

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента Волошина Дмитра Андрійовича
(ПІБ)

академічної групи 122-18-1
(шифр)

спеціальності 122 Комп'ютерні науки
(код і назва спеціальності)

освітньої програми Комп'ютерні науки
(назва освітньої програми)

на тему: Розробка десктопного додатку для SEO аналізу сайтів

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Спиринцев В.В.			
розділів:				
спеціальний	доц. Спиринцев В.В.			
економічний	доц. Касьяненко Л.В.			
Рецензент				
Нормоконтролер	доц. Гуліна І.Г.			

Дніпро
2022

РЕФЕРАТ

Пояснювальна записка: 116с., 64 рис., 3 дод., 23 джерела.

Об'єкт розробки: десктопний додаток для SEO-оптимізації сайтів.

Мета кваліфікаційної роботи: створити додаток, який буде знаходити недоліки технічного аспекту та мовного аспектів SEO-оптимізації, що присутні на веб-ресурсах, та які негативно впливають на їх ранжування у пошуковій видачі.

У вступі розглядається сучасний стан проблеми, конкретизується мета кваліфікаційної роботи, актуальність, галузь її застосування, уточнюється постановка завдання.

У першому розділі проаналізовано предметну галузь, дані визначення основним термінам, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформи для розробки, виконано проектування і розробка програми, описана робота програми, алгоритм і структура її функціонування, а також виклик та завантаження програми, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення додатку полягає у пошуку та наданні певної інформації про деякі недоліки які присутні на веб-ресурсах, котрі негативно впливають на їх ранжування.

Актуальність інформаційної системи визначається великим попитом на просування сайтів у мережі Інтернет, просування надає можливості для залучення інвестицій та приваблення клієнтів на веб-ресурс.

Список ключових слів: ДОДАТОК, SEO-ОПТИМІЗАЦІЯ, ПРОСУНЕННЯ, ІНТЕРНЕТ, ПОШУКОВІ СИСТЕМИ.

ABSTRACT

Explanatory note: 116p., 64 figs., 3 add., 23 sources.

Object of development: desktop application for SEO-optimization of sites.

The purpose of the qualification work: to create an application that will find the shortcomings of the technical aspect and the linguistic aspects of SEO optimization present on web resources that negatively affect their ranking in search results.

In the introduction, the current state of the problem is considered, the purpose of the qualification work, the relevance, the scope of its application are specified, the problem statement is specified.

In the first section analyzes the subject industry, these definitions of the main terms, determines the relevance of the task and the purpose of the development, formulates the problem statement, specifies the requirements for software implementation, technologies and software tools.

In the second section, the available solutions are analyzed, platforms for development are selected, the design and development of the program are carried out, the operation of the program, the algorithm and structure of its functioning, as well as the call and loading of the program are described, the input and initial data are determined, the composition of the parameters of technical means is characterized.

In the economic section, the labor intensity of the developed information system is determined, the cost of creating a program is calculated, and the time of its creation is calculated.

The practical value of the application is to find and provide certain information about some of the shortcomings that are present on web resources that negatively affect their ranking.

The relevance of the information system is determined by the great demand for website promotion on the Internet, promotion provides opportunities to attract investment and attract customers to a web resource.

List of keywords: APPLICATION, SEO-OPTIMIZATION, PROMOTION, INTERNET, SEARCH ENGINES.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	
1.1. Загальні відомості з предметної галузі.....	10
1.1.1 Основні поняття пошукової оптимізації.....	10
1.1.2 Аспекти SEO-оптимізації.....	16
1.2. Призначення розробки та галузь застосування.....	23
1.3. Підстава для розробки.....	24
1.4. Постановка завдання.....	24
1.5. Вимоги до програми або програмного виробу.....	25
1.5.1. Вимоги до функціональних характеристик.....	25
1.5.2. Вимоги до інформаційної безпеки.....	25
1.5.3. Вимоги до складу та параметрів технічних засобів.....	26
1.5.4. Вимоги до інформаційної та програмної сумісності	26
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	28
2.1. Функціональне призначення системи.....	28
2.2. Опис застосованих математичних методів.....	28
2.3. Опис використаних технологій та мов програмування.....	29
2.4. Опис структури системи та алгоритмів її функціонування	39
2.5. Обґрунтування та організація вхідних та вихідних даних програми.....	64
2.6. Опис розробленої системи	64
2.6.1. Використані технічні засоби.....	64
2.6.2. Використані програмні засоби.....	64
2.6.3. Виклик та завантаження програми.....	65
2.6.4. Опис інтерфейсу користувача.....	65

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	84
1.1. Розрахунок трудомісткості та вартості розробки програмного продукту.....	84
1.2. Розрахунок витрат на створення програми.....	87
ВИСНОВКИ.....	89
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	91
Додаток А. Код програми.....	93
Додаток Б. Відгук керівника економічного розділу.....	115
Додаток В. Перелік файлів на диску.....	116

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

SEO – search engine optimization (пошукова оптимізація);

JS – JavaScript (мова програмування);

CSS - Cascading Style Sheets (мова стилів на веб-сторінках);

HTTP/S - HyperText Transfer Protocol/Secure (протокол передачі гіпертексту);

GUI – Graphic user interface (графічний інтерфейс користувача);

IP – Internet protocol (маршрутизований протокол мережного рівня стека

TCP/IP);

URL/SEF URL – Uniform resource locator/ Search Engine Friendly URL (єдиний локатор ресурсів/ дружній до пошукової системи);

CMS – Content management system (Система управління контентом);

PDF/XLSX/CSV/XML – формат файлів.

ВСТУП

Із кількістю сайтів та користувачів ростуть й можливості самого Інтернету. На даний момент мало кого здивуєш, що можна зробити оплату комунальних послуг, доставки будь чого з іншого кінця світу, замовити їжу, придбати іграшки та одяг, та навіть заробити собі на життя. Всесвітня мережа у сьогоднішнього – це дуже впливова сила: незамінний засіб поширення та пошуку інформації, інструмент найшвидшого в світі способу просування, майже необширна можливість комунікації. Можна стверджувати, що на даний момент, розвиток інтернету не дійшов до своєї межі, існує безліч галузей які досі розвиваються.

У сьогоднішнього можна отримати актуальну інформацію про людей, які вас оточують, та тих що знаходяться за тисячі кілометрів від вас. Це стало реально завдяки такому явищу, як соціальні мережі, котрі об'єднали мільйони людей. У сучасній реальності кожен перший має акаунт бодай у одній з них. Чим привабливіше акаунт - тим більше у власника популярність та кількість фоловерів. Кожен, витративши деякий час та додавши трохи зусиль може отримати бажаний результат. А тепер давайте уявимо, що ці цілі переслідує будь-яка фірма, компанія або корпорація, яка зацікавлена у створенні певного імені для свого продукту. Ця компанія буде витратити кошти, та наймати певних спеціалістів для реалізації цього плану. Зачасту компанії використовують власні веб-сайти або інтернет-майданчики як спосіб розповсюдження інформації про себе та як засіб просування свого продукту та певних послуг.

Для того що б користувач знайшов сайт, який просуває компанія, виконується комплекс мір з покращення його ранжування у пошукових системах – SEO-оптимізація(Search Engine Optimization). SEO-оптимізацію використовують для виведення сайту у топ результатів по певному пошуковому запиту. Сукупність таких пошукових запитів називають семантичним ядром сайту. Також на одному рівні з SEO, використовуються такі технології, як SMM (Social Media Marketing) та SEA (Search Engine Advertising).

Дослідження показує (згідно Forrester Research Inc.), що найчастіше користувач наврядчи буде дивитися більше двох сторінок, що йому пропонує браузер на його запит. Одже, компанії вигідно та необхідно зробити так, щоб їх сайт відображався на цих двох перших сторінках.

Виходячи з точки зору розвитку, SEO-оптимізація є сферою достатньо вивченою, але швидко оновлюємою. Постійний розвиток та модернізація Інтернету впливає на актуальність багатьох галузей у певний проміжок часу, навіть дані за 2019 рік у 2022 можна вважати застарілими. Існують мовні та технічні аспекти SEO-оптимізації, вони входять до структури технології, та залишаються незмінними. Мовний аспект відноситься до підбору ключових слів у запитах користувача, контенту сайта, тощо. Трохи поглибившись у цю галузь, на перший погляд легий спосіб просування сайтів показує його складність, бо контент має бути унікальним, запити – короткими та зрозумілими.

Безпосередня реалізація мовного аспекту – це технічна складова SEO-оптимізації. Технічна сторона включає в себе складання HTML-коду з урахуванням основних слів та фраз, тегів, URL-адрес, тощо.

При створенні сайту компанії/організації/фірми використовується складний та індивідуальний комплекс маркетингових мір по просуванню бренду та предмету діяльності суб'єкта. При правильному підході розвитку, в сукупності з використанням технології пошукової оптимізації сайт починає представляти собою потужний інструмент маркетингового напрямку компанії. Сам сайт можна розглядати як засіб просування, то SEO можна розглядати як технологію покращення ефективності сайту, а не тільки як технологію просування. Таким чином, на сьогоднішня тема моєї дипломної роботи в край актуальна.

