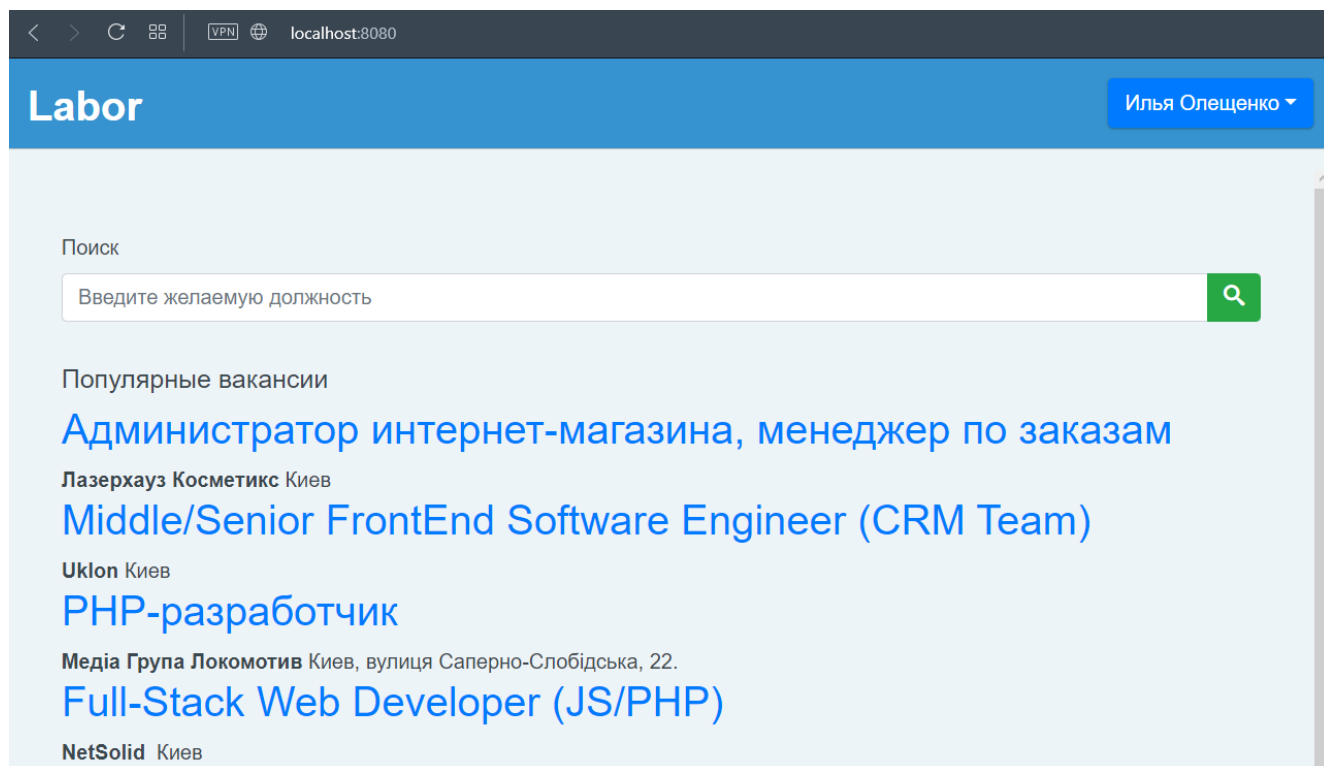


Рисунок 3.3 – головна сторінка з вакансіями

В боковій панелі можна потрапити на сторінку свого резюме, налаштування, техпідтримку або вийти зі свого облікового запису.

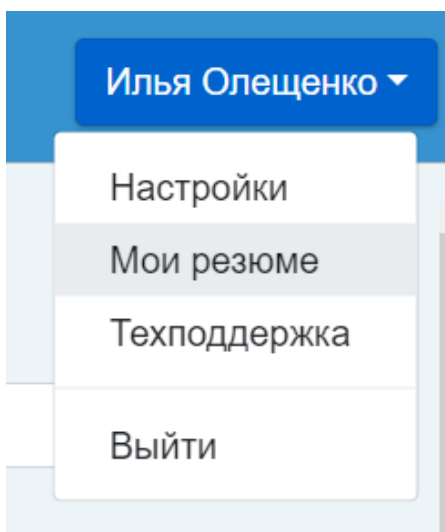


Рисунок 3.4 – бокове меню

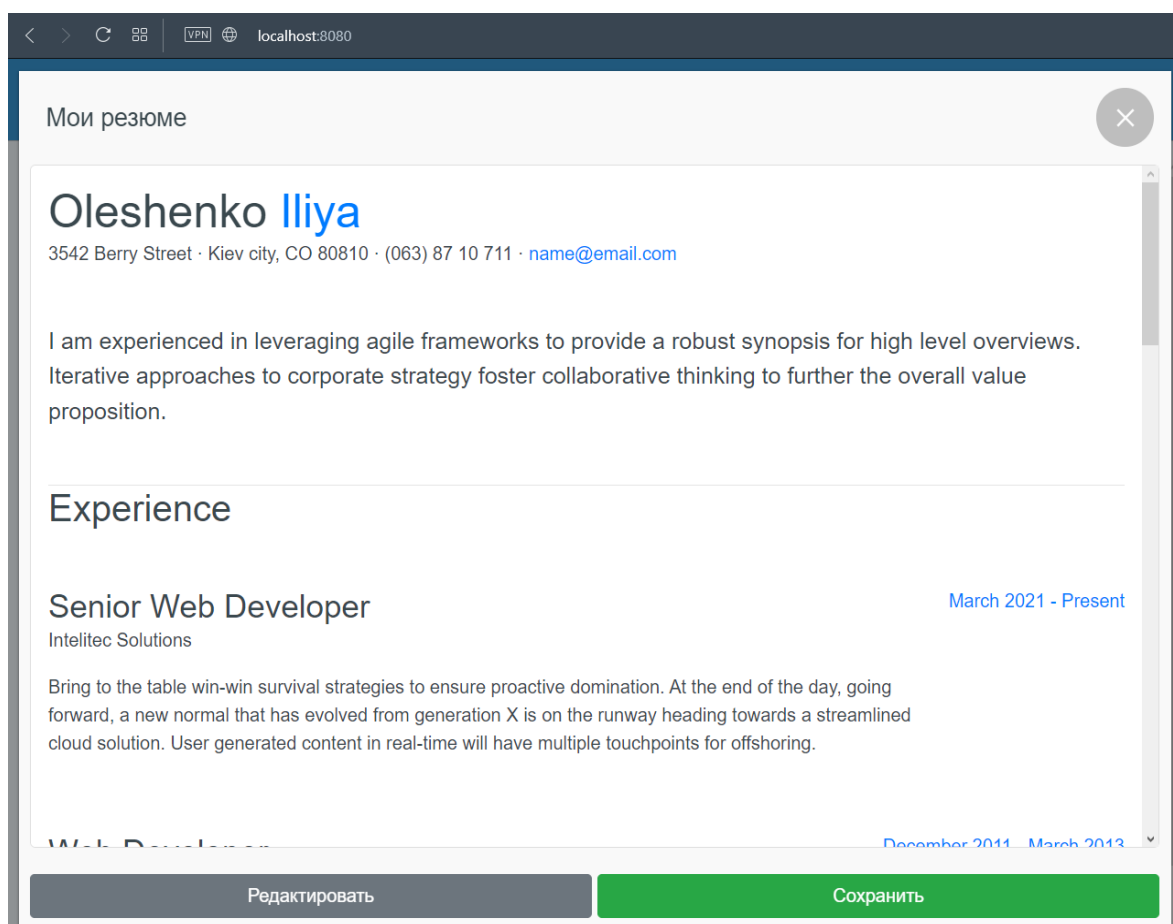


Рисунок 3.5 – модальне вікно з резюме

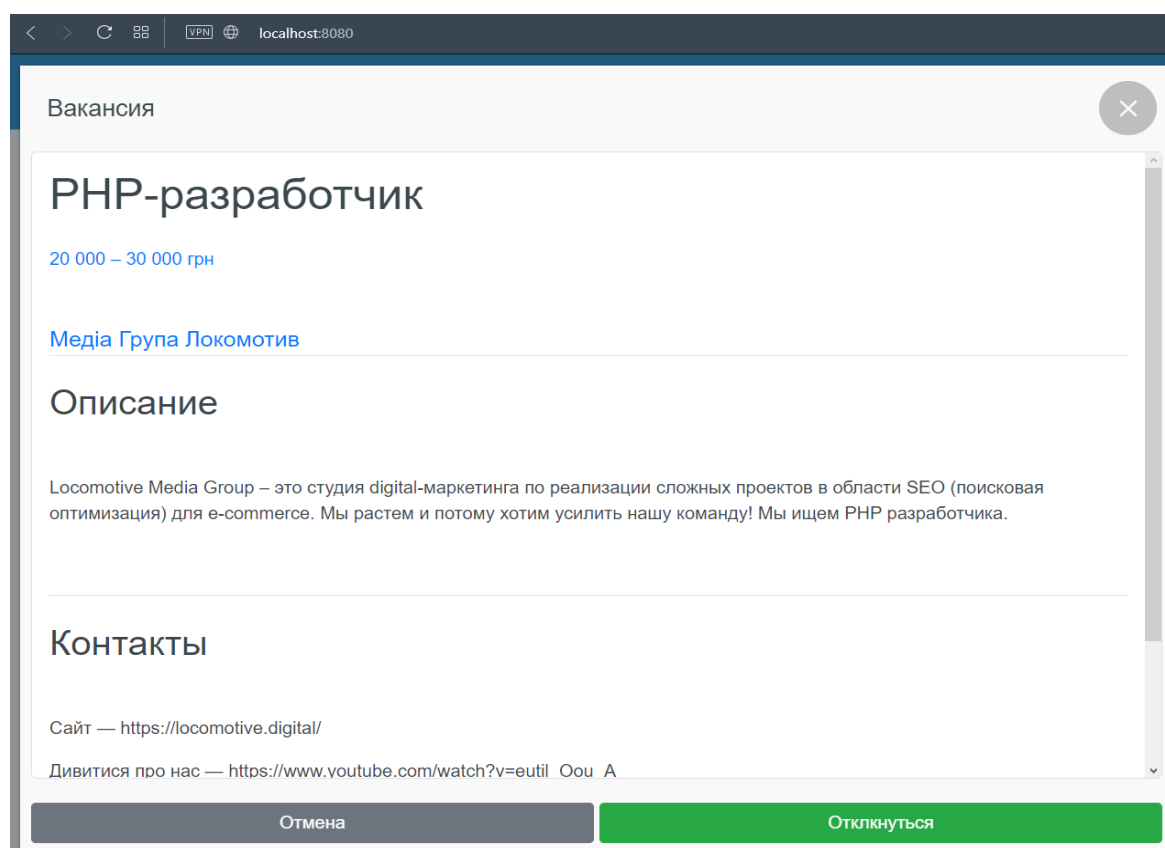


Рисунок 3.6 – модальне вікно вакансії

ВИСНОВКИ

Тема, що була досліджена при написанні дипломної роботи – пришвидшити процес підбору персоналу та спростити його.

