**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

|  |  |
| --- | --- |
|  |  |

**Курсова робота**

**з дисципліни «Програмування»**

**на тему: «**Веб-сайт пошуку роботи**»**

Виконав:

студент 1 курсу, групи ІА-33

Шмаровоз Олексій Андрійович \_\_\_\_\_\_\_\_\_\_

Керівник:

асистент кафедри ІСТ  
Мягкий Михайло Юрійович \_\_\_\_\_\_\_\_\_\_

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_\_\_\_\_\_

Київ – 2024 року

ЗМІСТ

# ВСТУП………………………………………………………………………... 3

# 1 ВИМОГИ ДО СИСТЕМИ…………………………………………………. 4

# 1.1 Функціональні вимоги до системи……………………………………... 4

# 1.2 Не функціональні вимоги до системи………………………………….. 4

# 2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ……………………………… 5

# 2.1 Діаграма прецедентів……………………………………………………. 5

# 2.2 Опис сценаріїв використання системи…………………………………. 6

# 3 АРХІТЕКТУРА СИСТЕМИ……………………………………………... 10

# 4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ……………………………. 11

# 4.1 Загальна структура проекту……………………………………………. 11

# 4.2 Компоненти рівня доступу до даних………………………………….. 12

# 4.3 Компоненти рівня бізнес-логіки………………………………………. 12

# 4.4 Компоненти рівня інтерфейсу користувача…………………………... 13

# ВИСНОВКИ…………………………………………………....…………… 14

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ………………………………….. 15

# ДОДАТОК А Лістінг програми…………………………………………… 16

# ВСТУП

# З розвитком технологій людство отримало багато нових видів діяльності, звідси пошук належної роботи став набагато складнішим. Тому замість пошуку вакансій у газетах чи особистих зв'язків, користувачі можуть скористатися спеціалізованими веб-сайтами для пошуку роботи. Це значно спрощує процес як для роботодавців, так і для кандидатів, оскільки дозволяє швидко знаходити бажані користувачем вакансії.

# Метою даної роботи є розробка веб-сайту для пошуку роботи, що забезпечить ефективну взаємодію між роботодавцями та кандидатами. Для цього система повинна мати певний перелік функцій та вирішувати наступні задачі:

# користувачі повинні мати можливість переглядати вакансії та виконувати пошук за ключовими словами, що дозволить швидко знаходити відповідні пропозиції.

# система повинна дозволяти роботодавцям створювати, редагувати та видаляти інформацію про вакансії. Кожна вакансія повинна містити опис, вимоги до кандидата та іншу корисну інформацію.

# інтерфейс сайту повинен бути інтуїтивно зрозумілим і зручним у використанні як для роботодавців, так і для кандидатів.

# 1 ВИМОГИ ДО СИСТЕМИ

# Функціональні вимоги до системи

# Система має відповідати наступним функціональним вимогам:

# незареєстровані користувачі повинні мати змогу переглядати всі вакансії, доступні на платформі;

# вакансії повинні бути представлені у вигляді списку, з коротким описом кожної вакансії (Роботодавець, зарплатня, опис, контактна інофрмація);

# зареєстрований роботодавець повинен мати змогу створювати нові вакансії, заповнюючи відповідні форми;

# роботодавці повинні мати змогу редагувати раніше створені вакансії;

# роботодавці повинні мати змогу видаляти вакансії, які більше не актуальні.

# Нефункціональні вимоги до системи:

# Система має відповідати наступним не функціональним вимогам:

# інтерфейс повинен бути інтуїтивно зрозумілим та зручним у використанні як для робітників, так і для роботодавців;

# пошук вакансій та відображення результатів повинні відбуватися в межах допустимого часу;

# система повинна бути розроблена з урахуванням можливості майбутнього розширення функціоналу та збільшення кількості користувачів.

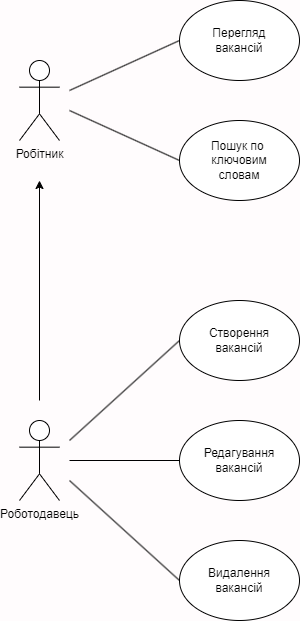
# 2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

# 2.1 Діаграма прецедентів

# Діаграма прецедентів системи показана на рис. 2.1.

# Акторами є користувачі системи: незареєстрований користувач (робітник) та зареєстрований користувач (роботодавець).

# Зареєстрованому користувачу доступна вся функціональність що і незареєстрованому користувачу, а також: створення, редагування, видалення вакансій. Детально сценарії використання описані в підрозділі 2.2.