Ціль кваліфікаційної роботи – створити програмний засіб який буде реалізувати SEO-аналіз сайтів для подальшої їх оптимізації.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

1.1.1. Основні поняття пошукової оптимізації

Згідно групі Forrester Research Inc., 84% людей шукають та знаходять потрібний їм веб-ресурс за допомогою браузерів та інших пошукових систем згідно [1]. Згідно отриманих даних саме коректна SEO-оптимізація допомагає залучати велику кількість людей на сайт. Найбільш підходяща цільова аудиторія це як раз люди, котрі знайшли сайт через пошукову систему, адже вони вводять до пошукового запиту слова, які формують пошукову оптимізацію. Саме тому можна стверджувати, що SEO має величезну роль інструмента інтернет-меркетингу.

Для того що б зрозуміти суть та принцип дії цього інструменту, потрібно дати йому визначення а також визначити цілі з функціями пошукової оптимізації сайтів.

З англійської SEO (Search Engine Optimization) перекладається як «оптимізація для пошукових систем» або «оптимізація пошуку».

Пошукова оптимізація (англ. search engine optimization, SEO) – це комплекс заходів щодо внутрішньої та зовнішньої оптимізації для підняття позицій сайту в результатах видачі пошукових систем за певними запитами користувачів, з метою збільшення мережного трафіку (для інформаційних ресурсів) та потенційних клієнтів (для комерційних ресурсів)) та наступної монетизації (отримання доходу) цього трафіку. SEO може бути орієнтоване на різні види пошуку, включаючи пошук інформації, товарів, послуг, зображень, відеороликів, новин та специфічні галузеві пошукові системи, згідно з [2].

Зібравши визначення з багатьох доступних ресурсів та проаналізувавши отримані дані, можна зробити висновок щодо тлумачення значення SEO-оптимізація.

SEO-оптимізація – це основний засіб, який використовують для просування сайтів у мережі Інтернет, а його суть – підвищення рангу сайту поміж інших зі схожою тематикою.

У SEO - оптимізації є цілі, задачі, функції та види.

Ціль SEO-оптимізації, виходячи з тлумачення, це просування сайту. Увесь цей комплекс дії приймається збільшення кількості користувачів сайту та клієнтів, які будуть давати прибуток або для інших цілей які переслідує компанія.

Задачі пошукової оптимізації згідно з [3]:

- оптимізація семантичного ядра та текстів сайту;
- організація підготовки сайту згідно певним правилам;
- створення такого контенту та текстів які напряму відносяться до тематики сайту.

Основні функції які виконує пошукова оптимізація згідно з [4]:

- підвищення популярності ресурсу;
- конвертація відвідувачів у покупців та постійних користувачів;
- пошукове просування.

Говорячи про види SEO-оптимізації, виділяють два основних: білий та чорний.

Білий – це стандартні та «правильні» методи реалізації SEO-оптимізації. Правильні, тому що вони не заборонені будь-якими правилами пошукових систем та не є шахрайством.

Чорний використовують для досягнення цілі не зважаючи уваги на засоби які використовуються, тому дуже часто блокується пошуковими системами.

Приклади «**Чорної**» пошукової оптимізації згідно [5]:

1) Спамдексинг: метод просування за найпопулярнішими ключовими запитами, при якому тексти сторінок надмірно насичуються певними ключовими словами, які часто не належать до тематики сайту.

2) Лінкбомбінг: штучне нарощування посилальної маси. Сьогодні цей метод використовується в більш цивілізованому вигляді - розміщення платних посилань на спеціальних біржах - і відноситься, швидше, до сірих методів.

3) Дорвей: При переході з пошукової системи на такий сайт ви моментально перенаправляєтесь на інший ресурс, який не відповідає тематиці вашого запиту.

4) Свопінг: метод, при якому вміст сторінки повністю замінюється на інший після того, як сайт був проіндексований пошуковою системою за певними запитами.

5) Клоакінг: маскування вмісту сторінки від пошукових роботів. За допомогою скрипту на сайті обчислюється IP-адреса пошукового робота, для якої тут же пропонується змінений контент. У той час як звичайні відвідувачі бачать іншу версію, яка не відповідає їхнім запитам.

Використання «чорних» методів оптимізації заборонено та обчислюється автоматично або встановлюється за скаргою до адміністрації пошукової системи. І в тому, і в іншому випадку сайту загрожує виняток із пошукової видачі з усіма наслідками. Сьогодні ці методи практично зжили себе і застосовуються лише для короткочасного нагону трафіку. Часто ними користуються ресурси із сумнівним чи навіть протизаконним змістом.

Види «Білого» SEO:

1) Оптимізація сайту: включає комплекс заходів щодо поліпшення якості контенту, навігації, вмісту html-коду. Зокрема, складається грамотне семантичне ядро у повній відповідності до тематики сайту. Використовується унікальний та змістовний контент, що повністю відповідає тематиці та не перенасичений ключовими запитами. Проводиться внутрішнє перелічення сторінок (проставляються посилання на розділи), підбираються якісні домени та піддомени, URL внутрішніх сторінок (що містять основні ключові запити,

написані транслітом). Відстежуються неякісні посилання та некоректні відповіді сервера на звернення до розділу (у разі помилки запиту, падіння хостинг-сервера, видалення сторінки тощо). Особлива увага приділяється складання файлу robots.txt, призначеного для повідомлення інформації про сайт пошуковим роботам, а також заповненню МЕТА-тегів у html: Title, h1, Decscription, Keywords.

2) Просування сайту: розміщення інформації та посилань на сайт на тематичних ресурсах, проведення рекламних кампаній, публікація якісних статей та інші засоби природного нарощування посилального оточення.

«Білі» методи пошукової оптимізації – офіційно дозволені та допустимі пошуковими системами. Вони полягають, перш за все, у грамотній роботі над сайтом, його зручністю та інформативністю для користувача, з дотриманням рекомендацій пошукових систем. Такі методи часто використовуються в рамках комплексного інтернет-маркетингу та виключають штучний вплив на позиції сайту у пошуковій видачі. Ефект від таких методів буде стабільним та тривалим, без ризику будь-яких санкцій з боку пошукових систем. Крім того, «біле» SEO забезпечить вашому сайту унікальний та інформативний контент, зробить його привабливішим для відвідувачів. Що, без сумніву, позначиться на конверсії.

Ранжування сайту

SEO-оптимізація – це сукупність всіх факторів для просування сайту, об'єднуючи зовнішню та внутрішню оптимізації ми отримаємо ранжування сайту.

Ранжування – це сортування інтернет-ресурсів у пошуковій видачі, у ході якого, найбільш оптимізовані та релевантні сайти та сторінки, які найточніше відповідають на пошуковий запит, потрапляють на перші сторінки браузеру.

Мета ранжування – визначити які ресурси мають максимальну практичну цінність для користувача за допомогою певних методів.

У сьогодення пошукові системи ураховують безліч факторів, які впливають на оптимізацію.

Ранжування має два основних типи складових факторів – зовнішні та внутрішні.

Зовнішні фактори ранжування - це фактори, які впливають на просування сайту і на які вебмайстер не може безпосередньо вплинути за допомогою маніпуляцій над сайтом, що просувається. Або фактори, що враховуються пошуковою системою але які не відносяться до якостей сайту згідно [6].

Внутрішні фактори ранжування - це показники якості сайту, які вебмайстри (або SEO-фахівці) можуть безпосередньо контролювати. Їх можна розділити на оптимізацію окремо взятої сторінки та оптимізацію показників, які стосуються сайту в цілому. Іншими словами, це сукупність критеріїв, які впливають на оцінювання пошуковими системами згідно [6].

Розберемо що входить до внутрішніх та зовнішніх факторів ранжування сайтів.

До внутрішніх факторів відноситься:

1) Наповнення та контент сайту, його якість, актуальність.

Під актуальністю контенту найчастіше розуміють його новизну, частоту оновлення та унікальність. Також у контенту є такий критерій, як ступінь неприродності частоти вживання ключових слів.

2) Структура сайту та навігація по ньому.

На даний момент кожен сайт має деревовидну структуру. Її суть у тому, що структура схожа на дерево з гілками: головна сторінка – стовбур дерева, з нього можна потрапити на інші розділи веб-ресурсу, як по гілкам. При проектуванні структури сайту ураховується її юзабельність, наприклад – структура дозволяє користувачу легко орієнтуватись усередині структури, та може за максимум три кліки дійти до сторінки яка йому потрібна.

3) Мета-теги сторінок веб-ресурсу.

Пошукові роботи реагують позитивно, коли вміст метатегів сторінки відповідає її змісту. Наприклад, якщо в метатегі keywords сторінки вказано:

«купити будинок, будинки у Києві», то й текст на сторінці має бути про те, як купити будинок у Києві.

4) Зовнішні посилання на інші веб-ресурси.