Спираючись на вище описану настанову користувача можна прийти до висновку що система достатньо дружня до користувачів і дозволяє без великих часових витрат навчитися використовувати систему для пошуку та підбору кандидатів. Дизайн розроблено з урахуванням особливостей використання системи і спроектовано таким чином щоб новому користувачу була видна вся найважливіша інформація, а досвідченому забезпечити швидкий доступ до потрібних йому елементів сторінки.

Для отримання даних про поточних кандидатів було створено систему котра дозволяє додавати, редагувати та змінювати надані ними анкети та резюме. Аналогічно в системі була реалізована можливість управління вакансіями для того щоб рекрутери могли сповіщати кандидатів про необхідність зайняття робочого місця.

Звісно окрім можливості зберігати та редагувати інформацію про кандидатів, вакансії, важливість певних пунктів вакансії система використовує алгоритми машинного навчання для фільтрації резюме та рекомендацій кандидатам вакансій і навпаки. Для отримання такого результату було використано декілька підходів та алгоритмів. Першим є метод опорних векторів - один з класичних алгоритмів машинного навчання котрий дозволяє прогнозувати професію кандидата за його анкетною, а також підбирати найкращого кандидата до вакансії. Для порівняння та покращення результатів було використано ще один алгоритм машинного навчання та проведена його модифікація.

Визначені чинники впливу, стимулятори і обмежувальні фактори. Дослідження ринку існуючих інформаційних систем, призначених для компанії електронних додатків, виявив, що системи, які могли забезпечити виконання всього, вимагали б багато часу та вивчення. І системи, які є безкоштовними, в повному обсязі здатні виконувати всі необхідні завдання. У документі вказується список вимог, які необхідно було виконати. Визначено необхідне обладнання і

програмне забезпечення. Чітко визначені основні бізнес-процеси, які необхідно виконати для розвитку системи. Були розроблені діаграми та побудовано блок схеми алгоритмів роботи системи.

Результатом роботи є оригінальна автоматизована система для компаній, що працюють з працевлаштуванням. Розвинений зручний і зрозумілий інтерфейс взаємодії клієнта з базою даних. Для створення бази даних використовувався MS SQL Server. Для створення об'єктів бази даних (таблиці, відносини між таблицями, стандарти значення, обмеження, збережені процедури) була використана мова запитів SQL. Додаток було розроблено в «Microsoft Visual Studio Code» з використанням таких мов програмування як JS та PHP. Для повної працездатності автоматизованої системи при кінцевій роботі потрібен тільки браузер. Розроблено та протестовано керівництво користувача. додаток для оцінки якості роботи.

Аналіз даних випробувань показав, що додаток виконує свої основні функції без помилок і тому підходить для використання.

Таким чином, завдання вирішені в повному обсязі, мета досягнута спроектована інформаційна система для пошуку вакансій, яка підвищить ефективність використання користувачами веб-додатку, що є актуальним в цілях проєктування АІС.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація по JS [Електронний ресурс]:
<https://learn.javascript.ru/intro#chto-takoe-javascript>
2. Документація по JS [Електронний ресурс]:
https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript
3. Документація по MySQL [Електронний ресурс] – Режим доступу до ресурсу:
<http://www.mysql.ru/docs/mysql-man-4.0-ru/>
4. Бітрікс: Управління сайтом: офіційний сайт [Електронний ресурс] / Режим доступу до ресурсу: <https://www.bitrix.ua>.
5. Возний М. І. Міжнародна електронна торгівля. Проблеми та перспективи розвитку в Україні /М. І. Возний // Збірник наук. праць Буковинського університету. Економічні науки. - 2011. - Вип. 7. - С. 243–252.
6. Волков, О. Стандарты и методологии моделирования бизнеспроцессов [Електронний ресурс]. – Режим доступу до ресурсу:
<http://www.connect.ru/article.asp?id=5799>.
7. Ганин А.Н. Применение информационных технологий для моделирования бизнес-процессов на предприятии радиоэлектронного комплекса // Российское предпринимательство. – 2016. – Том 17. – № 22. – С. 3171–3184.
8. Гольшевский Д. В. Использование технологии Asp.Net для создания онлайн-ресурсов / Д. В. Гольшевский, В. А. Губин. // Системи обробки інформації.. – 2012. – №3. – С. 109–111.
9. Дослідження інтернет-аудиторії [Електронний ресурс] // Інтернет Асоціація України – Режим доступу до ресурсу: <http://inau.ua>
10. Информационная система [Електронний ресурс] / Википедия. Свободная энциклопедия. – Режим доступу до ресурсу:
http://ru.wikipedia.org/wiki/Информационная_система. 88
11. Карасева О.А. Конспект лекций по дисциплине БЗ.Б.2 «Моделирование бизнес-процессов». – Екатеринбург: Институт экономики и управления Уральского государственного лесотехнического университета, 2014. – 55 с.

12. Класс GridView [Електронний ресурс] // Microsoft – Режим доступу до ресурсу: [https://msdn.microsoft.com/ru/library/system.web.ui.webcontrols.gridview\(v=vs.110\).aspx](https://msdn.microsoft.com/ru/library/system.web.ui.webcontrols.gridview(v=vs.110).aspx)
13. Кудіна, О. Ю. Розвиток електронної торгівлі в умовах становлення глобального інформаційного простору [Текст] / О. Ю. Кудіна // Бюлетень Міжнародного Нобелівського економічного форуму. – 2011. – № 1 (4). – С. 196–202.
14. Нотация IDEF0 [Електронний ресурс] // Business Studio Wiki – Режим доступу до ресурсу: <http://www.businessstudio.ru/wiki/docs/v4/doku.php/ru/csdesign/bpmodeling/idef0>.
15. Патраманська Л.Ю. Електронна комерція: переваги та недоліки / Л.Ю. Патраманська. // Ефективна економіка. – 2015. – №11. – С. 55–60.
16. Рынок e-commerce в Украине [Електронний ресурс] // Кредитпромбанк. – Режим доступу до ресурсу: http://www.kreditprombank.com/upload/content/512/E-commerce_Market_in_Ukraine.pdf.
17. Система управління сайтами DIAFAN.CMS: офіційний сайт [Електронний ресурс] / DIAFAN.CMS – Режим доступу до ресурсу: <https://www.diafan.ru>
18. Трёхуровневая архитектура [Електронний ресурс] // Википедия. Свободная энциклопедия – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Трёхуровневая_архитектура.
19. Філіппова Л. Л. Електронна комерція: за і проти / Л. Л. Філіппова // Вісник Нац. техн. ун-туХП. – Харків : НТУ "ХП". – 2013. – № 44 (1017). – С. 58-65.
20. Ховрак І. В. Електронна комерція в Україні: переваги та недоліки / І. В. Ховрак // Економіка. Фінанси. Право. - 2013. - № 4. - С. 16-20. 89
21. Электронная коммерция (e-Commerce): тренды и прогноз развития в Украине на 2017-2018 гг. [Електронний ресурс] // Агентство Интернетмаркетинга «Веб-машина» – Режим доступу до ресурсу: <https://www.webmashina.com/web-blog/ecommerce-prognoz-elektronnoi-kommercii-ukrainy2017-2018>.

22. Бізнес-вимоги [Електронний ресурс] – Режим доступу до ресурсу: <http://um.co.ua/8/8-2/8-218955.html>.
23. Functional Requirements, Non-Functional Requirements [Електронний ресурс] – Режим доступу до ресурсу: <https://www.quality-assurance-group.com/requirement-types/>.
24. Діаграма прецедентів [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D1%96%D0%B2.
25. Диаграмма последовательности [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8

ДОДАТКИ

ДОДАТОК 1

Текст програми

```

<?php
/**
 * Confirms that the activation key that is sent in an email after a user signs
 * up for a new site matches the key for that user and then displays confirmation.
 *
 * @package WordPress
 */

define( 'WP_INSTALLING', true );

/** Sets up the WordPress Environment. */
require __DIR__ . '/wp-load.php';

require __DIR__ . '/wp-blog-header.php';

if ( ! is_multisite() ) {
    wp_redirect( wp_registration_url() );
    die();
}

$valid_error_codes = array( 'already_active', 'blog_taken' );

list( $activate_path ) = explode( '?', wp_unslash( $_SERVER['REQUEST_URI'] ) );
$activate_cookie      = 'wp-activate-' . COOKIEHASH;

$key = "";
$result = null;

if ( isset( $_GET['key'] ) && isset( $_POST['key'] ) && $_GET['key'] !== $_POST['key'] ) {
    wp_die( __( 'A key value mismatch has been detected. Please follow the link provided in your activation email.' ), __( 'An error occurred during the activation' ), 400 );
} elseif ( ! empty( $_GET['key'] ) ) {
    $key = $_GET['key'];
} elseif ( ! empty( $_POST['key'] ) ) {
    $key = $_POST['key'];
}

if ( $key ) {
    $redirect_url = remove_query_arg( 'key' );

    if ( remove_query_arg( false ) !== $redirect_url ) {
        setcookie( $activate_cookie, $key, 0, $activate_path, COOKIE_DOMAIN, is_ssl(), true );
        wp_safe_redirect( $redirect_url );
        exit;
    } else {
        $result = wpmu_activate_signup( $key );
    }
}

```

```

    }
}

if ( null === $result && isset( $_COOKIE[ $activate_cookie ] ) ) {
    $key   = $_COOKIE[ $activate_cookie ];
    $result = wpmu_activate_signup( $key );
    setcookie( $activate_cookie, '', time() - YEAR_IN_SECONDS, $activate_path, COOKIE_DOMAIN,
    is_ssl(), true );
}

if ( null === $result || ( is_wp_error( $result ) && 'invalid_key' === $result->get_error_code() ) ) {
    status_header( 404 );
} elseif ( is_wp_error( $result ) ) {
    $error_code = $result->get_error_code();

    if ( ! in_array( $error_code, $valid_error_codes, true ) ) {
        status_header( 400 );
    }
}

nocache_headers();

if ( is_object( $wp_object_cache ) ) {
    $wp_object_cache->cache_enabled = false;
}

// Fix for page title.
$wp_query->is_404 = false;

/**
 * Fires before the Site Activation page is loaded.
 *
 * @since 3.0.0
 */
do_action( 'activate_header' );

/**
 * Adds an action hook specific to this page.
 *
 * Fires on {@see 'wp_head'}.
 *
 * @since MU (3.0.0)
 */
function do_activate_header() {
    /**
     * Fires before the Site Activation page is loaded.
     *
     * Fires on the {@see 'wp_head'} action.
     *
     * @since 3.0.0
     */