# Рисунок 2.1 – діаграма прецедентів

# 2.2 Опис сценаріїв використання системи

# Детальні описи сценаріїв використання наведено у таблицях 2.1–2.4.

# Таблиця 2.1 – Сценарій використання “Пошук по тегам”

|  |  |
| --- | --- |
| Назва | Пошук роботи, пошук по тегу |
| ID | 1 |
| Опис | Робітник використовуючи поле для пошуку шукає пропозицію, що задовольняє робітника |
| Актори | Робітник, Роботодавець |
| Вигоди компанії | Якщо користувачі не будуть шукати роботу то вони не будуть користуватися сервісом. |
| Частота користування | Постійно |
| Тригери | Робітник вводить запит у полі для пошуку |
| Передумови | На екрані показані декілька пропозицій |
| Постумови | Робітник телефонує роботодавцю |
| Основний розвиток | Робітник домовляється з роботодавцем та отримує роботу |
| Альтернативні розвитки | Робітник не домовився з роботодавцем, та вимушений продовжити пошук |
| Виняткові ситуації | – |

# Таблиця 2.2 – Сценарій використання “Створення вакансії”

|  |  |
| --- | --- |
| Назва | Створення вакансії |
| ID | 2 |
| Опис | Роботодавець використовуючи форму створює вакансію |
| Актори | Робітник, Роботодавець |
| Вигоди компанії | Якщо користувачі не будуть створювати вакансії то не робітники не зможуть знайти роботу, що призведе до непотрібності сервісу |
| Частота користування | Постійно |
| Тригери | Роботодавець вводить дані в форму |
| Передумови | На сайті з’являється вакансія |
| Постумови | Робітник телефонує роботодавцю |
| Основний розвиток | Роботодавець домовляється з робітником та найняв його |
| Альтернативні розвитки | Роботодавець не домовився з робітником, та вимушений чекати інших дзвінків |
| Виняткові ситуації | – |

# Таблиця 2.3 – Сценарій використання “Зміна вакансії”

|  |  |
| --- | --- |
| Назва | Зміна даних в вакансії |
| ID | 3 |
| Опис | Роботодавець використовуючи форму змінює дані в вакансії |
| Актори | Роботодавець |
| Вигоди компанії | Це забезпечує користування сервісом |
| Частота користування | Постійно |
| Тригери | Роботодавець вводить дані в форму |
| Передумови | Зміна вимог до робітника або умов праці |
| Постумови | Роботодавець змінює дані в пропозиції |
| Основний розвиток | Роботодавець продовжує чекати телефонного дзвінка робітника |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

# Таблиця 2.4 – Сценарій використання “Видалення вакансії”

|  |  |
| --- | --- |
| Назва | Видалення роботодавцем вакансії |
| ID | 4 |
| Опис | Роботодавець використовуючи кнопку видаляє вакансію яка вже не потрібна |
| Актори | Роботодавець |
| Вигоди компанії | Це забезпечує користування сервісом |
| Частота користування | Постійно |
| Тригери | Роботодавець натискає кнопку видалення |
| Передумови | Роботодавець знайшов робітника |
| Постумови | Роботодавець видаляє вакансію |
| Основний розвиток | Робітник працює на роботодавця |
| Альтернативні розвитки | – |
| Виняткові ситуації | – |

# 3 АРХІТЕКТУРА СИСТЕМИ

# Загальна архітектура системи наведена на рис. 3.1.

# C:\Users\Acer\Downloads\Діаграма без назви.drawio.png

# Рисунок 3.1 – Загальна архітектура системи

# 4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

# 4.1 Загальна структура проекту

# Загальна структура проекту зображена на рис.4.1

# 

# Рисунок 4.1 – загальна структура проекту

# Проект складається з html сторінок, css, а також компонентів бізнес-логіки.

# 4.2 Компоненти рівня доступу до даних

# Доступ до даних виконують такі компоненти:

# login.js, logincheck.js та logout.js виконують доступ до інформації про авторизованого користувача;

* search.js, edit.js, create\_offer.js виконують доступ до даних пов’язаних з вакансіями.

# 4.3 Компоненти рівня бізнес-логіки

# Папка “scripts” містить файли необхідні для оброби інформації:

# create.js містить функції які дозволяють створювати та видаляти вакансії;

# edit.js містить функції які дозволяють редагувати вакансії роботодавцю;

# login.js містить функцію яка дозволяє авторизуватися на сайті;

# logincheck.js містить функцію яка не дозволяє переходити на сторінку створення вакансій незареєстрованим користувачам;

# logout.js видаляє інформацію про авторизованого користувача та переносить користувача на головну сторінку;

# navbar.js містить скрипт який створює мобільне меню, якщо розширення сайту;

# search.js містить функції необхідні для пошуку вакансій за тегом або мінімальною зарплатнею.

# 4.4 Компоненти рівня інтерфейсу користувача

# Папка “css” містить файли для створення стилю сайту:

# background.css містить стиль заднього фону сторінок;

# create.css містить стиль наповнення сторінки створення вакансій;

# footer.css містить стиль нижньої частини сайту відповідальної за контактну інформацію;

# general.css містить стиль;

# login\_form.css містить стиль форми входу на сайт;

# navbar.css містить стиль навігаційного меню;

# offers.css містить стиль вакансій для сторінки пошуку;

# search.css містить стиль форми пошуку.

# Папка images містить зображення в форматі .svg для покращення стилю головної сторінки

# ВИСНОВКИ

# Під час написання курсової роботи було розглянуто та проаналізовано ринок веб-сайтів пошуку роботи в Україні, а також визначено основні переваги та недоліки існуючих рішень. У результаті було прийнято рішення розробити власну систему, яка дозволить користувачам ефективно знаходити та керувати вакансіями.