Наявність на сайті нетематичних посилань на інші сайти може погіршити його ранжування у пошукових системах. Сучасні алгоритми аналізу контенту дозволяють пошуковим роботам досить ефективно розпізнавати рекламу на сайті навіть у завуальованому вигляді. Велика кількість реклами ігнорується для сайтів з великою вагою та авторитетністю, але може негативно вплинути на молодий інтернет-ресурс.

До зовнішніх факторів ранжування відноситься:

1) Зовнішні посилання на сайт.

Зовнішні посилання – це посилання на інших веб-ресурсах, які посилають на ваш сайт.

2) Згадування сайту в авторитетних каталогах.

Каталоги сайтів являють собою певним чином структурований набір посилань, які відсортовані згідно темам.

Каталоги поділяють на 4 типи згідно [7]:

1) Закриті – операції над цим каталогом може проводити тільки одна відповідальна за нього людина.

2) Білі каталоги – являють собою вже авторитетними майданчиками у мережі Інтернет. Дуже часто, веб-ресурси, які додають у такі каталоги рахуються краще за інші.

3) Сірі каталоги – надають можливість публікувати посилання на веб-ресурси.

4) Чорні каталоги – у такому каталозі посилання йдуть через редіректи або неактивні посилання, також на цей каталог необхідно також надавати зворотне посилання.

Існують такі фактори, які не можна віднести до будь-якої групи. Наприклад домен. Ранжування сайту відповідає рівню його домена. Іншим

фактором є географічне розташування. Пошукові системи ранжують домени згідно до розташування користувача.

З'ясувавши що таке ранжування сайтів та розібравши його аспекти та фактори, можна зробити такий висновок: мовний аспект та технічну складову складно віднести до певної групи факторів ранжування, хоча SEO і є здебільш внутрішньою оптимізацією веб-ресурсу. Успіх веб-ресурсу здебільш залежить від нього самого, на це впливають внутрішні фактори(такі як структура, контент, його унікальність та релевантність). Доцільно буде роздивитися ретельніше внутрішні фактори SEO-оптимізації – аспекти пошукової оптимізації.

1.1.2. Аспекти SEO-оптимізації

Семантичне ядро сайту та контент розміщений на сайті як основа мовного аспекту SEO.

Для початку реалізації семантичного ядра сайту, необхідно підібрати ключові слова – основи майбутнього успіху сайту.

Сформуємо загальну загальну концепцію поняття «семантичне ядро», зараз необхідно дати йому тлумачення. Використавши інформацію з декількох джерел можна сформулювати таке формулювання:

Семантичне ядро сайта - це список ключових слів та словосполучень, що наводять на сайт цільових відвідувачів, пошукових робіт та використовуються для просування сайту в пошукових системах згідно з [8].

Для коректної реалізації семантичного ядра необхідно розуміти спрямованість та тематику веб-ресурсу, бути експертом у предметній галузі, розбиратися у продукті та предметі діяльності компанії. Якщо семантичне ядро даного веб-ресурсу створено правильно, з дотриманням до тематики, буде відповідати запитам користувачів – сайт буде користуватися популярністю та виконувати свою основну задачу.

Ключові слова підбирають, згідно тематиці сайту, а також використовуючи семантичні ядра конкурентів згідно [9]. Головне завдання — створити на сайті сторінки для всіх запитів, які максимально повно відповідають потребам цільової аудиторії.

При підборі ключових слів, доцільно буде використовувати існуючі інструменти. Першим на думку спадає ідея вигадання потенційних запитів. Основний мінус цього способу – людський фактор, усіх запитів не вигадати та через деякий час оптимізатор перебере усі запити створені вручну. Саме через проблеми людського фактору та необхідності автоматизації створення семантичного ядра сайту були створені застосунки для реалізації цієї задачі. Мова іде про Google AdWords - створена для підбору тексту контекстної реклами [10].

Входячи на ринок послуг та товарів, компанія з вірогідністю 100% зустріне конкуренцію, саме конкуренти можуть не по своїй волі допомогти створити власне семантичне ядро. Не рідко компанії прибігають до аналізу фраз, які використовують конкуренти.

Не зважаючи на методи, які використовуються при створенні семантичного ядра сайту, є певні критерії та вимоги які дотримуються при його реалізації:

- 1) Створити коректну форму запиту для просування.
- 2) Не використовувати слова які не несуть сенсу в рамках запиту.
- 3) Відокремити ключові слова, необхідні тільки для накрутки.

Створюючи семантичне ядро сайту, його власник закладає основу, фундамент, при правильному та ретельному формуванні якого, отримується задовільний результат.

Технічний аспект SEO-оптимізації

Основним компонентом будь-якого сайту та веб-ресурсу є його технічний аспект. Зокрема великої кількості функціональних модулів, красивого оформлення та контенту, важлива й технічна оптимізація, яка забезпечить

роботу роботів пошукових систем, котрі індексують контент. Загальна суть технічного аспекту обумовлена сукупністю елементів, які впливають на роботу пошукових систем, так що проводиться правильна індексація та ранжування веб-ресурсу. Елементи, які створюють складний взаємозв'язаний комплекс можна поділити на декілька груп: основні (легальні) SEO-засоби, пошуковий спам та допоміжні (додаткові) засоби.

До основних (легальних) засобів SEO-оптимізації можна віднести, згідно з [11]:

- теги ключових слів;
- тег <TITLE>;
- основний текст сторінки;
- розсташування текст;
- заголовки;
- доменне ім'я;
- елементи веб-дизайну;
- CSS;
- зовнішній вигляд;
- текст посилань;
- рівень папок;
- меню та карта сайту;
- виділений IP-адрес;
- тематика;
- розширення сайтів.

До пошукового спаму входить:

- redirects;
- невидимий текст;
- теги форм сховані;
- повторна реєстрація сайту.

Допоміжні засоби включають:

- тег опису;

- тег коментарів;
- фрейми.

При реалізації технічного аспекту SEO-оптимізатор використовує класичні засоби та інструменти, які допомагають вивести сайт у топ пошукових систем. Одним з таких інструментів є кабінети вебмайстрів (Google Analytics + Google Webmaster Tools) та лічильники. Коли починають роботу над задачею просування веб-ресурсу, реєструються у кабінетах вебмайстрів, це служить найкращим сигналом для пошукових систем. Невдовзі після реєстрації у вебмайстра, новостворений сайт починають відвідувати «роботи». Але, тільки додати домен буде замало, кабінети вебмайстрів необхідно налагодити саме для вашого веб-ресурсу.

Багато виконати як мінімум такі важливі дії:

- 1) вказати регіональну належність.
- 2) перевірити коректність та правильність створення robots.txt.
- 3) додати файл карти сайту (sitemap.xml) до необхідного розділу.

Згадуючи про роботів та robots.txt важливо розуміти із себе представляють ці два поняття, та як вони пов'язані із пошуковою оптимізацією.

Пошуковий робот — програма, призначена для перебору сторінок Інтернету з метою занесення інформації про них до бази даних пошукової системи згідно [12].

Розібравшись із значенням цього терміну, необхідно виділити основну суть, яку несе цей засіб. Основна концепція полягає у тому, що «роботи» заходять на веб-ресурс/сайт, заносять вміст сайту до індексу та знаходять посилання, по яким переходять на інші сторінки.

Коли суть «роботів» з'ясована, варто звернути увагу на robots.txt. Robots.txt – це технічний файл, який включає в себе правила для індексації цільового сайту «роботами», та є першим файлом до якого вони звертаються потрапивши на сайт. Якщо сказати простіше, то це файл до якого занесені рекомендації, які вказують, що можна індексувати, а що ні. Ці правила та рекомендації формуються в залежності від системи управління, на якій працює сайт. Але, основний перелік того, що необхідно бути у файлі виглядає так:

1) Необхідно вказати правила для різних роботів, а не тільки одного (User-agent: GoogleBot, User-agent: Yandex, тощо.). Це робиться із-за того, що правила та директиви для роботів можуть бути різними.

2) Реалізується заборона індексації усіх системних ресурсів (шаблонів, виконавчих файлів, тощо).

3) Усе що не повинно знаходитися у пошуку закривається.

4) Вказана карта сайту sitemap.xml.

5) Для роботи роботу GoogleBot необхідно відкрити файли, впливаючі на відображення.

Одже, який висновок можна зробити – файл robots.txt використовується для управління трафіком пошукових роботів на вашому сайті. Завдяки йому можна частково впливати на поведінку ботів пошукової системи, які, в свою чергу, реалізують передачу даних про сайт на сервер, та завдяки цьому просувають або ні, сайт у пошуковій системі.

Після такого важливого інструменту, як «роботи», буде доцільно згадати про карту сайту, по якій вони орієнтуються.

Карта сайту – це файл у якому знаходиться інформація про те, як на сайті розподілений та організований увесь контент. Для роботів необхідно щоб карта сайту формувалась у форматі .xml і повинна включати усі посилання, які будуть індексуватися. Також у цьому файлі можна вказати які розділи сайту більш важливі та повідомити додаткову інформацію про них.