```

```

do_action( 'activate_wp_head' );
}
add_action( 'wp_head', 'do_activate_header' );

/**
 * Loads styles specific to this page.
 *
 * @since MU (3.0.0)
 */
function wpmu_activate_stylesheet() {
    ?>
    <style type="text/css">
        form { margin-top: 2em; }
        #submit, #key { width: 90%; font-size: 24px; }
        #language { margin-top: .5em; }
        .error { background: #f66; }
        span.h3 { padding: 0 8px; font-size: 1.3em; font-weight: 600; }
    </style>
    <?php
}
add_action( 'wp_head', 'wpmu_activate_stylesheet' );
add_action( 'wp_head', 'wp_strict_cross_origin_referrer' );
add_filter( 'wp_robots', 'wp_robots_sensitive_page' );

get_header( 'wp-activate' );

$blog_details = get_blog_details();
?>

<div id="signup-content" class="widecolumn">
    <div class="wp-activate-container">
        <?php if ( ! $key ) { ?>

            <h2><?php _e( 'Activation Key Required' ); ?></h2>
            <form name="activateform" id="activateform" method="post" action="<?php echo network_site_url( $blog_details->path . 'wp-activate.php' ); ?>">
                <p>
                    <label for="key"><?php _e( 'Activation Key:' ); ?></label>
                    <br /><input type="text" name="key" id="key" value="" size="50" />
                </p>
                <p class="submit">
                    <input id="submit" type="submit" name="Submit" class="submit" value="<?php esc_attr_e( 'Activate' ); ?>" />
                </p>
            </form>

            <?php
        } else {
            if ( is_wp_error( $result ) && in_array( $result->get_error_code(), $valid_error_codes, true ) ) {
                $signup = $result->get_error_data();
            }
        }
    }
}

```

```

<h2><?php _e( 'Your account is now active!' ); ?></h2>
<?php
echo '<p class="lead-in">';
if ( " === $signup->domain . $signup->path ) {
    printf(
        /* translators: 1: Login URL, 2: Username, 3: User email address, 4: Lost password URL. */
        __( 'Your account has been activated. You may now <a href="%1$s">log in</a> to the site using your chosen username of &#8220;%2$s&#8221;. Please check your email inbox at %3$s for your password and login instructions. If you do not receive an email, please check your junk or spam folder. If you still do not receive an email within an hour, you can <a href="%4$s">reset your password</a>.' ),
        network_site_url( $blog_details->path . 'wp-login.php', 'login' ),
        $signup->user_login,
        $signup->user_email,
        wp_lostpassword_url()
    );
} else {
    printf(
        /* translators: 1: Site URL, 2: Username, 3: User email address, 4: Lost password URL. */
        __( 'Your site at %1$s is active. You may now log in to your site using your chosen username of &#8220;%2$s&#8221;. Please check your email inbox at %3$s for your password and login instructions. If you do not receive an email, please check your junk or spam folder. If you still do not receive an email within an hour, you can <a href="%4$s">reset your password</a>.' ),
        sprintf( '<a href="http://%1$s%2$s">%1$s%2$s</a>', $signup->domain, $blog_details->path ),
        $signup->user_login,
        $signup->user_email,
        wp_lostpassword_url()
    );
}
echo '</p>';
} elseif ( null === $result || is_wp_error( $result ) ) {
    ?>
    <h2><?php _e( 'An error occurred during the activation' ); ?></h2>
    <?php if ( is_wp_error( $result ) ) : ?>
        <p><?php echo $result->get_error_message(); ?></p>
    <?php endif; ?>
    <?php
} else {
    $url = isset( $result['blog_id'] ) ? get_home_url( (int) $result['blog_id'] ) : '';
    $user = get_userdata( (int) $result['user_id'] );
    ?>
    <h2><?php _e( 'Your account is now active!' ); ?></h2>

    <div id="signup-welcome">
    <p><span class="h3"><?php _e( 'Username:' ); ?></span> <?php echo $user->user_login; ?></p>
    <p><span class="h3"><?php _e( 'Password:' ); ?></span> <?php echo $result['password']; ?></p>
    </div>

    <?php
    if ( $url && network_home_url( '', 'http' ) !== $url ) :
        switch_to_blog( (int) $result['blog_id'] );
        $login_url = wp_login_url();

```

```

        restore_current_blog();
        ?>
        <p class="view">
        <?php
            /* translators: 1: Site URL, 2: Login URL. */
            printf( __( 'Your account is now activated. <a href="%1$s">View your site</a> or <a href="%2
$s">Log in</a>' ), $url, esc_url( $login_url ) );
        ?>
        </p>
        <?php else : ?>
        <p class="view">
        <?php
            printf(
                /* translators: 1: Login URL, 2: Network home URL. */
                __( 'Your account is now activated. <a href="%1$s">Log in</a> or go back to the <a href="%
2$s">homepage</a>.' ),
                network_site_url( $blog_details->path . 'wp-login.php', 'login' ),
                network_home_url( $blog_details->path )
            );
        ?>
        </p>
        <?php
        endif;
    }
}
?>
</div>
</div>
<script type="text/javascript">
    var key_input = document.getElementById('key');
    key_input && key_input.focus();
</script>
<?php
get_footer( 'wp-activate' );

```

```

<?php
/**
 * Loads the WordPress environment and template.
 *
 * @package WordPress
 */

```

```

if ( ! isset( $wp_did_header ) ) {

```

```

    $wp_did_header = true;

```

```

    // Load the WordPress library.
    require_once __DIR__ . '/wp-load.php';

```

```

// Set up the WordPress query.
wp();

// Load the theme template.
require_once ABSPATH . WPINC . '/template-loader.php';

}

<?php
/**
 * Handles Comment Post to WordPress and prevents duplicate comment posting.
 *
 * @package WordPress
 */

if ( 'POST' !== $_SERVER['REQUEST_METHOD'] ) {
    $protocol = $_SERVER['SERVER_PROTOCOL'];
    if ( ! in_array( $protocol, array( 'HTTP/1.1', 'HTTP/2', 'HTTP/2.0' ), true ) ) {
        $protocol = 'HTTP/1.0';
    }

    header( 'Allow: POST' );
    header( "$protocol 405 Method Not Allowed" );
    header( 'Content-Type: text/plain' );
    exit;
}

/** Sets up the WordPress Environment. */
require __DIR__ . '/wp-load.php';

nocache_headers();

$comment = wp_handle_comment_submission( wp_unslash( $_POST ) );
if ( is_wp_error( $comment ) ) {
    $data = (int) $comment->get_error_data();
    if ( ! empty( $data ) ) {
        wp_die(
            '<p>' . $comment->get_error_message() . '</p>',
            __( 'Comment Submission Failure' ),
            array(
                'response' => $data,
                'back_link' => true,
            )
        );
    } else {
        exit;
    }
}

$user = wp_get_current_user();
$cookies_consent = ( isset( $_POST['wp-comment-cookies-consent'] ) );

```

```

/**
 * Perform other actions when comment cookies are set.
 *
 * @since 3.4.0
 * @since 4.9.6 The `cookies_consent` parameter was added.
 *
 * @param WP_Comment $comment      Comment object.
 * @param WP_User    $user         Comment author's user object. The user may not exist.
 * @param bool       $cookies_consent Comment author's consent to store cookies.
 */
do_action( 'set_comment_cookies', $comment, $user, $cookies_consent );

$location = empty( $_POST['redirect_to'] ) ? get_comment_link( $comment ) : $_POST['redirect_to'] .
'#comment-' . $comment->comment_ID;

// If user didn't consent to cookies, add specific query arguments to display the awaiting moderation m
essage.
if ( ! $cookies_consent && 'unapproved' === wp_get_comment_status( $comment ) && ! empty( $co
mment->comment_author_email ) ) {
    $location = add_query_arg(
        array(
            'unapproved' => $comment->comment_ID,
            'moderation-hash' => wp_hash( $comment->comment_date_gmt ),
        ),
        $location
    );
}

/**
 * Filters the location URI to send the commenter after posting.
 *
 * @since 2.0.5
 *
 * @param string $location The 'redirect_to' URI sent via $_POST.
 * @param WP_Comment $comment Comment object.
 */
$location = apply_filters( 'comment_post_redirect', $location, $comment );

wp_safe_redirect( $location );
exit;

<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:

```

```

*
* * MySQL settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://wordpress.org/support/article/editing-wp-config-php/
*
* @package WordPress
*/

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'database_name_here' );

/** MySQL database username */
define( 'DB_USER', 'username_here' );

/** MySQL database password */
define( 'DB_PASSWORD', 'password_here' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users
 * to have to log in again.
 *
 * @since 2.6.0
 */
define( 'AUTH_KEY',          'put your unique phrase here' );
define( 'SECURE_AUTH_KEY',   'put your unique phrase here' );
define( 'LOGGED_IN_KEY',     'put your unique phrase here' );
define( 'NONCE_KEY',         'put your unique phrase here' );
define( 'AUTH_SALT',         'put your unique phrase here' );
define( 'SECURE_AUTH_SALT',   'put your unique phrase here' );
define( 'LOGGED_IN_SALT',    'put your unique phrase here' );
define( 'NONCE_SALT',        'put your unique phrase here' );

/**#@-*/

```

```

/**
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'wp_';

/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
 * visit the documentation.
 *
 * @link https://wordpress.org/support/article/debugging-in-wordpress/
 */
define( 'WP_DEBUG', false );

/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';

<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress

```

```

*/

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'mf390032_burger' );

/** MySQL database username */
define( 'DB_USER', 'mf390032_burger' );

/** MySQL database password */
define( 'DB_PASSWORD', 'Burger12345' );

/** MySQL hostname */
define( 'DB_HOST', 'mf390032.mysql.tools' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users
 * to have to log in again.
 *
 * @since 2.6.0
 */
define( 'AUTH_KEY',      'z2YQGIZ4&$kot92CqWS9[ Q@h|JzD |}dA DH4t*SX~/QQ6>Ctg#@{9o
9^Mk[cX-' );
define( 'SECURE_AUTH_KEY', 'k8n,bMeeEL0_KVmBnYa/G%3v?.;Rtb-4idpRif-
113Tm#(1MI~Sn94@w.H{+5Lv' );
define( 'LOGGED_IN_KEY',  '.g/?d1}8;DbWD<}07ZO[0v__~<=C7gIW=?K3lBuS8fnyr+Mvh(IL.1
}|CtY+pIXV' );
define( 'NONCE_KEY',      'f@Mh99V:`V|*-t/Y_LJBgc,|A;~QN4;<;~JI4o9YuuMpNO_/c$Bp,$-
uMn:T_=gU' );
define( 'AUTH_SALT',      '5p$*RDjBP$~IdxXI<+|z#>l&2=o&x(Q=TGvc>HU Vi!xhlq[U,/mT];`qT
8ls<6g' );
define( 'SECURE_AUTH_SALT', 'bh~w33`$i|1C=~hjP^|M)IB6qT1FF2s*cU|}[8DgWb]!h-
QJ]:FR^ {CY[ 8=+RjY' );
define( 'LOGGED_IN_SALT',  ' )kv`LC`37MEO=S?0w=1GC5?+*wYairGab8_Go4=!?=rIS3y~F%tFv
I+Lpx8>S+^g' );
define( 'NONCE_SALT',      '@#m}-
#,Ize]7DX]wT~X0&NvAJ}Gk(0sivns+W0c!sx%8@s[*#G2u1TKN>P%m 2~P' );

/**#@-*/