# Першим кроком було сформулювати функціональні та нефункціональні вимоги до системи, що визначило очікувану поведінку та характеристики системи. Для реалізації проекту було обрано технології, які забезпечують швидкість та зручність розробки. Використовувалась мова програмування JavaScript для динамічної взаємодії з користувачем, а середа розробки Visual Studio Code забезпечила зручне середовище для кодування. HTML та CSS були використані для створення структури та стилізації веб-сторінок.

# Наступним етапом було описання сценаріїв використання системи. Було визначено основні дії, які можуть виконувати користувачі системи – роботодавці та робітники. Роботодавці мають можливість створювати, редагувати та видаляти інформацію про вакансії. Робітники, в свою чергу, можуть переглядати вакансії та здійснювати пошук по ключовим словам для швидкого знаходження відповідних пропозицій.

# Було розроблено загальну архітектуру системи з урахуванням принципів зручності та масштабованості. Система була розділена на логічні рівні, що забезпечило легкість її підтримки та розширення. Особлива увага була приділена створенню зручного та інтуїтивно зрозумілого графічного інтерфейсу, який дозволяє користувачам швидко орієнтуватися на сайті та знаходити необхідну інформацію.

# Далі було створено моделі сутностей, таких як вакансія, опис вакансії та вимоги до кандидата. Це дозволило структуровано зберігати дані та забезпечити легкий доступ до них. Завдяки цьому роботодавці можуть легко управляти інформацією про вакансії, а робітники – знаходити пропозиції, що відповідають їх кваліфікаціям та вимогам.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

# Сайт з безкоштовними кольоровими палітрами [Електронний ресурс] – Режим доступу до ресурсу: https://coolors.co

# Сайт з безкоштовними ілюстраціями [Електронний ресурс] – Режим доступу до ресурсу: <https://undraw.co/search>

# Підручник з CSS [Електронний ресурс] – Режим доступу до ресурсу: <https://w3schoolsua.github.io/css/index.html#gsc.tab=0>

# Підручник з HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://w3schoolsua.github.io/html/index.html#gsc.tab=0>

# Підручник з Javascript [Електронний ресурс] – Режим доступу до ресурсу: https://w3schoolsua.github.io/js/index.html#gsc.tab=0