Варто розібрати випадки, коли файл sitemap.xml є необхідним та використовується згідно з [13]:

- **Дуже великий розмір сайту.** Це означає що може трапитися так, що роботи Google не заіндексують новостворені або нещодавно змінені сторінки сайту.

- **Велика кількість сторінок які ізольовані або не зв'язані.** Для їх коректного сканування їх можна перелічити у sitemap.xml.

- **Якщо сайт новий, та має не багато зовнішніх посилань на нього.** Робота роботів Google являє собою перехід від однієї сторінки до іншої завдяки посиланням. Якщо на сайт вказує маленька кількість зовнішніх посилань, то його знаходження стає проблемою.

- **Велика кількість медіа або новостного контенту на сайті.** Пошукова система за допомогою sitemap.xml може отримувати необхідну додаткову інформацію для відображення у результатах пошуку.

Одже, можна зробити висновок, що карта сайту є необхідним елементом сайту, для його просування та оптимізації за допомогою роботів різних пошукових систем. Карта сайту спрощує орієнтацію по сайту для цих роботів, а одже й користувачам.

Згадаємо як працюють роботи – вони переходять по зовнішнім посиланням та індексують внутрішній контент, пересуваючись по сайту завдяки карті, до якої внесений перелік внутрішніх посилань. Але може трапитися що посилання некоректне – «**бите посилання**» яке видає помилку 404. Якщо робот знаходить таке посилання, то теж подає інформацію на сервер, що призводить до пониження рейтингу веб-ресурсу серед інших. Саме тому необхідно позбавлятися таких посилань.

Як саме впливають такі помилки на рейтинг:

Падіння позицій. Роботи, які при скануванні сайту потрапляють на безліч неіснуючих сторінок, починають розцінювати такий блог або інтернет-магазин, як неякісний.

Зниження краулінгового бюджету. Для кожного сайту в інтернеті встановлюється ліміт веб-сторінок, які сканує робот протягом дня. При появі битих посилань значення цього ліміту зменшується. В результаті нові сторінки

індексуються не відразу після їх появи на онлайн-ресурсі, а через кілька днів або навіть тижнів.

Наявність та кількість посилань з 404 кодом залежить, переважно, від людського фактору, тому і знайти їх буває іноді досить складно, ще складніше здогадатися про їх існування. Перевірити їх наявність можна декількома способами:

- **вручну** – пребираючи кожне посилання, але це занадто довго;
- **автоматично** – за допомогою сторонніх застосунків.

На посилання реагують не тільки роботи, а й користувачі веб-ресурсу, саме для користувачів необхідно використовувати SEF URL. SEF (Search Engine Friendly URL), або посилання яке легко сприймається людиною згідно з [14]. Перед тим як давати тлумачення важливо зрозуміти, для чого такі посилання потрібні. Візьмемо приклад неіснуючого посилання:

<https://www.exampleurlink.com/index.php?cPath=643>

Користувачу буде не зрозуміло у якому підрозділі веб-ресурсу він знаходиться, та яке в нього призначення. Такий вигляд посилання створює більшість CMS при створенні сторінок - хаотичний набір символів, які не несуть в собі ніякого сенсу. Вони погіршують зовнішній вид сайтів в очах користувача, що сказується також негативно на ранжуванні. Саме тому важливо використовувати user-friendly URL. Френдлі посилання, це такі посилання, які зручні для сприйняття людиною та пошуковою системою, та точно передають суть вкладену до них. Такі посилання повинні логічно описувати структуру сайту та шлях до сторінки.

Приклад нового френдлі посилання, яке відповідає усім критеріям, які були описані вище:

<https://www.exampleurlink.com/oblast/city/Dnipro>

Для чого необхідно використовувати SEF URL`s та їх переваги:

- Покращення позицій сайту при видачі пошуковою системою;
- Правильно налаштовані SEF посилання не відштовхують користувачів;
- Підвищують зручність користування ресурсом;

Висновок:

Зробивши висновки, та проаналізувавши усе вищеописане ми ретельно проаналізували суть понять, виявили цілі, задачі та обізначили цілі SEO-оптимізації. Тепер можна стверджувати, що SEO-оптимізація у сьогодення є основним інструментом просування сайтів та розвитку інтернет-ресурсів як незамінний елемент інтернет-маркетингу.

У цьому розділі також з'ясували суть поняття ранжування - упорядкування результатів пошуку з їхньої релевантності, іншими словами назначеня сайтам та веб-ресурсам певного порядку або рангу, за якими вони будуть відображатися у пошукових системах згідно певного пошукового запиту. З'ясували фактори оптимізації які виділяють, зробили опис мовного та технічного аспектів пошукової оптимізації. Сутність мовного аспекту – вибір певних слів та фраз, на які буде орієнтуватися пошукова система при видачі результатів пошукового запиту, що, у першу чергу, впливає на ранжування веб-ресурсу. Також і головну роль грає наповнення сайту, контент, його унікальність та якість, що впливає на ранжування на рівні із мовним аспектом. Але й не слід забувати про технічну складову, адже при не коректній або слабко реалізованій роботі її компонентів, усі зусилля при створенні семантичного ядра та наповненні сайту контентом зведуться ні та будуть марними.

1.2. Призначення розробки та галузь застосування

Призначення розробки – створення створення десктопного застосунку, для реалізації можливості пошуку помилок та недоліків технічного аспекту SEO-оптимізації.

Додаток може бути використаний для пошуку недоліків технічного аспекту пошукової оптимізації, власниками компаній/фірм та приватними особами, які мають свій власний сайт або веб-ресурс.

1.3. Підстави для розробки

Відповідно до ОПП, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу (проект). Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- ОПП за спеціальністю 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» No 268-с від 18 . травня 2022 р;
- завдання на дипломний проект на тему «Розробка десктопного додатку для SEO аналізу сайтів».

1.4. Постановка завдання

Непомітні або не явні, всі недоліки занижують рейтинг сайту поміж інших через не якісно проведену технічну підготовку сайту під час експлуатації або у період реалізації. Найскладніше, це знайти недолік та помилку, а виправити можна за декілька хвилин. Більшість засобів орієнтовані на пошук певного недоліку, а не великої кількості одразу

Метою даної роботи є створення десктопного додатку, який дозволяє знайти недоліки, які були зроблені при реалізації технічного аспекту SEO-оптимізації. Він аналізує та знаходить більшість проблем які заважають просуванню сайту. Цей додаток буде зручним, адже буде знаходити більшість недоліків одразу, за короткий проміжок часу та видаватиме результат у зручному форматі.

Продукт повинен працювати на усіх популярних платформах (Windows, MacOS, Linux), та всіх актуальних десктопних пристроях. Продукт має давати можливість користувачу зареєструватися та має належним чином захищати інформацію, надану користувачем.

1.5. Вимоги до програми або програмного виробу

Головною вимогою роботи є можливість використання додатку на всіх платформах, мати задовільну швидкість виконання операцій та видавати коректний результат роботи. Алгоритми парсингу сторінок сайту мають задіяти як умога менше ресурсів комп'ютеру, на якому працює додаток.

1.5.1. Вимоги до функціональних характеристик

До додатку, який розробляється в ході кваліфікаційної роботи, висовується наступні функціональні вимоги:

- Можливість реєстрації користувача;
- Можливість логіну користувача;
- Можливість обрати сайт який буде аналізуватися;
- Можливість придбати просунену версію додатку;
- Можливість завантажувати звіти роботи додатку;
- Зручний та зрозумілий інтерфейс;
- Можливість обрати мову додатку (англійська/українська/російська);
- Можливість змінити пароль;

1.5.2. Вимоги до інформаційної безпеки

Додаток має передавати дані користувача використовуючи спеціальні техніки, щоб третя сторона не могла отримати доступ до особистих даних користувача. Профіль має бути захищений паролем та логіном, який буде знати лише користувач. Пароль та логін користувача мають зберігатися лише на серверній частині додатку.

1.5.3. Вимоги до складу та параметрів технічних засобів

Додаток не є вимогливим до конфігурації пристрою, на якому працює. Основна вимога – наявність стабільного інтернет з'єднання та бажано програм, які будуть відкривати звіт роботи додатку.

Для нормального функціонування додатку необхідно мати такі мінімальні системні характеристики пристрою:

- процесор з частотою не менш 2,4GHz;
- оперативна пам'ять 4Gb;
- жорсткий диск 10 Gb;
- операційна система MacOS, Windows 10, Linux;

Програмне забезпечення: встановлена на пристрої Java (мінімальної версії 11.0.13).

Характеристики, що наведені вище є задовільною рекомендацією, це означає, що за наявністю характеристик не нижче наведених раніше, додаток буде працювати відповідно до усіх вимог, які обумовлені із замовником.

1.5.4. Вимоги до інформаційної та програмної сумісності

Для запуску додатку необхідні такі компоненти:

- Будь-яка стабільно працююча операційна система;
- Встановлена Java (бажано останньої версії).