```

```

/**
 * WordPress Database Table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'wp_';

/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
 * visit the documentation.
 *
 * @link https://wordpress.org/support/article/debugging-in-wordpress/
 */
define( 'WP_DEBUG', false );

/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';

<?php
/**
 * Bootstrap file for setting the ABSPATH constant
 * and loading the wp-config.php file. The wp-config.php
 * file will then load the wp-settings.php file, which
 * will then set up the WordPress environment.
 *
 * If the wp-config.php file is not found then an error
 * will be displayed asking the visitor to set up the
 * wp-config.php file.
 *
 * Will also search for wp-config.php in WordPress' parent
 * directory to allow the WordPress directory to remain
 * untouched.
 *
 * @package WordPress
 */

/** Define ABSPATH as this file's directory */

```

```

if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

error_reporting( E_CORE_ERROR | E_CORE_WARNING | E_COMPILE_ERROR | E_ERROR | E_
WARNING | E_PARSE | E_USER_ERROR | E_USER_WARNING | E_RECOVERABLE_ERROR )
;

/*
 * If wp-config.php exists in the WordPress root, or if it exists in the root and wp-settings.php
 * doesn't, load wp-config.php. The secondary check for wp-settings.php has the added benefit
 * of avoiding cases where the current directory is a nested installation, e.g. / is WordPress(a)
 * and /blog/ is WordPress(b).
 *
 * If neither set of conditions is true, initiate loading the setup process.
 */
if ( file_exists( ABSPATH . 'wp-config.php' ) ) {

    /** The config file resides in ABSPATH */
    require_once ABSPATH . 'wp-config.php';

} elseif ( @file_exists( dirname( ABSPATH ) . '/wp-
config.php' ) && ! @file_exists( dirname( ABSPATH ) . '/wp-settings.php' ) ) {

    /** The config file resides one level above ABSPATH but is not part of another installation */
    require_once dirname( ABSPATH ) . '/wp-config.php';

} else {

    // A config file doesn't exist.

    define( 'WPINC', 'wp-includes' );
    require_once ABSPATH . WPINC . '/load.php';

    // Standardize $_SERVER variables across setups.
    wp_fix_server_vars();

    require_once ABSPATH . WPINC . '/functions.php';

    $path = wp_guess_url() . '/wp-admin/setup-config.php';

    /*
     * We're going to redirect to setup-config.php. While this shouldn't result
     * in an infinite loop, that's a silly thing to assume, don't you think? If
     * we're traveling in circles, our last-ditch effort is "Need more help?"
     */
    if ( false === strpos( $_SERVER['REQUEST_URI'], 'setup-config' ) ) {
        header( 'Location: ' . $path );
        exit;
    }
}

```

```

define( 'WP_CONTENT_DIR', ABSPATH . 'wp-content' );
require_once ABSPATH . WPINC . '/version.php';

wp_check_php_mysql_versions();
wp_load_translations_early();

// Die with an error message.
$die = '<p>' . sprintf(
    /* translators: %s: wp-config.php */
    __( "There doesn't seem to be a %s file. I need this before we can get started." ),
    '<code>wp-config.php</code>'
) . '</p>';
$die .= '<p>' . sprintf(
    /* translators: %s: Documentation URL. */
    __( "Need more help? <a href='%s'>We got it</a>." ),
    __( 'https://wordpress.org/support/article/editing-wp-config-php/' )
) . '</p>';
$die .= '<p>' . sprintf(
    /* translators: %s: wp-config.php */
    __( "You can create a %s file through a web interface, but this doesn't work for all server setups. The safest way is to manually create the file." ),
    '<code>wp-config.php</code>'
) . '</p>';
$die .= '<p><a href="' . $path . '" class="button button-large">' . __( 'Create a Configuration File' ) . '</a></p>';

wp_die( $die, __( 'WordPress &rsquo; Error' ) );
}

```

Login logic

```

<?php
/**
 * WordPress User Page
 *
 * Handles authentication, registering, resetting passwords, forgot password,
 * and other user handling.
 *
 * @package WordPress
 */

/** Make sure that the WordPress bootstrap has run before continuing. */
require __DIR__ . '/wp-load.php';

// Redirect to HTTPS login if forced to use SSL.
if ( force_ssl_admin() && ! is_ssl() ) {
    if ( 0 === strpos( $_SERVER['REQUEST_URI'], 'http' ) ) {
        wp_safe_redirect( set_url_scheme( $_SERVER['REQUEST_URI'], 'https' ) );
        exit;
    } else {
        wp_safe_redirect( 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'] );
    }
}

```

```

    exit;
}
}

/**
 * Output the login page header.
 *
 * @since 2.1.0
 *
 * @global string $error Login error message set by deprecated pluggable wp_login() function
 * or plugins replacing it.
 * @global bool|string $interim_login Whether interim login modal is being displayed. String 'success'
 * upon successful login.
 * @global string $action The action that brought the visitor to the login page.
 *
 * @param string $title Optional. WordPress login Page title to display in the `<title>` element.
 * Default 'Log In'.
 * @param string $message Optional. Message to display in header. Default empty.
 * @param WP_Error $wp_error Optional. The error to pass. Default is a WP_Error instance.
 */
function login_header( $title = 'Log In', $message = '', $wp_error = null ) {
    global $error, $interim_login, $action;

    // Don't index any of these forms.
    add_filter( 'wp_robots', 'wp_robots_sensitive_page' );
    add_action( 'login_head', 'wp_strict_cross_origin_referrer' );

    add_action( 'login_head', 'wp_login_viewport_meta' );

    if ( ! is_wp_error( $wp_error ) ) {
        $wp_error = new WP_Error();
    }

    // Shake it!
    $shake_error_codes = array( 'empty_password', 'empty_email', 'invalid_email', 'invalidcombo', 'empty_username', 'invalid_username', 'incorrect_password', 'retrieve_password_email_failure' );
    /**
     * Filters the error codes array for shaking the login form.
     *
     * @since 3.0.0
     *
     * @param array $shake_error_codes Error codes that shake the login form.
     */
    $shake_error_codes = apply_filters( 'shake_error_codes', $shake_error_codes );

    if ( $shake_error_codes && $wp_error->has_errors() && in_array( $wp_error->get_error_code(), $shake_error_codes, true ) ) {
        add_action( 'login_footer', 'wp_shake_js', 12 );
    }

    $login_title = get_bloginfo( 'name', 'display' );

```

```

/* translators: Login screen title. 1: Login screen name, 2: Network or site name. */
$login_title = sprintf( __( '%1$s &lsquo; %2$s &#8212; WordPress' ), $title, $login_title );

if ( wp_is_recovery_mode() ) {
    /* translators: %s: Login screen title. */
    $login_title = sprintf( __( 'Recovery Mode &#8212; %s' ), $login_title );
}

/**
 * Filters the title tag content for login page.
 *
 * @since 4.9.0
 *
 * @param string $login_title The page title, with extra context added.
 * @param string $title       The original page title.
 */
$login_title = apply_filters( 'login_title', $login_title, $title );

?><!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
<meta http-equiv="Content-
Type" content="<?php bloginfo( 'html_type' ); ?>; charset=<?php bloginfo( 'charset' ); ?>" />
<title><?php echo $login_title; ?></title>
<?php

wp_enqueue_style( 'login' );

/*
 * Remove all stored post data on logging out.
 * This could be added by add_action('login_head'...) like wp_shake_js(),
 * but maybe better if it's not removable by plugins.
 */
if ( 'loggedout' === $wp_error->get_error_code() ) {
    ?>
    <script>if("sessionStorage" in window){try{for(var key in sessionStorage){if(key.indexOf("wp-
autosave-")!=-1){sessionStorage.removeItem(key)}}}catch(e){}};</script>
    <?php
}

/**
 * Enqueue scripts and styles for the login page.
 *
 * @since 3.1.0
 */
do_action( 'login_enqueue_scripts' );

/**
 * Fires in the login page header after scripts are enqueued.
 *

```

```

* @since 2.1.0
*/
do_action( 'login_head' );

$login_header_url = __( 'https://wordpress.org/' );

/**
 * Filters link URL of the header logo above login form.
 *
 * @since 2.1.0
 *
 * @param string $login_header_url Login header logo URL.
 */
$login_header_url = apply_filters( 'login_headerurl', $login_header_url );

$login_header_title = "";

/**
 * Filters the title attribute of the header logo above login form.
 *
 * @since 2.1.0
 * @deprecated 5.2.0 Use {@see 'login_headertext'} instead.
 *
 * @param string $login_header_title Login header logo title attribute.
 */
$login_header_title = apply_filters_deprecated(
    'login_headertitle',
    array( $login_header_title ),
    '5.2.0',
    'login_headertext',
    __( 'Usage of the title attribute on the login logo is not recommended for accessibility reasons. Use the link text instead.' )
);

$login_header_text = empty( $login_header_title ) ? __( 'Powered by WordPress' ) : $login_header_title;

/**
 * Filters the link text of the header logo above the login form.
 *
 * @since 5.2.0
 *
 * @param string $login_header_text The login header logo link text.
 */
$login_header_text = apply_filters( 'login_headertext', $login_header_text );

$classes = array( 'login-action-' . $action, 'wp-core-ui' );

if ( is_rtl() ) {
    $classes[] = 'rtl';
}

```

```

if ( $interim_login ) {
    $classes[] = 'interim-login';

    ?>
    <style type="text/css">html{background-color: transparent;}</style>
    <?php

    if ( 'success' === $interim_login ) {
        $classes[] = 'interim-login-success';
    }
}

$classes[] = ' locale-' . sanitize_html_class( strtolower( str_replace( '_', '-', get_locale() ) ) );

/**
 * Filters the login page body classes.
 *
 * @since 3.5.0
 *
 * @param array $classes An array of body classes.
 * @param string $action The action that brought the visitor to the login page.
 */
$classes = apply_filters( 'login_body_class', $classes, $action );

?>
</head>
<body class="login no-js <?php echo esc_attr( implode( ' ', $classes ) ); ?>">
<script type="text/javascript">
    document.body.className = document.body.className.replace('no-js','js');
</script>
<?php
/**
 * Fires in the login page header after the body tag is opened.
 *
 * @since 4.6.0
 */
do_action( 'login_header' );

?>
<div id="login">
    <h1><a href="<?php echo esc_url( $login_header_url ); ?>"><?php echo $login_header_text; ?></a>
</h1>
<?php
/**
 * Filters the message to display above the login form.
 *
 * @since 2.1.0
 *
 * @param string $message Login message text.
 */