# ДОДАТОК А

# Лістінг програми

# Посилання на GitHub: https://github.com/ShmarovozOleksii/Coursework/

# Html сторінки проекту:

# Index.html (головна сторінка):

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Home</title>

    <link rel="stylesheet" href="css/general.css">

    <link rel="stylesheet" href="css/navbar.css">

    <link rel="stylesheet" href="css/footer.css">

</head>

<body>

    <nav class="navbar">

        <div class="navbar\_container">

            <a id="navbar\_text">Jobs.ua</a>

            <div class="navbar\_toggle" id="mobile-menu">

                <span class="bar"></span>

                <span class="bar"></span>

                <span class="bar"></span>

            </div>

            <ul class="navbar\_menu">

                <li class="navbar\_item">

                    <a href="index.html" class="navbar\_links">Home</a>

                </li>

                <li class="navbar\_item">

                    <a href="search.html" class="navbar\_links">Search</a>

                </li>

                <li class="navbar\_btn">

                    <a href="authorisation.html" class="button">Log in</a>

                </li>

            </ul>

        </div>

    </nav>

    <div class="main">

        <div class="main\_container">

            <div class="main\_content">

                <h1>Find the job</h1>

                <h2>Of your dreams!</h2>

            </div>

            <div class="main\_img-container">

                <img src="images/job\_hunt.svg" alt="pic" id="main\_img">

            </div>

        </div>

    </div>

    <footer class="footer">

        <div class="footer\_container">

            <div class="footer\_content">

                <p>Contact us:</p>

                <p>Phone: +380 - 123 - 456 - 789</p>

                <p>Email: support@jobs.ua</p>

            </div>

            <div class="footer\_content">

                <p>&copy; 2024 Jobs.ua | All rights reserved.</p>

            </div>

        </div>

    </footer>

    <script src="scripts/navbar.js"></script>

</body>

</html>

# search.html (сторінка пошуку вакансій):

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Search</title>

    <link rel="stylesheet" href="css/navbar.css">

    <link rel="stylesheet" href="css/footer.css">

    <link rel="stylesheet" href="css/offers.css">

    <link rel="stylesheet" href="css/background.css">

    <link rel="stylesheet" href="css/search.css">

</head>

<body>

    <nav class="navbar">

        <div class="navbar\_container">

            <a id="navbar\_text">Jobs.ua</a>

            <div class="navbar\_toggle" id="mobile-menu">

                <span class="bar"></span>

                <span class="bar"></span>

                <span class="bar"></span>

            </div>

            <ul class="navbar\_menu">

                <li class="navbar\_item">

                    <a href="index.html" class="navbar\_links">Home</a>

                </li>

                <li class="navbar\_item">

                    <a href="search.html" class="navbar\_links">Search</a>

                </li>

                <li class="navbar\_btn">

                    <a href="authorisation.html" class="button">Log in</a>

                </li>

            </ul>

        </div>

    </nav>

    <div class="search-container">

        <h1>Search</h1>

        <input type="text" id="tagInput" placeholder="Tag">

        <input type="number" id="salaryInput" placeholder="Minimum salary">

        <button onclick="searchJobs()">Search</button>

    </div>

    <h1>Offer Information</h1>

    <div id="displayInfo"></div>

    <script src="scripts/search.js"></script>

    <footer class="footer">

        <div class="footer\_container">

            <div class="footer\_content">

                <p>Contact us:</p>

                <p>Phone: +380 - 123 - 456 - 789</p>

                <p>Email: support@jobs.ua</p>

            </div>

            <div class="footer\_content">

                <p>&copy; 2024 Jobs.ua | All rights reserved.</p>

            </div>

        </div>

    </footer>

    <script src="scripts/navbar.js"></script>

</body>

</html>

# authorization.html (сторінка авторизації):

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Log in</title>

    <link rel="stylesheet" href="css/navbar.css">

    <link rel="stylesheet" href="css/footer.css">

    <link rel="stylesheet" href="css/login\_form.css">

    <link rel="stylesheet" href="css/background.css">

</head>

<body>

    <nav class="navbar">

        <div class="navbar\_container">

            <a id="navbar\_text">Jobs.ua</a>

            <div class="navbar\_toggle" id="mobile-menu">

                <span class="bar"></span>

                <span class="bar"></span>

                <span class="bar"></span>

            </div>

            <ul class="navbar\_menu">

                <li class="navbar\_item">

                    <a href="index.html" class="navbar\_links">Home</a>

                </li>

                <li class="navbar\_item">

                    <a href="search.html" class="navbar\_links">Search</a>

                </li>

                <li class="navbar\_btn">

                    <a href="authorisation.html" class="button">Log in</a>

                </li>

            </ul>

        </div>

    </nav>

    <div class="container">

        <h1>Log in</h1>

        <form class="loginform">

                <label for="username" class="label">Username</label>

                <input type="text" name="username" />

                <label for="password" class="label">Password</label>

                <input type="password" name="password" />

                <input type="submit" value="Login" />

        </form>

    </div>

    <footer class="footer">

        <div class="footer\_container">

            <div class="footer\_content">

                <p>Contact us:</p>

                <p>Phone: +380 - 123 - 456 - 789</p>

                <p>Email: support@jobs.ua</p>

            </div>

            <div class="footer\_content">

                <p>&copy; 2024 Jobs.ua | All rights reserved.</p>

            </div>

        </div>

    </footer>

    <script src="scripts/login.js"></script>

    <script src="scripts/navbar.js"></script>

</body>

</html>

# create\_offer.html (сторінка створення/редагування/видалення вакансій):

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Create-offer</title>

    <link rel="stylesheet" href="css/navbar.css">

    <link rel="stylesheet" href="css/footer.css">

    <link rel="stylesheet" href="css/offers.css">

    <link rel="stylesheet" href="css/background.css">

    <link rel="stylesheet" href="css/create.css">

</head>

<body>

    <nav class="navbar">

        <div class="navbar\_container">

            <a id="navbar\_text">Jobs.ua</a>

            <div class="navbar\_toggle" id="mobile-menu">

                <span class="bar"></span>

                <span class="bar"></span>

                <span class="bar"></span>

            </div>

            <ul class="navbar\_menu">