Для розробки додатку необхідні наступні компоненти:

- Середовище розробки (IntelliJ IDEA, Eclipse або інш.);
- Візуальний інструмент дизайну SceneBuilder;
- JavaFX;
- Бібліотека IText;
- Бібліотека Jsoup;
- Бібліотека ControlsFX;

- Бібліотека MongoDB;
- Складальник проекту launch4j;
- Бібліотека OpenCSV;
- Бібліотека log4j;
- Json;
- Додаток launch4j;
- Бібліотека Apache POI.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Призначення розробки системи – створення десктопного додатку для SEO-аналізу сайтів.

Розроблений в ході виконання кваліфікаційної роботи додаток має наступні функції та призначення:

- Зручний та інтуїтивно зрозумілий інтерфейс;
- Можливість реєстрації;
- Здійснювати SEO-аналіз певного сайту;
- Мати стабільний робочий стан;
- Давати можливість користувачу завантажувати звіт роботи додатку;
- Відображати результат роботи у спеціальних таблицях з результатами;
- Можливість обрати преміум версію та придбати її.

Додаток може бути використаний будь-яким SEO-оптимізатором або власником сайту для пошуку недоліків, які негативно впливають на ранжування сайту у мережі Інтернет.

2.2. Опис застосованих математичних методів

Оскільки особливості предметної області розв'язуваної задачі не передбачають застосування математичних методів, при розробці десктопного додатку математичні методи не використовувалися.

2.3. Опис використаних технологій та мов програмування

Перелік основних технологій використаних для реалізації додатку:

- 1) Java – мова програмування використана для розробки додатку.
- 2) JavaFX – набір інструментів для реалізації графічної складової додатку.
- 3) SceneBuilder – інструмент для конструювання інтерфейсу.
- 4) Maven – конструктор проекту.
- 5) Jsoup – бібліотека для роботи із HTML-файлами.
- 6) GitLab – репозиторій для проекту.
- 7) ITEXT – бібліотека для роботи з PDF файлами.
- 8) Apache POI – бібліотека для роботи з .XLSX файлами.
- 9) MongoDB – для зберігання даних про користувача.
- 10) Log4j – бібліотека для логування.

Java

Мова розробки **Java** об'єктно-орієнтована, існує більше 25 років та здобула велику довіру серед розробників усього світу. Не один рік ця мова визнавалася найпопулярнішою поміж інших мов програмування.

Основними сильними сторонами Java є кросплатформенність, ком'юніті, надійність, об'єктноорієнтованість, відносна простота, кількість зовнішніх бібліотек та модулів.

Java – достаньо популярна мова: єю користується велика кількість розробників, та рішення майже кожної великої проблеми, яка може з'явитися при роботі з Java, вже хтось винайшов. Завдяки тисячам бібліотек та форумів, можна знайти готове рішення майже кожної існуючої проблеми та задачі. На GitHub, наприклад, є відкриті проекти та документації, а на форумі Stack Overflow можна звернутися за допомогою (рис. 2.1).

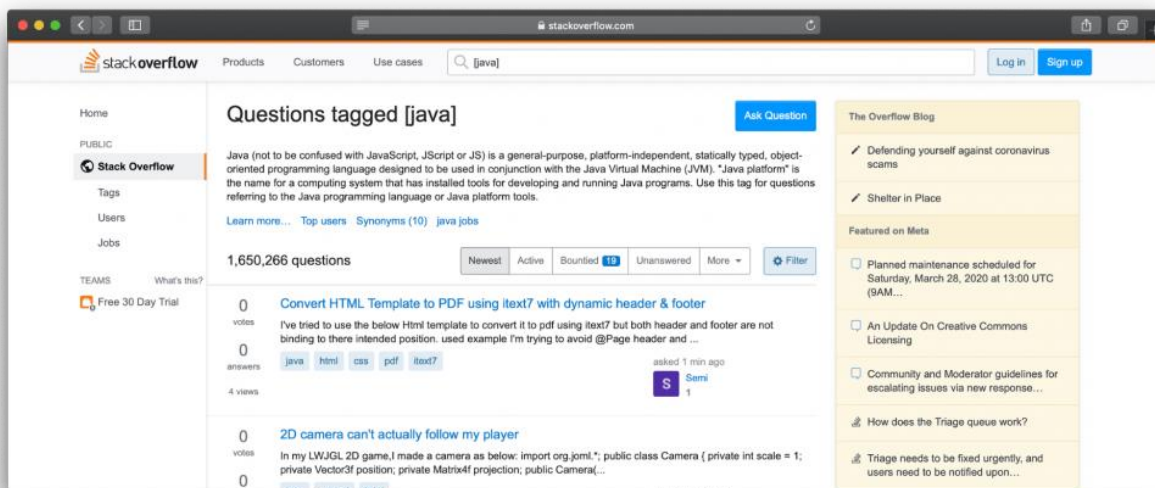


Рис. 2.1. Ком'юніті на Stackoverflow.com

Усі бібліотеки, написані коли-небудь для Java, - це класи, які відповідають за функціональність мови. Будь-який додаток на Java – набір класів, які описують різні об'єкти. Це добре, тому що дозволяє створювати складні програми, але які легко підтримувати. І в цілому Java – мультипарадигмова мова, іншими словами підтримує більшість принципів програмування, що дозволяє ефективно вирішувати різні задачі.

Ще однією перевагою Java є її відносна простота. З Java не так просто почати роботу, як з Python, але він набагато простіше C або C++. Під час розробки Java було спрямована велика кількість уваги простоті мови розробки ПЗ, саме тому програми на Java, якщо порівнювати з програмами на інших мовах, легше писати, компілювати, відлагоджувати та вивчати.

Maven

Maven - фреймворк для автоматизації складання проектів на основі опису їх структури у файлах мовою POM (Project Object Model), що є підмножиною XML. Проект Maven видається спільнотою Apache Software Foundation, де формально є частиною Jakarta Project (рис. 2.2) згідно з [15].



Рис. 2.2. Логотип Maven

Maven забезпечує складання проекту. У файлах опису проекту міститься його специфікація, а не окремі команди виконання. Всі завдання з обробки файлів, описані в специфікації, Maven виконує за допомогою їх обробки послідовністю вбудованих та зовнішніх плагінів.

Завдяки Maven ми можемо підключати різноманітні бібліотеки та репозиторії для створення цільового проекту. Для того, щоб підключити бібліотеку, необхідно додати залежність до POM-файлу.

Приклад залежності:

```
<dependency>  
    <groupId>org.openjfx</groupId>  
    <artifactId>javafx</artifactId>  
    <version>11</version>  
    <type>pom</type>  
</dependency>
```

JavaFX

Для реалізації UI складової використовувався набір інструментів для створення кросплатформених додатків на основі Java – JavaFX (рис. 2.3).



Рис. 2.3. Логотип JavaFX

Цей інструментарій був створений з ціллю спростити створення додатків. Іншими словами, JavaFX – це декларативна мова сценаріїв зі статичною типізацією. Структура та API-інтерфейси призначені для створення анімації, 2D та 3D-геометрії, діаграм, спеціальних ефектів, кольорових градієнтів, графічних елементів управління та простих маніпуляцій з мультимедіа, включаючи аудіо, відео та зображення. Візуалізація графіки JavaFX користується можливостями апаратного прискорення. Це забезпечує хорошу продуктивність рендерингу графіки та анімації. Також додаток на JavaFX підтримує CSS – стандарт, який використовується багатьма дизайнерами. На стан по 2010 року JavaFX підтримує усі доступні платформи операційних систем.

Особливості JavaFX згідно [16]:

- JavaFX спочатку поставляється з великим набором частин графічного інтерфейсу, таких як будь-які кнопки, текстові поля, таблиці, дерева, меню, діаграми і т.д., що в свою чергу заощадить нам вагон часу;
- JavaFX часто використовує стилі CSS, і ми зможемо використовувати спеціальний формат FXML для створення GUI, а не робити це в кодї Java. Це полегшує швидке розміщення графічного інтерфейсу користувача або зміну зовнішнього вигляду чи композиції без необхідності довго грати у кодї Java;
- JavaFX має готові до використання частини діаграми, тому нам не потрібно писати їх з нуля будь-коли, коли вам потрібна базова діаграма.

SceneBuilder

Допоміжним засобом створення GUI став застосунок Scenebuilder. У JavaFX **SceneBuilder** – це інструмент, за допомогою якого ми можемо конструювати основні графічні елементи додатку (рис. 2.4). Ця програма на основі сконструйованого результату буде створювати .FXML файл, які ми будемо підтягувати до нашого додатку, це забезпечить зв'язок між кодом та інтерактивними графічними елементами, такими як кнопки, таблиці, тощо.