```

```

$message = apply_filters( 'login_message', $message );

if ( ! empty( $message ) ) {
    echo $message . "\n";
}

// In case a plugin uses $error rather than the $wp_errors object.
if ( ! empty( $error ) ) {
    $wp_error->add( 'error', $error );
    unset( $error );
}

if ( $wp_error->has_errors() ) {
    $errors = "";
    $messages = "";

    foreach ( $wp_error->get_error_codes() as $code ) {
        $severity = $wp_error->get_error_data( $code );
        foreach ( $wp_error->get_error_messages( $code ) as $error_message ) {
            if ( 'message' === $severity ) {
                $messages .= ' ' . $error_message . "<br />\n";
            } else {
                $errors .= ' ' . $error_message . "<br />\n";
            }
        }
    }

    if ( ! empty( $errors ) ) {
        /**
         * Filters the error messages displayed above the login form.
         *
         * @since 2.1.0
         *
         * @param string $errors Login error message.
         */
        echo '<div id="login_error">' . apply_filters( 'login_errors', $errors ) . "</div>\n";
    }

    if ( ! empty( $messages ) ) {
        /**
         * Filters instructional messages displayed above the login form.
         *
         * @since 2.5.0
         *
         * @param string $messages Login messages.
         */
        echo '<p class="message">' . apply_filters( 'login_messages', $messages ) . "</p>\n";
    }
}
} // End of login_header().

```

```

/**
 * Outputs the footer for the login page.
 *
 * @since 3.1.0
 *
 * @global bool|string $interim_login Whether interim login modal is being displayed. String 'success'
 *      upon successful login.
 *
 * @param string $input_id Which input to auto-focus.
 */
function login_footer( $input_id = " ) {
    global $interim_login;

    // Don't allow interim logins to navigate away from the page.
    if ( ! $interim_login ) {
        ?>
        <p id="backtoblog">
            <?php
            $html_link = sprintf(
                '<a href="%s">%s</a>',
                esc_url( home_url( '/' ) ),
                sprintf(
                    /* translators: %s: Site title. */
                    _x( '&larr; Go to %s', 'site' ),
                    get_bloginfo( 'title', 'display' )
                )
            );
            /**
             * Filter the "Go to site" link displayed in the login page footer.
             *
             * @since 5.7.0
             *
             * @param string $link HTML link to the home URL of the current site.
             */
            echo apply_filters( 'login_site_html_link', $html_link );
            ?>
        </p>
        <?php

        the_privacy_policy_link( '<div class="privacy-policy-page-link">', '</div>' );
    }

    ?>
</div><?php // End of <div id="login">. ?>

<?php

if ( ! empty( $input_id ) ) {
    ?>
    <script type="text/javascript">
    try{document.getElementById('<?php echo $input_id; ?>').focus();}catch(e){}

```

```

    if(typeof wpOnload=='function')wpOnload();
    </script>
    <?php
}

/**
 * Fires in the login page footer.
 *
 * @since 3.1.0
 */
do_action( 'login_footer' );

?>
<div class="clear"></div>
</body>
</html>
<?php
}

/**
 * Outputs the JavaScript to handle the form shaking on the login page.
 *
 * @since 3.0.0
 */
function wp_shake_js() {
    ?>
    <script type="text/javascript">
    document.querySelector('form').classList.add('shake');
    </script>
    <?php
}

/**
 * Outputs the viewport meta tag for the login page.
 *
 * @since 3.7.0
 */
function wp_login_viewport_meta() {
    ?>
    <meta name="viewport" content="width=device-width" />
    <?php
}

//
// Main.
//

$action = isset( $_REQUEST['action'] ) ? $_REQUEST['action'] : 'login';
$errors = new WP_Error();

if ( isset( $_GET['key'] ) ) {

```

```

    $action = 'resetpass';
}

if ( isset( $_GET['checkemail'] ) ) {
    $action = 'checkemail';
}

$default_actions = array(
    'confirm_admin_email',
    'postpass',
    'logout',
    'lostpassword',
    'retrievepassword',
    'resetpass',
    'rp',
    'register',
    'checkemail',
    'confirmation',
    'login',
    WP_Recovery_Mode_Link_Service::LOGIN_ACTION_ENTERED,
);

// Validate action so as to default to the login screen.
if ( ! in_array( $action, $default_actions, true ) && false === has_filter( 'login_form_' . $action ) ) {
    $action = 'login';
}

nocache_headers();

header( 'Content-Type: ' . get_bloginfo( 'html_type' ) . '; charset=' . get_bloginfo( 'charset' ) );

if ( defined( 'RELOCATE' ) && RELOCATE ) { // Move flag is set.
    if ( isset( $_SERVER['PATH_INFO'] ) && ( $_SERVER['PATH_INFO'] !== $_SERVER['PHP_SELF'] ) ) {
        $_SERVER['PHP_SELF'] = str_replace( $_SERVER['PATH_INFO'], "", $_SERVER['PHP_SELF'] );
    }

    $url = dirname( set_url_scheme( 'http://' . $_SERVER['HTTP_HOST'] . $_SERVER['PHP_SELF'] ) );

    if ( get_option( 'siteurl' ) !== $url ) {
        update_option( 'siteurl', $url );
    }
}

// Set a cookie now to see if they are supported by the browser.
$secure = ( 'https' === parse_url( wp_login_url(), PHP_URL_SCHEME ) );
setcookie( TEST_COOKIE, 'WP Cookie check', 0, COOKIEPATH, COOKIE_DOMAIN, $secure );

if ( SITECOOKIEPATH !== COOKIEPATH ) {

```

```

    setcookie( TEST_COOKIE, 'WP Cookie check', 0, SITECOOKIEPATH, COOKIE_DOMAIN, $secure );
}

```

```

/**

```

```

 * Fires when the login form is initialized.

```

```

 *

```

```

 * @since 3.2.0

```

```

 */

```

```

do_action( 'login_init' );

```

```

/**

```

```

 * Fires before a specified login form action.

```

```

 *

```

```

 * The dynamic portion of the hook name, `$action`, refers to the action

```

```

 * that brought the visitor to the login form. Actions include 'postpass',

```

```

 * 'logout', 'lostpassword', etc.

```

```

 *

```

```

 * @since 2.8.0

```

```

 */

```

```

do_action( "login_form_{$action}" );

```

```

$http_post = ( 'POST' === $_SERVER['REQUEST_METHOD'] );

```

```

$interim_login = isset( $_REQUEST['interim-login'] );

```

```

/**

```

```

 * Filters the separator used between login form navigation links.

```

```

 *

```

```

 * @since 4.9.0

```

```

 *

```

```

 * @param string $login_link_separator The separator used between login form navigation links.

```

```

 */

```

```

$login_link_separator = apply_filters( 'login_link_separator', ' | ' );

```

```

switch ( $action ) {

```

```

    case 'confirm_admin_email':

```

```

        /*

```

```

         * Note that `is_user_logged_in()` will return false immediately after logging in

```

```

         * as the current user is not set, see wp-includes/pluggable.php.

```

```

         * However this action runs on a redirect after logging in.

```

```

        */

```

```

        if ( ! is_user_logged_in() ) {

```

```

            wp_safe_redirect( wp_login_url() );

```

```

            exit;

```

```

        }

```

```

        if ( ! empty( $_REQUEST['redirect_to'] ) ) {

```

```

            $redirect_to = $_REQUEST['redirect_to'];

```

```

        } else {

```

```

            $redirect_to = admin_url();

```

```

}

if ( current_user_can( 'manage_options' ) ) {
    $admin_email = get_option( 'admin_email' );
} else {
    wp_safe_redirect( $redirect_to );
    exit;
}

/**
 * Filters the interval for dismissing the admin email confirmation screen.
 *
 * If `0` (zero) is returned, the "Remind me later" link will not be displayed.
 *
 * @since 5.3.1
 *
 * @param int $interval Interval time (in seconds). Default is 3 days.
 */
$remind_interval = (int) apply_filters( 'admin_email_remind_interval', 3 * DAY_IN_SECONDS );

if ( ! empty( $_GET['remind_me_later'] ) ) {
    if ( ! wp_verify_nonce( $_GET['remind_me_later'], 'remind_me_later_nonce' ) ) {
        wp_safe_redirect( wp_login_url() );
        exit;
    }

    if ( $remind_interval > 0 ) {
        update_option( 'admin_email_lifespan', time() + $remind_interval );
    }

    $redirect_to = add_query_arg( 'admin_email_remind_later', 1, $redirect_to );
    wp_safe_redirect( $redirect_to );
    exit;
}

if ( ! empty( $_POST['correct-admin-email'] ) ) {
    if ( ! check_admin_referer( 'confirm_admin_email', 'confirm_admin_email_nonce' ) ) {
        wp_safe_redirect( wp_login_url() );
        exit;
    }
}

/**
 * Filters the interval for redirecting the user to the admin email confirmation screen.
 *
 * If `0` (zero) is returned, the user will not be redirected.
 *
 * @since 5.3.0
 *
 * @param int $interval Interval time (in seconds). Default is 6 months.
 */

```

```

$admin_email_check_interval = (int) apply_filters( 'admin_email_check_interval', 6 * MONTH_IN_SECONDS );

if ( $admin_email_check_interval > 0 ) {
    update_option( 'admin_email_lifespan', time() + $admin_email_check_interval );
}

wp_safe_redirect( $redirect_to );
exit;
}

login_header( __( 'Confirm your administration email' ), "", $errors );

/**
 * Fires before the admin email confirm form.
 *
 * @since 5.3.0
 *
 * @param WP_Error $errors A `WP_Error` object containing any errors generated by using invalid
 *                          credentials. Note that the error object may not contain any errors.
 */
do_action( 'admin_email_confirm', $errors );

?>

<form class="admin-email-confirm-form" name="admin-email-confirm-form" action="<?php echo esc_url( site_url( 'wp-login.php?action=confirm_admin_email', 'login_post' ) ); ?>" method="post">
    <?php
    /**
     * Fires inside the admin-email-confirm-form form tags, before the hidden fields.
     *
     * @since 5.3.0
     */
    do_action( 'admin_email_confirm_form' );

    wp_nonce_field( 'confirm_admin_email', 'confirm_admin_email_nonce' );

?>
    <input type="hidden" name="redirect_to" value="<?php echo esc_attr( $redirect_to ); ?>" />

    <h1 class="admin-email__heading">
        <?php _e( 'Administration email verification' ); ?>
    </h1>
    <p class="admin-email__details">
        <?php _e( 'Please verify that the <strong>administration email</strong> for this website is still correct.' ); ?>
    <?php

    /* translators: URL to the WordPress help section about admin email. */