                <li class="navbar\_btn">

                    <a href="" onclick="Logout()" class="button">Logout</a>

                </li>

            </ul>

        </div>

    </nav>

    <h1>Job Information Form</h1>

    <form id="jobForm">

        <input type="hidden" id="editIndex" name="editIndex" value="">

        <label for="tag">Tag:</label>

        <input type="text" id="tag" name="tag" required><br><br>

        <label for="salary">Salary:</label>

        <input type="text" id="salary" name="salary" required><br><br>

        <label for="employer">Employer:</label>

        <input type="text" id="employer" name="employer" required><br><br>

        <label for="description">Description:</label>

        <textarea id="description" name="description" required></textarea><br><br>

        <label for="contact\_number">Contact Number:</label>

        <input type="text" id="contact\_number" name="contact\_number" required><br><br>

        <button type="button" onclick="submitForm()">Submit</button>

    </form>

    <h1>Submitted Job Information</h1>

    <div id="displayInfo"></div>

    <script src="scripts/create.js"></script>

    <footer class="footer">

        <div class="footer\_container">

            <div class="footer\_content">

                <p>Contact us:</p>

                <p>Phone: +380 - 123 - 456 - 789</p>

                <p>Email: support@jobs.ua</p>

            </div>

            <div class="footer\_content">

                <p>&copy; 2024 Jobs.ua | All rights reserved.</p>

            </div>

        </div>

    </footer>

    <script src="scripts/navbar.js"></script>

    <script src="scripts/logincheck.js"></script>

    <script src="scripts/logout.js"></script>

    <script src="scripts/edit.js"></script>

</body>

</html>

# Бізнес логіка проекту (javascript):

# create.js:

document.addEventListener('DOMContentLoaded', () => {

    displayData();

});

function submitForm() {

    const tag = document.getElementById('tag').value;

    const salary = document.getElementById('salary').value;

    const employer = document.getElementById('employer').value;

    const description = document.getElementById('description').value;

    const contactNumber = document.getElementById('contact\_number').value;

    const loggedInUser = localStorage.getItem('loggedInUser');

    const jobInfo = {

        tag: tag,

        salary: salary,

        employer: employer,

        description: description,

        contactNumber: contactNumber,

        createdBy: loggedInUser

    };

    let jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    jobInfoList.push(jobInfo);

    localStorage.setItem('jobInfoList', JSON.stringify(jobInfoList));

    displayData();

    clearForm();

}

function displayData() {

    const displayInfo = document.getElementById('displayInfo');

    displayInfo.innerHTML = '';

    const loggedInUser = localStorage.getItem('loggedInUser');

    let jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    jobInfoList = jobInfoList.filter(job => job.createdBy === loggedInUser);

    jobInfoList.forEach((job, index) => {

        const jobDiv = document.createElement('div');

        jobDiv.classList.add('job-box');

        jobDiv.innerHTML = `

            <h3>Offer #${index + 1}</h3>

            <p><strong>Tag:</strong> ${job.tag}</p>

            <p><strong>Salary:</strong> ${job.salary}</p>

            <p><strong>Employer:</strong> ${job.employer}</p>

            <p><strong>Description:</strong> ${job.description}</p>

            <p><strong>Contact Number:</strong> ${job.contactNumber}</p>

        `;

        const removeButton = document.createElement('button');

        removeButton.textContent = 'Remove';

        removeButton.classList.add('remove-job');

        removeButton.onclick = () => removeJob(index);

        jobDiv.appendChild(removeButton);

        displayInfo.appendChild(jobDiv);

    });

}

function removeJob(index) {

    let jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    const loggedInUser = localStorage.getItem('loggedInUser');

    jobInfoList = jobInfoList.filter(job => job.createdBy === loggedInUser);

    jobInfoList.splice(index, 1);

    const allJobs = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    const updatedAllJobs = allJobs.filter(job => job.createdBy !== loggedInUser).concat(jobInfoList);

    localStorage.setItem('jobInfoList', JSON.stringify(updatedAllJobs));

    displayData();

}

function clearForm() {

    document.getElementById('jobForm').reset();

}

# edit.js:

document.addEventListener('DOMContentLoaded', () => {

    displayData();

});

function submitForm() {

    const tag = document.getElementById('tag').value;

    const salary = document.getElementById('salary').value;

    const employer = document.getElementById('employer').value;

    const description = document.getElementById('description').value;

    const contactNumber = document.getElementById('contact\_number').value;

    const loggedInUser = localStorage.getItem('loggedInUser');

    const editIndex = document.getElementById('editIndex').value;

    const jobInfo = {

        tag: tag,

        salary: salary,

        employer: employer,

        description: description,

        contactNumber: contactNumber,

        createdBy: loggedInUser

    };

    let jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    if (editIndex === "") {

        jobInfoList.push(jobInfo);

    } else {

        jobInfoList[editIndex] = jobInfo;

        document.getElementById('editIndex').value = "";

    }

    localStorage.setItem('jobInfoList', JSON.stringify(jobInfoList));

    displayData();

    clearForm();

}

function displayData() {

    const displayInfo = document.getElementById('displayInfo');

    displayInfo.innerHTML = '';

    const loggedInUser = localStorage.getItem('loggedInUser');

    let jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    jobInfoList = jobInfoList.filter(job => job.createdBy === loggedInUser);

    jobInfoList.forEach((job, index) => {

        const jobDiv = document.createElement('div');

        jobDiv.classList.add('job-box');

        jobDiv.innerHTML = `

            <h3>Offer #${index + 1}</h3>

            <p><strong>Tag:</strong> ${job.tag}</p>

            <p><strong>Salary:</strong> ${job.salary}</p>

            <p><strong>Employer:</strong> ${job.employer}</p>

            <p><strong>Description:</strong> ${job.description}</p>

            <p><strong>Contact Number:</strong> ${job.contactNumber}</p>

        `;

        const removeButton = document.createElement('button');

        removeButton.textContent = 'Remove';

        removeButton.classList.add('remove-job');

        removeButton.onclick = () => removeJob(index);

        jobDiv.appendChild(removeButton);

        const editButton = document.createElement('button');

        editButton.textContent = 'Edit';

        editButton.classList.add('edit-job');

        editButton.onclick = () => editJob(index);

        jobDiv.appendChild(editButton);

        displayInfo.appendChild(jobDiv);