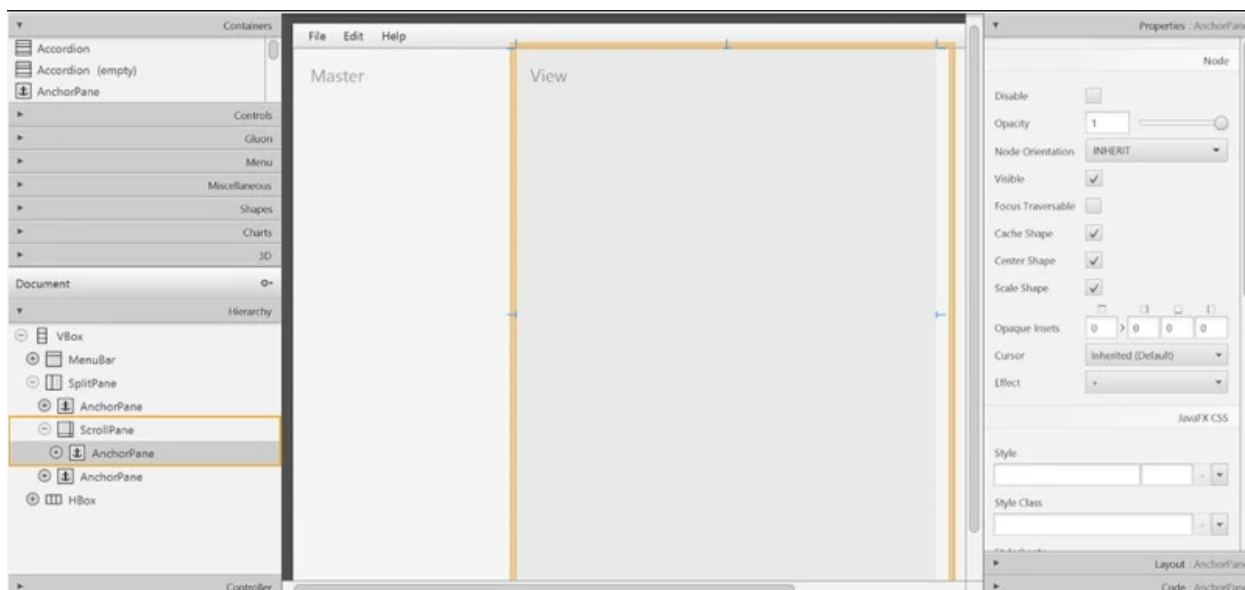


Рис. 2.4. Приклад роботи SceneBuilder

Jsoup

Для реалізації парсингу сайтів та створення головного ядра додатку, використана бібліотека **Jsoup**. **Jsoup** – це бібліотека на основі Java для роботи з контентом на основі HTML. Вона надає дуже зручний API для вилучення та обробки даних, використовуючи найкращі методи DOM, CSS та jquery-подібні. Він реалізує специфікацію WHATWG HTML5 та аналізує HTML у тому ж DOM, що й сучасні браузерери.

Бібліотека **jsoup** реалізує специфікацію WHATWG HTML5 та аналізує вміст HTML у тому ж DOM, що й у сучасних браузерах.

Бібліотека **Jsoup** забезпечує такі функціональні можливості згідно з [17]:

- Підтримка множинного читання - він читає та аналізує HTML, використовуючи URL, файл або рядок;

- Селектори CSS – можуть знаходити та вилучати дані, використовуючи обхід DOM або селектори CSS;

- DOM Manipulation – може маніпулювати HTML-елементами, атрибутами та текстом;

iText

Для реалізації звітів у PDF форматі та їх структури, була використана бібліотека зручна **iText** (рис. 2.5).



Рис. 2.5. Логотип ІТЕХТ

Це бібліотека класів для генерації, аналізу та зміни документів у форматах Portable Document Format (PDF), а також XML, HTML та RTF згідно [18].

Нижче наведено особливості бібліотеки iText [19]:

- **Інтерактивний** – iText надає класи (API) для створення інтерактивних документів PDF. Використовуючи їх, ви можете створювати карти та книги.

- **Додавання закладок, номерів сторінок тощо** – Використовуючи iText, ви можете додавати закладки, номери сторінок та водяні знаки.

- **Розділити та об'єднати** – Використовуючи iText, ви можете розбити наявний PDF-файл на кілька PDF-файлів, а також додати/об'єднати додаткові сторінки.

- **Заповнення форм** – Використовуючи iText, ви можете заповнювати інтерактивні форми у PDF-документі.

- **Зберегти як зображення** – Використовуючи iText, можна зберігати PDF-файли у вигляді файлів зображень, таких як PNG або JPEG.

- **Canvas** – бібліотека iText надає вам клас Canvas, за допомогою якого ви можете малювати різні геометричні фігури у документі PDF, такі як коло, лінія та ін.

- **Створення PDF-файлів** – Використовуючи iText, ви можете створити новий PDF-файл із програм Java. Ви можете включити зображення та шрифти теж.

Apache POI

Що до звітів у .XSLX форматі, була використана бібліотека Apache POI, яка реалізує формування файлів у потрібному нам форматі (рис. 2.6).



Рис. 2.6. Логотип Apache-POI

У розробці корпоративних додатків рано чи пізно доводиться працювати з різними електронними документами: таблицями та текстовими документами (переважно у форматах .xls, .xlsx, .doc, .docx). Java для цих цілей існує чудова бібліотека з відкритим вихідним кодом: Apache POI.

Apache POI це бібліотека Java з відкритим вихідним кодом, наданий Apache, це сильна бібліотека, яка допомагає вам працювати з документами Microsoft, як Word, Excel, Power point, Visio, тощо.

POI це абревіатура "Poor Obfuscation Implementation". Формати файлів Microsoft приховані. Інженери Apache повинні були постаратися зрозуміти, і вони побачили, що Microsoft створила складні формати, коли це не потрібно. І назва бібліотеки має походження від гумору.

Apache POI підтримує вас при роботі з форматами Microsoft, його класи часто мають приставку HSSF, XSSF, HPSF, тощо. Дивлячись на приставки класу, ви можете дізнатися, який формат підтримує цей клас згідно [20].

Jsoup

Jsoup – бібліотека з відкритим вихідним кодом, створена для роботи із даними, які знаходяться усередині HTML документів. Це включає: аналіз, вилучення та управління цими даними (рис. 2.7).



Рис. 2.7. Логотип Jsoup

Бібліотека jsoup забезпечує такі функціональні можливості згідно з [21]:

- Підтримка множинного читання – він читає та аналізує HTML, використовуючи URL, файл чи рядок;
- Селектори CSS – можуть знаходити та вилучати дані, використовуючи обхід DOM або селектори CSS;
- DOM Manipulation – може маніпулювати HTML-елементами, атрибутами та текстом;
- Запобігати атакам XSS – він може очищати контент користувача від заданого безпечного білого списку, щоб запобігти атакам XSS;
- Tidy виводить чистий HTML;
- Обробляє недійсні дані - sup може обробляти незамкнені теги, неявні теги і може створювати структуру документа.

GitLab та контроль версій додатку

Функцію та задачу контролю версій виконував сервіс **GitLab** (рис. 2.8).

GitLab — веб-додаток та система керування репозиторіями програмного коду для **Git**.

GitLab пропонує рішення для зберігання коду та спільної розробки масштабних програмних проектів. Репозиторій включає систему контролю версій для розміщення різних ланцюжків розробки та гілок, дозволяючи розробникам перевіряти код і відкочуватися до стабільної версії софту у разі непередбачених проблем згідно [22].



Рис. 2.8. Логотип GitLab

Можливості GitLab:

- управління (Manage)
- створення (Create);
- перевірка (Verify);
- упаковка (Package);
- безпека (Secure);
- релізи (Release);
- конфігурування (Configure);
- моніторинг (Monitoring);
- захист (Defend).

Гілки проекту (рис. 2.9):

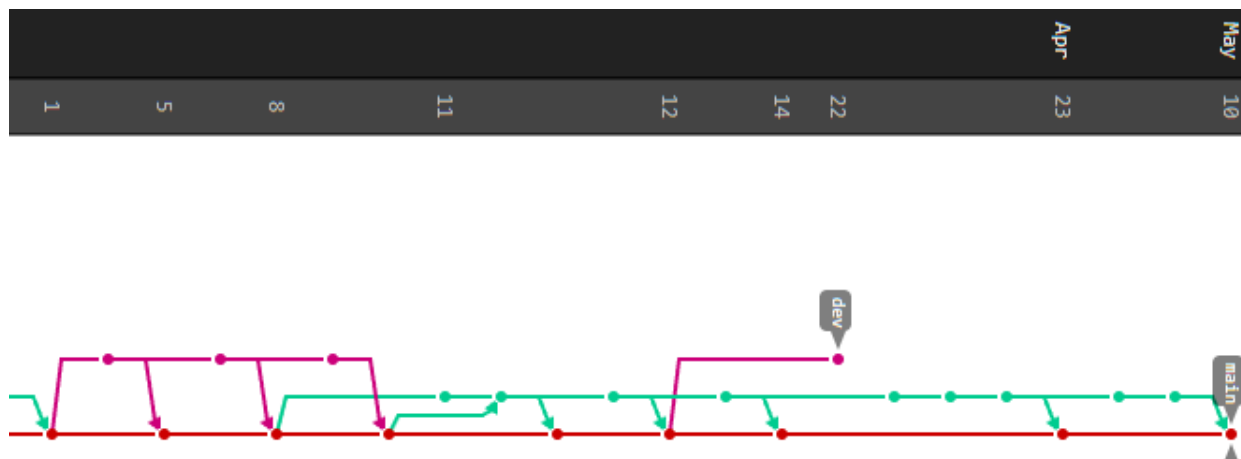


Рис. 2.9. Дерево гілок проекту на репозиторії

Компілятор проектів Launch4j

Для компіляції проекту з .jar файлу до .exe я використав застосунок **Launch4j** (рис. 2.10).