```

```
$admin_email_help_url = __( 'https://wordpress.org/support/article/settings-general-screen/#email-address' );
```

```
/* translators: Accessibility text. */
$accessibility_text = sprintf( '<span class="screen-reader-text"> %s</span>', __( '(opens in a new tab)' ) );
```

```
printf(
    '<a href="%s" rel="noopener" target="_blank">%s%s</a>',
    esc_url( $admin_email_help_url ),
    __( 'Why is this important?' ),
    $accessibility_text
);
```

```
?>
</p>
<p class="admin-email__details">
<?php
```

```
printf(
    /* translators: %s: Admin email address. */
    __( 'Current administration email: %s' ),
    '<strong>' . esc_html( $admin_email ) . '</strong>'
);
```

```
?>
</p>
<p class="admin-email__details">
<?php _e( 'This email may be different from your personal email address.' ); ?>
</p>
```

```
<div class="admin-email__actions">
<div class="admin-email__actions-primary">
<?php
```

```
$change_link = admin_url( 'options-general.php' );
$change_link = add_query_arg( 'highlight', 'confirm_admin_email', $change_link );
```

```
?>
<a class="button button-
large" href="<?php echo esc_url( $change_link ); ?>"><?php _e( 'Update' ); ?></a>
<input type="submit" name="correct-admin-email" id="correct-admin-
email" class="button button-primary button-
large" value="<?php esc_attr_e( 'The email is correct' ); ?>" />
</div>
```

```
<?php if ( $remind_interval > 0 ) : ?>
<div class="admin-email__actions-secondary">
<?php
```

```
$remind_me_link = wp_login_url( $redirect_to );
$remind_me_link = add_query_arg(
```

```

        array(
            'action'      => 'confirm_admin_email',
            'remind_me_later' => wp_create_nonce( 'remind_me_later_nonce' ),
        ),
        $remind_me_link
    );

    ?>
    <a href="<?php echo esc_url( $remind_me_link ); ?>"><?php _e( 'Remind me later' ); ?></a>
</div>
<?php endif; ?>
</div>
</form>

<?php

login_footer();
break;

case 'postpass':
    if ( ! array_key_exists( 'post_password', $_POST ) ) {
        wp_safe_redirect( wp_get_referer() );
        exit;
    }

    require_once ABSPATH . WPINC . '/class-phpass.php';
    $hasher = new PasswordHash( 8, true );

    /**
     * Filters the life span of the post password cookie.
     *
     * By default, the cookie expires 10 days from creation. To turn this
     * into a session cookie, return 0.
     *
     * @since 3.7.0
     *
     * @param int $expires The expiry time, as passed to setcookie().
     */
    $expire = apply_filters( 'post_password_expires', time() + 10 * DAY_IN_SECONDS );
    $referer = wp_get_referer();

    if ( $referer ) {
        $secure = ( 'https' === parse_url( $referer, PHP_URL_SCHEME ) );
    } else {
        $secure = false;
    }

    setcookie( 'wp-postpass_' . COOKIEHASH, $hasher->HashPassword( wp_unslash( $_POST['post_password'] ) ), $expire, COOKIEPATH, COOKIE_DO
MAIN, $secure );

```

```

wp_safe_redirect( wp_get_referer() );
exit;

case 'logout':
    check_admin_referer( 'log-out' );

    $user = wp_get_current_user();

    wp_logout();

    if ( ! empty( $_REQUEST['redirect_to'] ) ) {
        $redirect_to = $_REQUEST['redirect_to'];
        $requested_redirect_to = $redirect_to;
    } else {
        $redirect_to = add_query_arg(
            array(
                'loggedout' => 'true',
                'wp_lang' => get_user_locale( $user ),
            ),
            wp_login_url()
        );

        $requested_redirect_to = "";
    }

    /**
     * Filters the log out redirect URL.
     *
     * @since 4.2.0
     *
     * @param string $redirect_to The redirect destination URL.
     * @param string $requested_redirect_to The requested redirect destination URL passed as a parameter.
     * @param WP_User $user The WP_User object for the user that's logging out.
     */
    $redirect_to = apply_filters( 'logout_redirect', $redirect_to, $requested_redirect_to, $user );

    wp_safe_redirect( $redirect_to );
    exit;

case 'lostpassword':
case 'retrievepassword':
    if ( $http_post ) {
        $errors = retrieve_password();

        if ( ! is_wp_error( $errors ) ) {
            $redirect_to = ! empty( $_REQUEST['redirect_to'] ) ? $_REQUEST['redirect_to'] : 'wp-login.php?checkemail=confirm';
            wp_safe_redirect( $redirect_to );
            exit;
        }
    }

```

```

    }

    if ( isset( $_GET['error'] ) ) {
        if ( 'invalidkey' === $_GET['error'] ) {
            $errors-
>add( 'invalidkey', __( 'Your password reset link appears to be invalid. Please request a new link below
.' ) );
        } elseif ( 'expiredkey' === $_GET['error'] ) {
            $errors-
>add( 'expiredkey', __( 'Your password reset link has expired. Please request a new link below.' ) );
        }
    }

    $lostpassword_redirect = ! empty( $_REQUEST['redirect_to'] ) ? $_REQUEST['redirect_to'] : "";
    /**
     * Filters the URL redirected to after submitting the lostpassword/retrievepassword form.
     *
     * @since 3.0.0
     *
     * @param string $lostpassword_redirect The redirect destination URL.
     */
    $redirect_to = apply_filters( 'lostpassword_redirect', $lostpassword_redirect );

    /**
     * Fires before the lost password form.
     *
     * @since 1.5.1
     * @since 5.1.0 Added the `$errors` parameter.
     *
     * @param WP_Error $errors A `WP_Error` object containing any errors generated by using invalid
     *                        credentials. Note that the error object may not contain any errors.
     */
    do_action( 'lost_password', $errors );

    login_header( __( 'Lost Password' ), '<p class="message">' . __( 'Please enter your username or email address. You will receive an email message with instructions on how to reset your password.' ) . '</p>', $errors );

    $user_login = "";

    if ( isset( $_POST['user_login'] ) && is_string( $_POST['user_login'] ) ) {
        $user_login = wp_unslash( $_POST['user_login'] );
    }

    ?>

    <form name="lostpasswordform" id="lostpasswordform" action="<?php echo esc_url( network_site_url( 'wp-login.php?action=lostpassword', 'login_post' ) ); ?>" method="post">
        <p>
            <label for="user_login"><?php _e( 'Username or Email Address' ); ?></label>

```

```

        <input type="text" name="user_login" id="user_login" class="input" value="<?php echo esc_attr
( $user_login ); ?>" size="20" autocapitalize="off" />
    </p>
    <?php

    /**
     * Fires inside the lostpassword form tags, before the hidden fields.
     *
     * @since 2.1.0
     */
    do_action( 'lostpassword_form' );

    ?>
    <input type="hidden" name="redirect_to" value="<?php echo esc_attr( $redirect_to ); ?>" />
    <p class="submit">
        <input type="submit" name="wp-submit" id="wp-submit" class="button button-primary button-
large" value="<?php esc_attr_e( 'Get New Password' ); ?>" />
    </p>
</form>

<p id="nav">
    <a href="<?php echo esc_url( wp_login_url() ); ?>"><?php _e( 'Log in' ); ?></a>
    <?php

    if ( get_option( 'users_can_register' ) ) {
        $registration_url = sprintf( '<a href="%s">%s</a>', esc_url( wp_registration_url() ), __( 'Register'
) );

        echo esc_html( $login_link_separator );

        /** This filter is documented in wp-includes/general-template.php */
        echo apply_filters( 'register', $registration_url );
    }

    ?>
</p>
<?php

login_footer( 'user_login' );
break;

case 'resetpass':
case 'rp':
    list( $rp_path ) = explode( '?', wp_unslash( $_SERVER['REQUEST_URI'] ) );
    $rp_cookie      = 'wp-resetpass-' . COOKIEHASH;

    if ( isset( $_GET['key'] ) ) {
        $value = sprintf( '%s:%s', wp_unslash( $_GET['login'] ), wp_unslash( $_GET['key'] ) );
        setcookie( $rp_cookie, $value, 0, $rp_path, COOKIE_DOMAIN, is_ssl(), true );

        wp_safe_redirect( remove_query_arg( array( 'key', 'login' ) ) );
    }

```

```

    exit;
}

if ( isset( $_COOKIE[ $rp_cookie ] ) && 0 < strpos( $_COOKIE[ $rp_cookie ], ':' ) ) {
    list( $rp_login, $rp_key ) = explode( ':', wp_unslash( $_COOKIE[ $rp_cookie ] ), 2 );

    $user = check_password_reset_key( $rp_key, $rp_login );

    if ( isset( $_POST['pass1'] ) && ! hash_equals( $rp_key, $_POST['rp_key'] ) ) {
        $user = false;
    }
} else {
    $user = false;
}

if ( ! $user || is_wp_error( $user ) ) {
    setcookie( $rp_cookie, '', time() - YEAR_IN_SECONDS, $rp_path, COOKIE_DOMAIN, is_ssl(),
true );

    if ( $user && $user->get_error_code() === 'expired_key' ) {
        wp_redirect( site_url( 'wp-login.php?action=lostpassword&error=expiredkey' ) );
    } else {
        wp_redirect( site_url( 'wp-login.php?action=lostpassword&error=invalidkey' ) );
    }

    exit;
}

$errors = new WP_Error();

if ( isset( $_POST['pass1'] ) && $_POST['pass1'] !== $_POST['pass2'] ) {
    $errors->add( 'password_reset_mismatch', __( 'The passwords do not match.' ) );
}

/**
 * Fires before the password reset procedure is validated.
 *
 * @since 3.5.0
 *
 * @param WP_Error $errors WP Error object.
 * @param WP_User|WP_Error $user WP_User object if the login and reset key match. WP_Error
object otherwise.
 */
do_action( 'validate_password_reset', $errors, $user );

if ( ( ! $errors->has_errors() ) && isset( $_POST['pass1'] ) && ! empty( $_POST['pass1'] ) ) {
    reset_password( $user, $_POST['pass1'] );
    setcookie( $rp_cookie, '', time() - YEAR_IN_SECONDS, $rp_path, COOKIE_DOMAIN, is_ssl(),
true );
}