    });

}

function editJob(index) {

    let jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    const loggedInUser = localStorage.getItem('loggedInUser');

    jobInfoList = jobInfoList.filter(job => job.createdBy === loggedInUser);

    const job = jobInfoList[index];

    document.getElementById('tag').value = job.tag;

    document.getElementById('salary').value = job.salary;

    document.getElementById('employer').value = job.employer;

    document.getElementById('description').value = job.description;

    document.getElementById('contact\_number').value = job.contactNumber;

    document.getElementById('editIndex').value = index;

}

function removeJob(index) {

    let jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    const loggedInUser = localStorage.getItem('loggedInUser');

    jobInfoList = jobInfoList.filter(job => job.createdBy === loggedInUser);

    jobInfoList.splice(index, 1);

    const allJobs = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    const updatedAllJobs = allJobs.filter(job => job.createdBy !== loggedInUser).concat(jobInfoList);

    localStorage.setItem('jobInfoList', JSON.stringify(updatedAllJobs));

    displayData();

}

function clearForm() {

    document.getElementById('jobForm').reset();

    document.getElementById('editIndex').value = "";

}

# login.js:

const form = document.querySelector("form")

form.addEventListener("submit", (e) => {

    e.preventDefault()

    const username = form.username.value

    const password = form.password.value

    const authenticated = authentication(username, password)

    if (authenticated) {

        localStorage.setItem('loggedInUser', username);

        window.location.href = "create\_offer.html"

    } else {

        alert("Username or Password is incorrect")

    }

})

var users = [

    { username: 'admin', password: 'password' },

    { username: 'user1', password: 'password1' },

    { username: 'user2', password: 'password2' }

];

function authentication(username, password) {

    for (var i = 0; i < users.length; i++) {

        if (username === users[i].username && password === users[i].password) {

            return true

        }

    }

    return false

}

# logincheck.js:

document.addEventListener("DOMContentLoaded", () => {

    const loggedInUser = localStorage.getItem('loggedInUser');

    if (loggedInUser === null) {

        window.location.href = "authorisation.html";

    }

});

# logout.js:

function Logout() {

    localStorage.removeItem('loggedInUser');

    window.location.href = 'index.html';

}

# navbar.js:

const menu = document.querySelector('#mobile-menu');

const menuLinks = document.querySelector('.navbar\_menu');

menu.addEventListener('click', function() {

    menu.classList.toggle('is-actives');

    menuLinks.classList.toggle('active');

})

# search.js:

document.addEventListener('DOMContentLoaded', () => {

    displayData();

});

function displayData(jobInfoList = null) {

    const displayInfo = document.getElementById('displayInfo');

    displayInfo.innerHTML = '';

    if (!jobInfoList) {

        jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    }

    jobInfoList.forEach((job, index) => {

        const jobDiv = document.createElement('div');

        jobDiv.classList.add('job-box');

        jobDiv.innerHTML = `

            <h3>Position #${index + 1}</h3>

            <p><strong>Tag:</strong> ${job.tag}</p>

            <p><strong>Salary:</strong> ${job.salary}</p>

            <p><strong>Employer:</strong> ${job.employer}</p>

            <p><strong>Description:</strong> ${job.description}</p>

            <p><strong>Contact Number:</strong> ${job.contactNumber}</p>

        `;

        displayInfo.appendChild(jobDiv);

    });

}

function searchJobs() {

    const tagInput = document.getElementById('tagInput').value.toLowerCase();

    const salaryInput = parseInt(document.getElementById('salaryInput').value, 10);

    const jobInfoList = JSON.parse(localStorage.getItem('jobInfoList')) || [];

    const filteredJobs = jobInfoList.filter(job => {

        const matchesTag = job.tag.toLowerCase().includes(tagInput);

        const matchesSalary = !isNaN(salaryInput) ? job.salary >= salaryInput : true;

        return matchesTag && matchesSalary;

    });

    displayData(filteredJobs);

}

# Css проекту:

# background.css:

body {

    background: linear-gradient(135deg, #240046, #9D4EDD, #FF9E00, #FF6D00);

    background-size: 400% 400%;

    animation: gradientBG 40s ease infinite;

    margin: 0;

    padding: 0;

    font-family: Arial, sans-serif;

}

@keyframes gradientBG {

    0% { background-position: 0% 50%; }

    50% { background-position: 100% 50%; }

    100% { background-position: 0% 50%; }

}

# create.css:

#jobForm {

    background-color: #101010;

    padding: 40px;

    max-width: 600px;

    margin: 20px auto;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

    text-align: center;