```

```

    login_header( __( 'Password Reset' ), '<p class="message reset-
pass">' . __( 'Your password has been reset.' ) . ' <a href="' . esc_url( wp_login_url() ) . '">' . __( 'Log i
n' ) . '</a></p>' );
    login_footer();
    exit;
}

wp_enqueue_script( 'utils' );
wp_enqueue_script( 'user-profile' );

login_header( __( 'Reset Password' ), '<p class="message reset-
pass">' . __( 'Enter your new password below or generate one.' ) . '</p>', $errors );

?>
<form name="resetpassform" id="resetpassform" action="<?php echo esc_url( network_site_url( 'w
p-login.php?action=resetpass', 'login_post' ) ); ?>" method="post" autocomplete="off">
    <input type="hidden" id="user_login" value="<?php echo esc_attr( $rp_login ); ?>" autocomplete
="off" />

    <div class="user-pass1-wrap">
        <p>
            <label for="pass1"><?php _e( 'New password' ); ?></label>
        </p>

        <div class="wp-pwd">
            <input type="password" data-reveal="1" data-
pw="<?php echo esc_attr( wp_generate_password( 16 ) ); ?>" name="pass1" id="pass1" class="input
password-input" size="24" value="" autocomplete="off" aria-describedby="pass-strength-result" />

            <button type="button" class="button button-secondary wp-hide-pw hide-if-no-js" data-
toggle="0" aria-label="<?php esc_attr_e( 'Hide password' ); ?>">
                <span class="dashicons dashicons-hidden" aria-hidden="true"></span>
            </button>
            <div id="pass-strength-result" class="hide-if-no-js" aria-
live="polite"><?php _e( 'Strength indicator' ); ?></div>
        </div>
        <div class="pw-weak">
            <input type="checkbox" name="pw_weak" id="pw_weak" class="pw-checkbox" />
            <label for="pw_weak"><?php _e( 'Confirm use of weak password' ); ?></label>
        </div>
    </div>

    <p class="user-pass2-wrap">
        <label for="pass2"><?php _e( 'Confirm new password' ); ?></label>
        <input type="password" name="pass2" id="pass2" class="input" size="20" value="" autocomplet
e="off" />
    </p>

    <p class="description indicator-hint"><?php echo wp_get_password_hint(); ?></p>
    <br class="clear" />

```

```

<?php

/**
 * Fires following the 'Strength indicator' meter in the user password reset form.
 *
 * @since 3.9.0
 *
 * @param WP_User $user User object of the user whose password is being reset.
 */
do_action( 'resetpass_form', $user );

?>
<input type="hidden" name="rp_key" value="<?php echo esc_attr( $rp_key ); ?>" />
<p class="submit reset-pass-submit">
    <button type="button" class="button wp-generate-pw hide-if-no-js" aria-
expanded="true"><?php _e( 'Generate Password' ); ?></button>
    <input type="submit" name="wp-submit" id="wp-submit" class="button button-primary button-
large" value="<?php esc_attr_e( 'Save Password' ); ?>" />
</p>
</form>

<p id="nav">
    <a href="<?php echo esc_url( wp_login_url() ); ?>"><?php _e( 'Log in' ); ?></a>
<?php

if ( get_option( 'users_can_register' ) ) {
    $registration_url = sprintf( '<a href="%s">%s</a>', esc_url( wp_registration_url() ), __( 'Register'
) );

    echo esc_html( $login_link_separator );

    /** This filter is documented in wp-includes/general-template.php */
    echo apply_filters( 'register', $registration_url );
}

?>
</p>
<?php

login_footer( 'user_pass' );
break;

case 'register':
    if ( is_multisite() ) {
        /**
         * Filters the Multisite sign up URL.
         *
         * @since 3.0.0
         *
         * @param string $sign_up_url The sign up URL.
         */
    }

```

```

wp_redirect( apply_filters( 'wp_signup_location', network_site_url( 'wp-signup.php' ) ) );
exit;
}

if ( ! get_option( 'users_can_register' ) ) {
    wp_redirect( site_url( 'wp-login.php?registration=disabled' ) );
    exit;
}

$user_login = "";
$user_email = "";

if ( $http_post ) {
    if ( isset( $_POST['user_login'] ) && is_string( $_POST['user_login'] ) ) {
        $user_login = wp_unslash( $_POST['user_login'] );
    }

    if ( isset( $_POST['user_email'] ) && is_string( $_POST['user_email'] ) ) {
        $user_email = wp_unslash( $_POST['user_email'] );
    }

    $errors = register_new_user( $user_login, $user_email );

    if ( ! is_wp_error( $errors ) ) {
        $redirect_to = ! empty( $_POST['redirect_to'] ) ? $_POST['redirect_to'] : 'wp-
login.php?checkemail=registered';
        wp_safe_redirect( $redirect_to );
        exit;
    }
}

$registration_redirect = ! empty( $_REQUEST['redirect_to'] ) ? $_REQUEST['redirect_to'] : "";

/**
 * Filters the registration redirect URL.
 *
 * @since 3.0.0
 *
 * @param string $registration_redirect The redirect destination URL.
 */
$redirect_to = apply_filters( 'registration_redirect', $registration_redirect );

login_header( __( 'Registration Form' ), '<p class="message register">' . __( 'Register For This Site'
) . '</p>', $errors );

?>
<form name="registerform" id="registerform" action="<?php echo esc_url( site_url( 'wp-
login.php?action=register', 'login_post' ) ); ?>" method="post" novalidate="novalidate">
    <p>
        <label for="user_login"><?php _e( 'Username' ); ?></label>

```

```

        <input type="text" name="user_login" id="user_login" class="input" value="<?php echo esc_attr
( wp_unslash( $user_login ) ); ?>" size="20" autocapitalize="off" />
    </p>
    <p>
        <label for="user_email"><?php _e( 'Email' ); ?></label>
        <input type="email" name="user_email" id="user_email" class="input" value="<?php echo esc_a
ttr( wp_unslash( $user_email ) ); ?>" size="25" />
    </p>
    <?php

    /**
     * Fires following the 'Email' field in the user registration form.
     *
     * @since 2.1.0
     */
    do_action( 'register_form' );

    ?>
    <p id="reg_passmail">
        <?php _e( 'Registration confirmation will be emailed to you.' ); ?>
    </p>
    <br class="clear" />
    <input type="hidden" name="redirect_to" value="<?php echo esc_attr( $redirect_to ); ?>" />
    <p class="submit">
        <input type="submit" name="wp-submit" id="wp-submit" class="button button-primary button-
large" value="<?php esc_attr_e( 'Register' ); ?>" />
    </p>
</form>

<p id="nav">
    <a href="<?php echo esc_url( wp_login_url() ); ?>"><?php _e( 'Log in' ); ?></a>
    <?php echo esc_html( $login_link_separator ); ?>
    <a href="<?php echo esc_url( wp_lostpassword_url() ); ?>"><?php _e( 'Lost your password?' ); ?>
</a>
</p>
<?php

login_footer( 'user_login' );
break;

case 'checkemail':
    $redirect_to = admin_url();
    $errors      = new WP_Error();

    if ( 'confirm' === $_GET['checkemail'] ) {
        $errors->add(
            'confirm',
            sprintf(
                /* translators: %s: Link to the login page. */
                __( 'Check your email for the confirmation link, then visit the <a href="%s">login page</a>.' ),
                wp_login_url()
            )
        );
    }

```

```

    ),
    'message'
  );
} elseif ( 'registered' === $_GET['checkemail'] ) {
  $errors->add(
    'registered',
    sprintf(
      /* translators: %s: Link to the login page. */
      __( 'Registration complete. Please check your email, then visit the <a href="%s">login page</a>' ),
      wp_login_url()
    ),
    'message'
  );
}

/** This action is documented in wp-login.php */
$errors = apply_filters( 'wp_login_errors', $errors, $redirect_to );

login_header( __( 'Check your email' ), '', $errors );
login_footer();
break;

case 'confirmation':
  if ( ! isset( $_GET['request_id'] ) ) {
    wp_die( __( 'Missing request ID.' ) );
  }

  if ( ! isset( $_GET['confirm_key'] ) ) {
    wp_die( __( 'Missing confirm key.' ) );
  }

  $request_id = (int) $_GET['request_id'];
  $key        = sanitize_text_field( wp_unslash( $_GET['confirm_key'] ) );
  $result     = wp_validate_user_request_key( $request_id, $key );

  if ( is_wp_error( $result ) ) {
    wp_die( $result );
  }

  /**
   * Fires an action hook when the account action has been confirmed by the user.
   *
   * Using this you can assume the user has agreed to perform the action by
   * clicking on the link in the confirmation email.
   *
   * After firing this action hook the page will redirect to wp-login a callback
   * redirects or exits first.
   *
   * @since 4.9.6
   */

```

```

* @param int $request_id Request ID.
*/
do_action( 'user_request_action_confirmed', $request_id );

$message = _wp_privacy_account_request_confirmed_message( $request_id );

login_header( __( 'User action confirmed.' ), $message );
login_footer();
exit;

case 'login':
default:
    $secure_cookie = "";
    $customize_login = isset( $_REQUEST['customize-login'] );

    if ( $customize_login ) {
        wp_enqueue_script( 'customize-base' );
    }

    // If the user wants SSL but the session is not SSL, force a secure cookie.
    if ( ! empty( $_POST['log'] ) && ! force_ssl_admin() ) {
        $user_name = sanitize_user( wp_unslash( $_POST['log'] ) );
        $user = get_user_by( 'login', $user_name );

        if ( ! $user && strpos( $user_name, '@' ) ) {
            $user = get_user_by( 'email', $user_name );
        }

        if ( $user ) {
            if ( get_user_option( 'use_ssl', $user->ID ) ) {
                $secure_cookie = true;
                force_ssl_admin( true );
            }
        }
    }

    if ( isset( $_REQUEST['redirect_to'] ) ) {
        $redirect_to = $_REQUEST['redirect_to'];
        // Redirect to HTTPS if user wants SSL.
        if ( $secure_cookie && false !== strpos( $redirect_to, 'wp-admin' ) ) {
            $redirect_to = preg_replace( '^http://', 'https://', $redirect_to );
        }
    } else {
        $redirect_to = admin_url();
    }

    $reauth = empty( $_REQUEST['reauth'] ) ? false : true;

    $user = wp_signon( array(), $secure_cookie );

    if ( empty( $_COOKIE[ LOGGED_IN_COOKIE ] ) ) {

```

```

if ( headers_sent() ) {
    $user = new WP_Error(
        'test_cookie',
        sprintf(
            /* translators: 1: Browser cookie documentation URL, 2: Support forums URL. */
            __( '<strong>Error</strong>: Cookies are blocked due to unexpected output. For help, please s
ee <a href="%1$s">this documentation</a> or try the <a href="%2$s">support forums</a>.' ),
            __( 'https://wordpress.org/support/article/cookies/' ),
            __( 'https://wordpress.org/support/forums/' )
        )
    );
} elseif ( isset( $_POST['testcookie'] ) && empty( $_COOKIE[ TEST_COOKIE ] ) ) {
    // If cookies are disabled, we can't log in even with a valid user and password.
    $user = new WP_Error(
        'test_cookie',
        sprintf(
            /* translators: %s: Browser cookie documentation URL. */
            __( '<strong>Error</strong>: Cookies are blocked or not supported by your browser. You must
<a href="%s">enable cookies</a> to use WordPress.' ),
            __( 'https://wordpress.org/support/article/cookies/#enable-cookies-in-your-browser' )
        )
    );
}
}

$requested_redirect_to = isset( $_REQUEST['redirect_to'] ) ? $_REQUEST['redirect_to'] : '';
/**
 * Filters the login redirect URL.
 *
 * @since 3.0.0
 *
 * @param string      $redirect_to      The redirect destination URL.
 * @param string      $requested_redirect_to The requested redirect destination URL passed as a
parameter.
 * @param WP_User|WP_Error $user        WP_User object if login was successful, WP_Error
object otherwise.
 */
$redirect_to = apply_filters( 'login_redirect', $redirect_to, $requested_redirect_to, $user );

if ( ! is_wp_error( $user ) && ! $reauth ) {
    if ( $interim_login ) {
        $message = '<p class="message">' . __( 'You have logged in successfully.' ) . '</p>';
        $interim_login = 'success';
        login_header( "", $message );

        ?>
    </div>
    <?php

    /** This action is documented in wp-login.php */
    do_action( 'login_footer' );

```

```

if ( $customize_login ) {
    ?>
    <script type="text/javascript">setTimeout( function(){ new wp.customize.Messenger({ url: '<?p
hp echo wp_customize_url(); ?>', channel: 'login' }).send('login' ) }, 1000 );</script>
    <?php
    }

    ?>
    </body></html>
    <?php

    exit;
}

// Check if it is time to add a redirect to the admin email confirmation screen.
if ( is_a( $user, 'WP_User' ) && $user->exists() && $user->has_cap( 'manage_options' ) ) {
    $admin_email_lifespan = (int) get_option( 'admin_email_lifespan' );

    // If `0` (or anything "falsey" as it is cast to int) is returned, the user will not be redirected
    // to the admin email confirmation screen.
    /** This filter is documented in wp-login.php */
    $admin_email_check_interval = (int) apply_filters( 'admin_email_check_interval', 6 * MONTH_I
N_SECONDS );

    if ( $admin_email_check_interval > 0 && time() > $admin_email_lifespan ) {
        $redirect_to = add_query_arg(
            array(
                'action' => 'confirm_admin_email',
                'wp_lang' => get_user_locale( $user ),
            ),
            wp_login_url( $redirect_to )
        );
    }
}

if ( ( empty( $redirect_to ) || 'wp-admin/' === $redirect_to || admin_url() === $redirect_to ) ) {
    // If the user doesn't belong to a blog, send them to user admin. If the user can't edit posts, send th
em to their profile.
    if ( is_multisite() && ! get_active_blog_for_user( $user->ID ) && ! is_super_admin( $user-
>ID ) ) {
        $redirect_to = user_admin_url();
    } elseif ( is_multisite() && ! $user->has_cap( 'read' ) ) {
        $redirect_to = get_dashboard_url( $user->ID );
    } elseif ( ! $user->has_cap( 'edit_posts' ) ) {
        $redirect_to = $user->has_cap( 'read' ) ? admin_url( 'profile.php' ) : home_url();
    }

    wp_redirect( $redirect_to );
    exit;
}

```

```

    wp_safe_redirect( $redirect_to );
    exit;
}

$errors = $user;
// Clear errors if loggedout is set.
if ( ! empty( $_GET['loggedout'] ) || $reauth ) {
    $errors = new WP_Error();
}

if ( empty( $_POST ) && $errors-
>get_error_codes() === array( 'empty_username', 'empty_password' ) ) {
    $errors = new WP_Error( "", "" );
}

if ( $interim_login ) {
    if ( ! $errors->has_errors() ) {
        $errors-
>add( 'expired', __( 'Your session has expired. Please log in to continue where you left off.' ), 'message'
);
    }
} else {
    // Some parts of this script use the main login form to display a message.
    if ( isset( $_GET['loggedout'] ) && $_GET['loggedout'] ) {
        $errors->add( 'loggedout', __( 'You are now logged out.' ), 'message' );
    } elseif ( isset( $_GET['registration'] ) && 'disabled' === $_GET['registration'] ) {
        $errors->add( 'registerdisabled', __( 'User registration is currently not allowed.' ) );
    } elseif ( strpos( $redirect_to, 'about.php?updated' ) ) {
        $errors-
>add( 'updated', __( '<strong>You have successfully updated WordPress!</strong> Please log back in t
o see what&#8217;s new.' ), 'message' );
    } elseif ( WP_Recovery_Mode_Link_Service::LOGIN_ACTION_ENTERED === $action ) {
        $errors-
>add( 'enter_recovery_mode', __( 'Recovery Mode Initialized. Please log in to continue.' ), 'message' );
    } elseif ( isset( $_GET['redirect_to'] ) && false !== strpos( $_GET['redirect_to'], 'wp-
admin/authorize-application.php' ) ) {
        $query_component = wp_parse_url( $_GET['redirect_to'], PHP_URL_QUERY );
        parse_str( $query_component, $query );

        if ( ! empty( $query['app_name'] ) ) {
            /* translators: 1: Website name, 2: Application name. */
            $message = sprintf( 'Please log in to %1$s to authorize %2$s to connect to your account.', get_b
loginfo( 'name', 'display' ), '<strong>'. esc_html( $query['app_name'] ) . '</strong>' );
        } else {
            /* translators: %s: Website name. */
            $message = sprintf( 'Please log in to %s to proceed with authorization.', get_bloginfo( 'name', 'di
splay' ) );
        }

        $errors->add( 'authorize_application', $message, 'message' );
    }
}

```

```

    }
}

/**
 * Filters the login page errors.
 *
 * @since 3.6.0
 *
 * @param WP_Error $errors WP Error object.
 * @param string $redirect_to Redirect destination URL.
 */
$errors = apply_filters( 'wp_login_errors', $errors, $redirect_to );

// Clear any stale cookies.
if ( $reauth ) {
    wp_clear_auth_cookie();
}

login_header( __( 'Log In' ), "", $errors );

if ( isset( $_POST['log'] ) ) {
    $user_login = ( 'incorrect_password' === $errors-
>get_error_code() || 'empty_password' === $errors-
>get_error_code() ) ? esc_attr( wp_unslash( $_POST['log'] ) ) : "";
}

$rememberme = ! empty( $_POST['rememberme'] );

if ( $errors->has_errors() ) {
    $aria_describedby_error = ' aria-describedby="login_error"';
} else {
    $aria_describedby_error = "";
}

wp_enqueue_script( 'user-profile' );
?>

<form name="loginform" id="loginform" action="<?php echo esc_url( site_url( 'wp-
login.php', 'login_post' ) ); ?>" method="post">
    <p>
        <label for="user_login"><?php _e( 'Username or Email Address' ); ?></label>
        <input type="text" name="log" id="user_login"<?php echo $aria_describedby_error; ?> class="in
put" value="<?php echo esc_attr( $user_login ); ?>" size="20" autocapitalize="off" />
    </p>

    <div class="user-pass-wrap">
        <label for="user_pass"><?php _e( 'Password' ); ?></label>
        <div class="wp-pwd">
            <input type="password" name="pwd" id="user_pass"<?php echo $aria_describedby_error; ?> cl
ass="input password-input" value="" size="20" />

```

```

        <button type="button" class="button button-secondary wp-hide-pw hide-if-no-js" data-
toggle="0" aria-label="<?php esc_attr_e( 'Show password' ); ?>">
        <span class="dashicons dashicons-visibility" aria-hidden="true"></span>
    </button>
</div>
</div>
<?php

/**
 * Fires following the 'Password' field in the login form.
 *
 * @since 2.1.0
 */
do_action( 'login_form' );

?>
<p class="forgetmenot"><input name="rememberme" type="checkbox" id="rememberme" value=
"forever" <?php checked( $rememberme ); ?> /> <label for="rememberme"><?php esc_html_e( 'Rem
ember Me' ); ?></label></p>
<p class="submit">
    <input type="submit" name="wp-submit" id="wp-submit" class="button button-primary button-
large" value="<?php esc_attr_e( 'Log In' ); ?>" />
<?php

if ( $interim_login ) {
    ?>
    <input type="hidden" name="interim-login" value="1" />
    <?php
} else {
    ?>
    <input type="hidden" name="redirect_to" value="<?php echo esc_attr( $redirect_to ); ?>" />
    <?php
}

if ( $customize_login ) {
    ?>
    <input type="hidden" name="customize-login" value="1" />
    <?php
}

?>
<input type="hidden" name="testcookie" value="1" />
</p>
</form>

<?php

if ( ! $interim_login ) {
    ?>
    <p id="nav">
        <?php

```

```

if ( get_option( 'users_can_register' ) ) {
    $registration_url = sprintf( '<a href="%s">%s</a>', esc_url( wp_registration_url() ), __( 'Register' ) );

    /** This filter is documented in wp-includes/general-template.php */
    echo apply_filters( 'register', $registration_url );

    echo esc_html( $login_link_separator );
}

?>
<a href="<?php echo esc_url( wp_lostpassword_url() ); ?>"><?php _e( 'Lost your password?' ); ?
></a>
</p>
<?php
}

$login_script = 'function wp_attempt_focus() {';
$login_script .= 'setTimeout( function() {';
$login_script .= 'try {';

if ( $user_login ) {
    $login_script .= 'd = document.getElementById( "user_pass" ); d.value = ""';
} else {
    $login_script .= 'd = document.getElementById( "user_login" );';

    if ( $errors->get_error_code() === 'invalid_username' ) {
        $login_script .= 'd.value = ""';
    }
}

$login_script .= 'd.focus(); d.select();';
$login_script .= '} catch( er ) {';
$login_script .= '}, 200);';
$login_script .= '}\n"; // End of wp_attempt_focus().

/**
 * Filters whether to print the call to `wp_attempt_focus()` on the login screen.
 *
 * @since 4.8.0
 *
 * @param bool $print Whether to print the function call. Default true.
 */
if ( apply_filters( 'enable_login autofocus', true ) && ! $error ) {
    $login_script .= "wp_attempt_focus();\n";
}

// Run `wpOnload()` if defined.
$login_script .= "if ( typeof wpOnload === 'function' ) { wpOnload() }";

```

```

?>
<script type="text/javascript">
  <?php echo $login_script; ?>
</script>
<?php

if ( $interim_login ) {
  ?>
  <script type="text/javascript">
  ( function() {
    try {
      var i, links = document.getElementsByTagName( 'a' );
      for ( i in links ) {
        if ( links[i].href ) {
          links[i].target = '_blank';
          links[i].rel = 'noopener';
        }
      }
    } catch( er ) {}
  }());
  </script>
  <?php
}

login_footer();
break;
} // End action switch.

```