}

#jobForm label {

    color: #fff;

    margin: 10px 0 5px;

    font-size: 1.2rem;

    display: block;

    text-align: left;

}

#jobForm input[type="text"],

#jobForm textarea {

    width: 100%;

    padding: 10px;

    margin-bottom: 0px;

    border: 1px solid #333;

    border-radius: 5px;

    background-color: #222;

    color: #fff;

}

#jobForm textarea {

    height: 150px;

    resize: vertical;

}

#jobForm button[type="button"] {

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    background: #FF9100;

    color: #fff;

    font-size: 1.2rem;

    cursor: pointer;

}

#jobForm button[type="button"]:hover {

    background: #9D4EDD;

    transition: 0.3s ease;

}

#jobForm input[type="text"]:focus,

#jobForm textarea:focus {

    border-color: #FF9100;

    outline: none;

}

button.remove-job,

button.edit-job {

    padding: 10px 20px;

    margin: 5px 10px;

    border: none;

    border-radius: 5px;

    background: #FF9E00;

    color: #fff;

    font-size: 1rem;

    cursor: pointer;

}

button.remove-job:hover {

    background: #9D4EDD;

    transition: 0.3s ease;

}

@media screen and (max-width: 480px) {

    #jobForm {

        width: 90%;

        padding: 20px;

    }

}

# footer.css:

.footer {

    background-color: #101010;

    color: #fff;

    padding: 20px 0;

    text-align: center;

    position: relative;

    width: 100%;

    bottom: 0;

}

.footer\_container {

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    max-width: 1300px;

    margin: 0 auto;

    padding: 0 50px;

}

.footer\_content{

    margin: 5px 0;

    font-size: 1rem;

}

.footer\_contact {

    margin-bottom: 10px;

}

.content {

    flex: 1;

}

# general.css:

.main {

    background-color: #181818;

}

.main\_container {

    display: grid;

    grid-template-columns: 1fr 1fr;

    align-items: center;

    justify-self: center;

    margin: 0 auto;

    height: 110vh;

    background-color: #181818;;

    z-index: 1;

    width: 100%;

    max-width: 1300px;

    padding: 0 50px;

}

.main\_content h1 {

    font-size: 4rem;

    background-color: #FF8500;

    background-image: linear-gradient(to top, #FF6D00 0%, #FF9E00 100%);

    background-size: 100%;

    -webkit-background-clip: text;

    -moz-background-clip: text;

    -webkit-text-fill-color: transparent;

    -moz-text-fill-color: transparent;

    size-adjust: 100%;

}

.main\_content h2 {

    font-size: 4rem;

    background-color: #FF8500;

    background-image: linear-gradient(to top, #5A189A 0%, #9D4EDD 100%);

    background-size: 100%;

    -webkit-background-clip: text;

    -moz-background-clip: text;

    -webkit-text-fill-color: transparent;

    -moz-text-fill-color: transparent;

    size-adjust: 100%;

}

#main\_img {

    max-height: 110%;

    max-width: 110%;

}

# login\_form.css:

.container {

    background-color: #101010;

    padding: 40px;

    max-width: 400px;

    margin: 53px auto;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

    text-align: center;

}

.container h1 {

    color: #fff;

    margin-bottom: 20px;

    font-size: 2.3rem;

}

.loginform {

    display: flex;

    flex-direction: column;

    align-items: center;

}

.loginform .label {

    color: #fff;

    margin: 10px 0 5px;

    font-size: 1.2rem;

}

.loginform input[type="text"],

.loginform input[type="password"] {

    width: 100%;

    padding: 10px;

    margin-bottom: 20px;

    border: 1px solid #333;

    border-radius: 5px;

    background-color: #222;

    color: #fff;

}

.loginform input[type="submit"] {

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    background: #FF9100;

    color: #fff;

    font-size: 1.2rem;

    cursor: pointer;

}

.loginform input[type="submit"]:hover {

    background: #9D4EDD;

    transition: 0.3s ease;

}

.loginform input[type="text"]:focus,

.loginform input[type="password"]:focus {

    border-color: #FF9100;

    outline: none;

}

@media screen and (max-width: 480px) {

    .container {

        width: 90%;

        padding: 20px;

    }

}

# navbar.css:

    \* {

        box-sizing: border-box;

        margin: 0;

        padding: 0;

        font-family: 'Kumbh Sans', sans-serif;

    }

    .navbar {

        background: #101010;

        height: 80px;

        display: flex;

        justify-content: center;

        align-items: center;

        font-size: 1.4rem;

        position: sticky;

        top: 0;

        z-index: 999;

    }

    .navbar\_container {

        display: flex;

        justify-content: space-between;

        height: 80;

        z-index: 1;

        width: 100%;

        max-width: 1300px;

        margin: 0 5%;

    }

    #navbar\_text {

        background-color: #7B2CBF;

        background-image: linear-gradient(to top, #7B2CBF 0%, #9D4EDD 100%);

        background-size: 100%;

        -webkit-background-clip: text;

        -moz-background-clip: text;

        -webkit-text-fill-color: transparent;

        -moz-text-fill-color: transparent;

        display: flex;

        align-items: center;

        cursor: pointer;

        font-size: 2.5rem;

    }

    .navbar\_menu {

        display: flex;

        align-items: center;

        list-style: none;

        text-align: center;

    }

    .navbar\_item {

        height: 80px;

    }

    .navbar\_links {

        color: #fff;

        display: flex;

        align-items: center;

        justify-content: center;

        text-decoration: none;

        padding: 0 1rem;

        height: 100%;

    }

    .navbar\_btn {

        display: flex;

        justify-content: center;

        align-items: center;

        padding: 0 1rem;

        width: 100%;

    }

    .button {

        font-family: Arial;

        display: flex;

        justify-content: center;

        align-items: center;

        text-decoration: none;

        padding: 10px 20px;

        height: 100%;

        width: 100%;

        border: none;

        outline: none;

        border-radius: 4px;

        background: #FF9100;

        color: #fff;

    }

    .button:hover {

        background: #9D4EDD;

        transition: all 0.5s ease;

    }

    .navbar\_links:hover {

        color: #FF9100;

        transition: 0.5s ease;

    }

    @media screen and (max-width: 960px) {

        .navbar\_container {

            display: flex;

            justify-content: space-between;

            height: 80px;

            z-index: 1;

            width: 100%;

            max-width: 1300px;

            padding: 0;

        }

        .navbar\_menu {

            display: grid;

            grid-template-columns: auto;

            margin: 0;

            width: 100%;

            position: absolute;

            top: -1000px;

            opacity: 0;

            transition: all 0.5s ease;

            height: 50vh;

            z-index: -1;

            background: #101010;

        }

        .navbar\_menu.active {

            background: #101010;

            top: 100%;

            opacity: 1;

            transition: all 0.5s ease;

            z-index: 99;

            height: 50vh;

            font-size: 1.6rem;

        }

        .navbar\_toggle .bar {

            width: 25px;

            height: 3px;

            margin: 5px auto;

            transition: all 0.5s ease-in-out;

            background: #fff;

        }

        .navbar\_item {

            width: 100%;

        }

        .navbar\_links {

            text-align: center;

            padding: 2rem;

            width: 100%;

            display: table;

        }

        #mobile-menu {

            position: absolute;

            top: 20%;

            right: 5%;

            transform: translate(5%, 20%);

        }

        .navbar\_btn {

            padding-bottom: 2rem;

        }

        .button {

            display: flex;

            justify-content: center;

            align-items: center;

            width: 80%;

            height: 80px;

            margin: 0;

        }

        .navbar\_toggle .bar {

            display: block;

            cursor: pointer;

        }

        #mobile-menu.is-active .bar:nth-child(2){

            opacity: 0;

        }

        #mobile-menu.is-active .bar:nth-child(1){

            transform: translate(8px) rotate(45deg);

        }

        #mobile-menu.is-active .bar:nth-child(1){

            transform: translate(-8px) rotate(-45deg);

        }

    }

# offers.css:

h1 {

    margin: 1% 3%;

    color: #000;

}

.job-container {

    display: flex;

    flex-direction: column;

    align-items: center;

    padding: 20px;

}

.job-box {

    background: #181818;

    border: 1px solid #000;

    border-radius: 8px;

    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

    padding: 20px;

    margin: 1% auto;

    width: 80%;

    max-width: 800px;

}

.job-box h3 {

    margin-bottom: 10px;

    color: #fff;

}

.job-box p {

    margin: 5px 0;

    color: #fff;

}

.job-box p {

    margin: 5px 0;

    color: #fff;

    word-wrap: break-word;

    white-space: pre-wrap;

}

h1 {

    text-align: center;

}

# search.css:

.search-container {

    background-color: #101010;

    padding: 20px;

    max-width: 600px;

    margin: 40px auto;

    border-radius: 10px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

    text-align: center;

}

.search-container h1 {

    color: #fff;

    margin-bottom: 20px;

    font-size: 2.3rem;

}

.search-container .searchform {

    display: flex;

    flex-direction: column;

    align-items: center;

}

.search-container .label {

    color: #fff;

    margin: 10px 0 5px;

    font-size: 1.2rem;

}

.search-container input[type="text"],

.search-container input[type="number"] {

    width: 100%;

    padding: 10px;

    margin-bottom: 20px;

    border: 1px solid #333;

    border-radius: 5px;

    background-color: #222;

    color: #fff;

}

.search-container button {

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    background: #FF9100;

    color: #fff;

    font-size: 1.2rem;

    cursor: pointer;

}

.search-container button:hover {

    background: #9D4EDD;

    transition: 0.3s ease;

}

.search-container input[type="text"]:focus,

.search-container input[type="number"]:focus {

    border-color: #FF9100;

    outline: none;

}

@media screen and (max-width: 480px) {

    .search-container {

        width: 90%;

        padding: 20px;

    }

}