Рис. 2.10. Логотип launch4j

Launch4j - інструмент для упаковки Java-додатків, із вигляду jar-файлів, в власні файли Windows, що виконуються. Виконуваний файл можна налаштувати для пошуку певної версії JRE або використання пов'язаної версії. Оболонка також забезпечує кращий інтерфейс користувача завдяки значку програми, власному екрану-заставці до JRE і сторінці завантаження Java на випадок, якщо відповідна JRE не може бути знайдена згідно [23].

MongoDB

MongoDB – це система що використовується для управління нереляційними базами даних. Основою перевагою цієї системи є використання замість таблиць та схем – документів або колекцій.

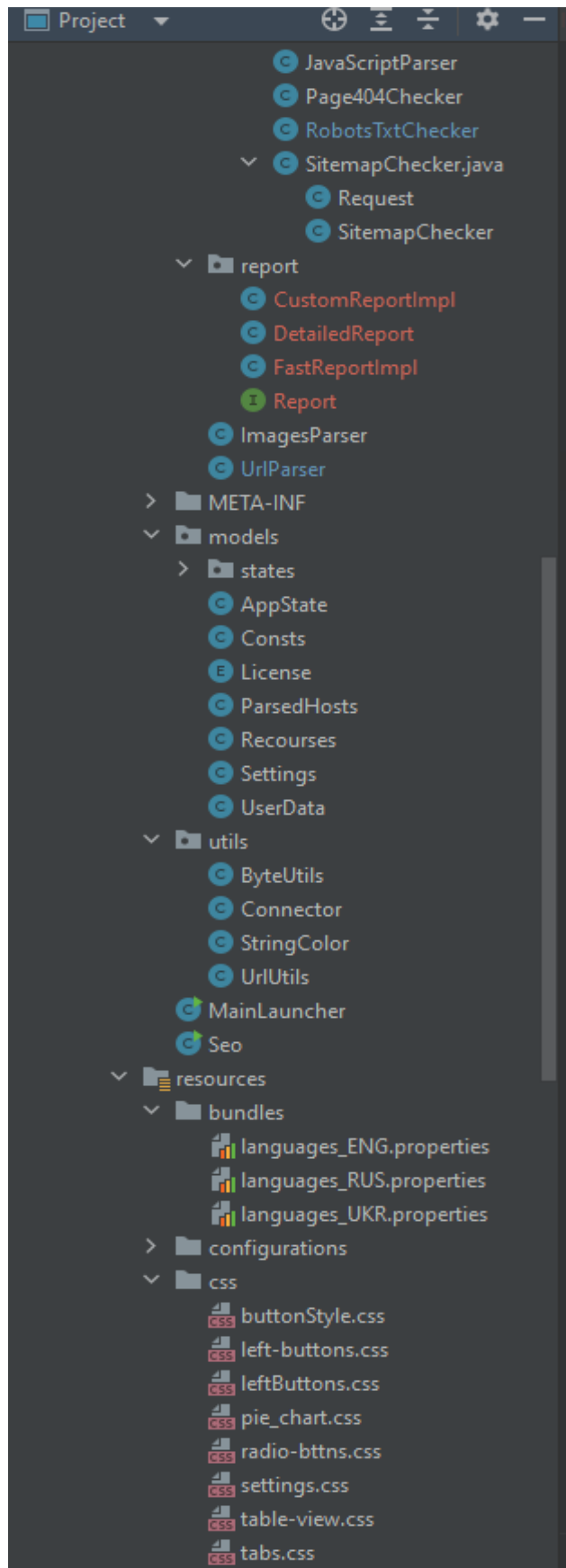


Рис. 2.12. Пакети та класи проекту

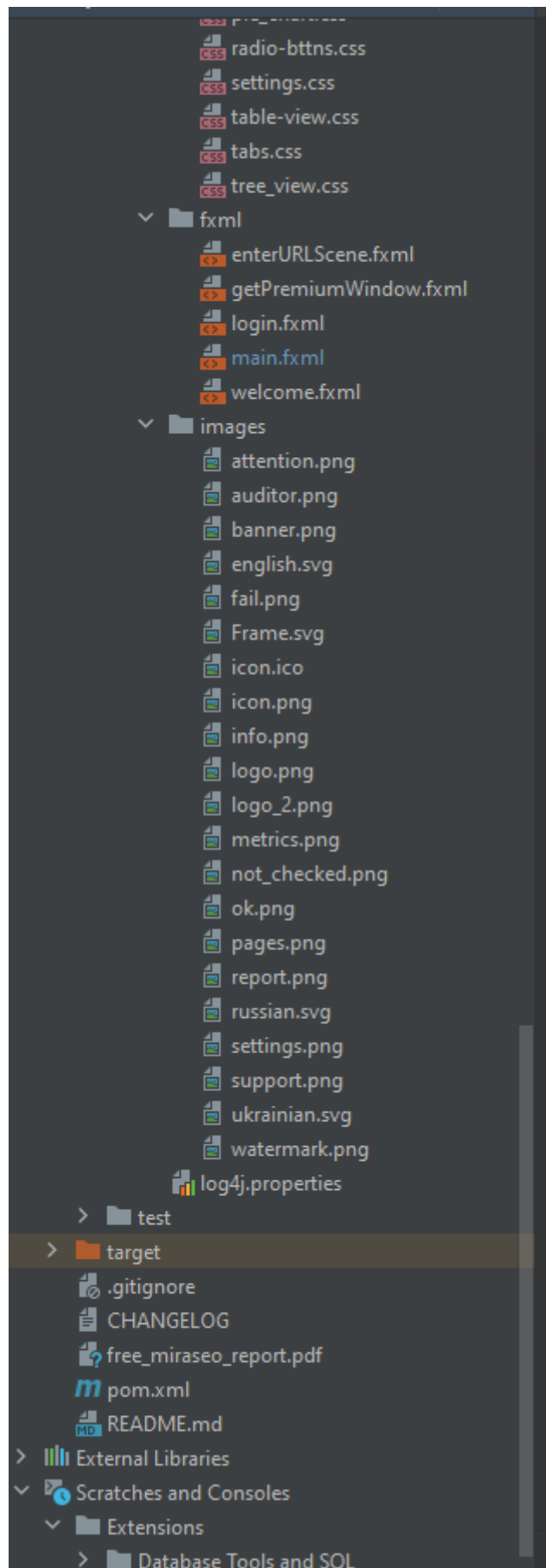


Рис. 2.13. Пакети та класи проекту

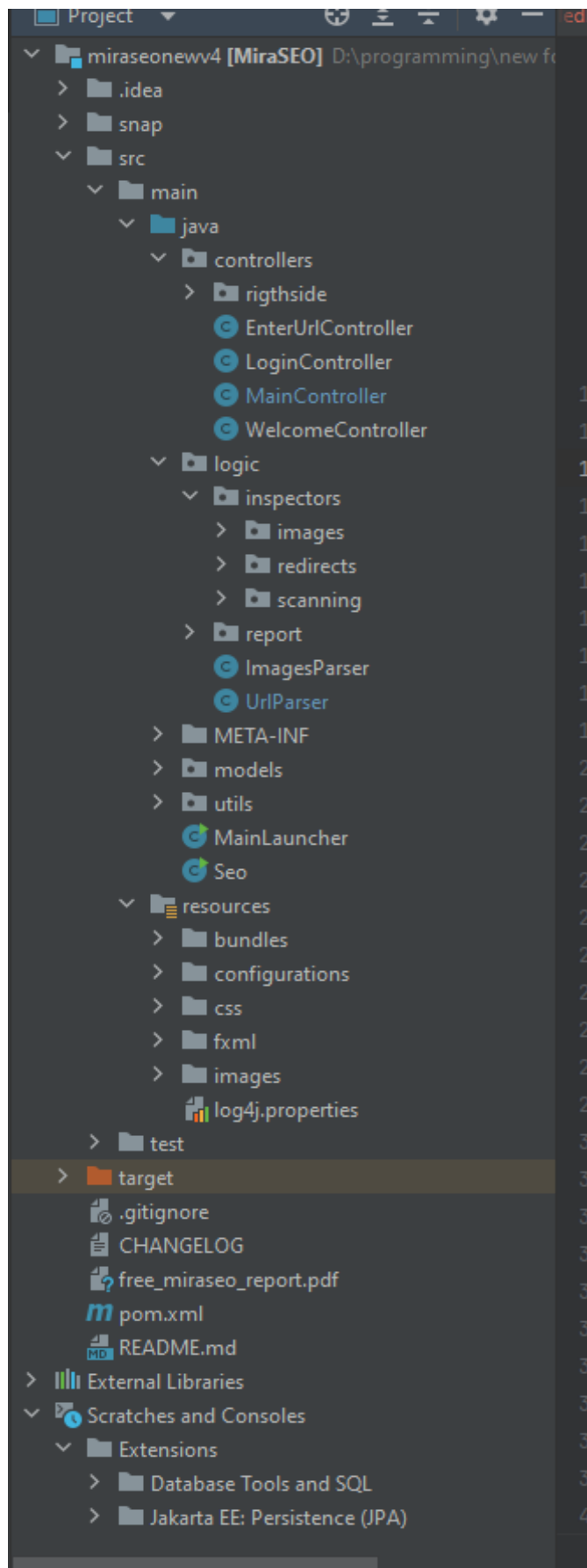


Рис. 2.14. Пакети та класи проекту

Опис файлів структури:

Основними пакетами є:

- controllers – містить усі контролери сцен застосунку, які реалізують їх функціональність та роботоспроможність;
- logic – ця папка містить класи, які реалізують парсинг сайтів відповідно до критеріїв, тобто у цьому пакеті знаходиться ядро програми;
- resources – ця папка містить такі дані як: мовні пакети та тексти, файли стилів css, файли сцен .fxml, різні зображення;
- models - ця папка містить у собі класи у яких містяться змінні, які будуть вміщувати у себе інформацію зібрану з сайтів та над якими в подальшому будуть проводитися маніпуляції;

Структура та наповнення папок:

Папка controllers:

- rightside – містить у собі контролери сцен, які будуть виводити оброблену інформацію та створювати таблиці;

Вона містить такі класи (рис. 2.15):

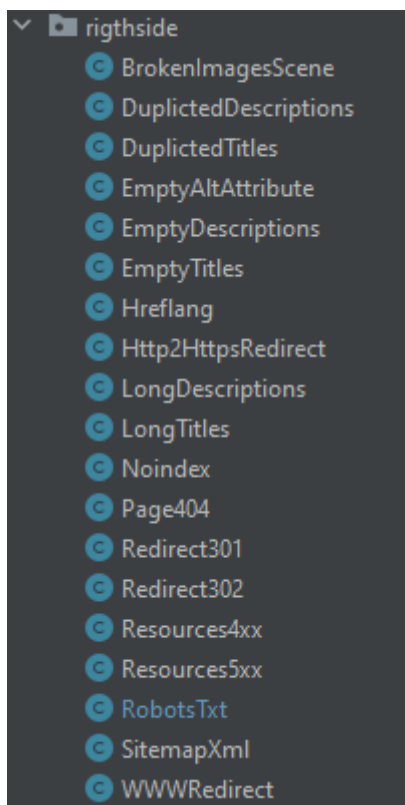


Рис. 2.15. Пакет rightside

Класи контролерів, які реалізують основні сцени додатку (рис. 2.16):



Рис. 2.16. Пакет controllers

WelcomeController реалізує функціонал стартової сцени із вибором дій (рис. 2.17): залогінитися, придбати преміум та випробувати безкоштовну версію.

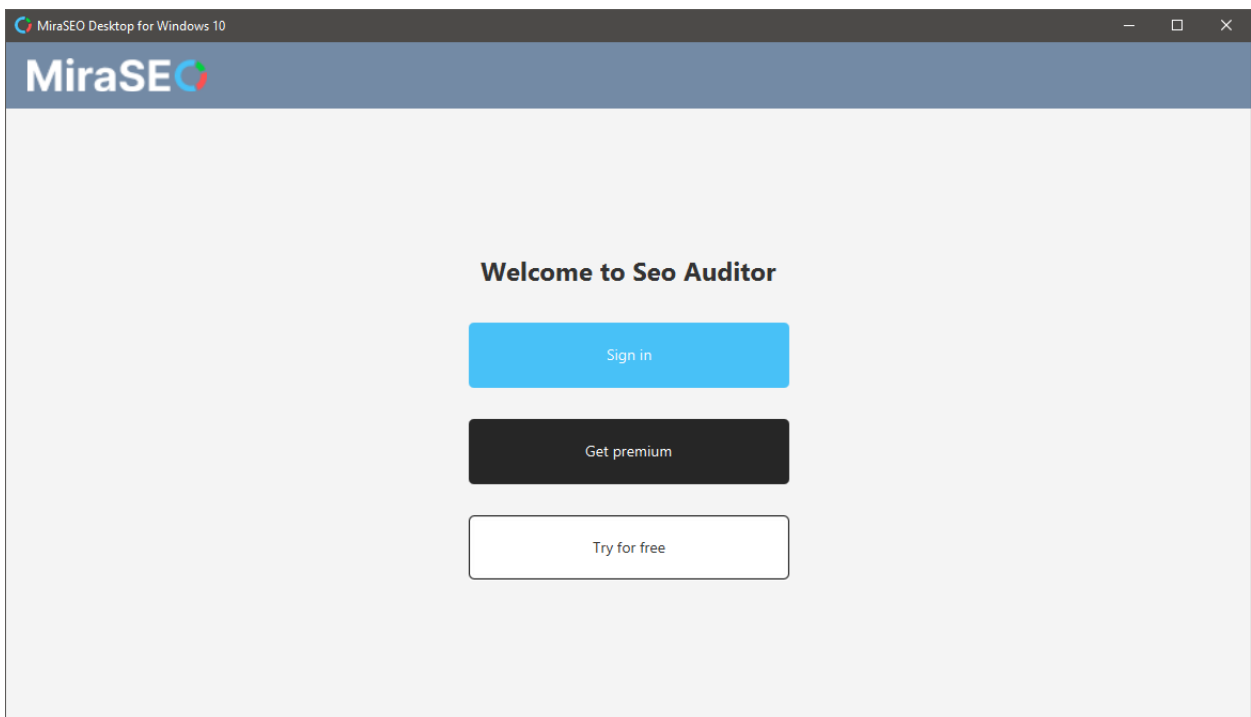


Рис. 2.17. Початкова сторінка

Клас LoginController реалізує функціонування сцени із вводом паролю та логіну (рис. 2.18).

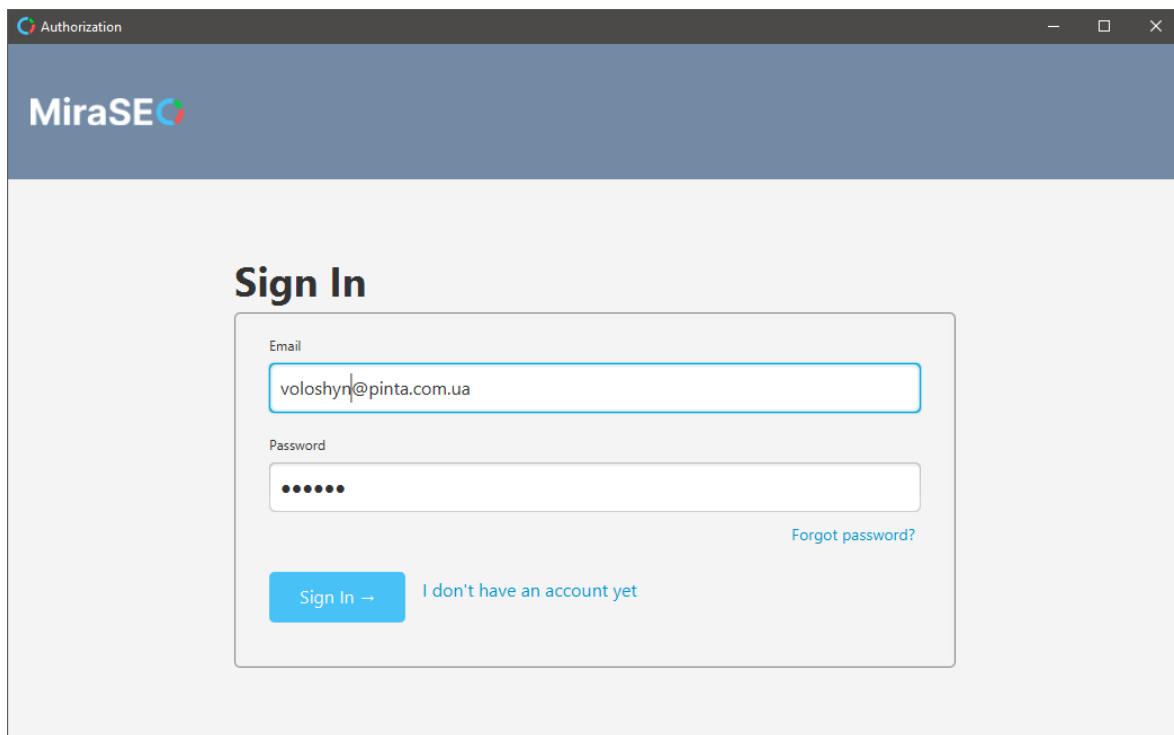


Рис. 2.18. Сторінка авторизації

EnterUrlController реалізує функціонал сцени із вводом посилання на сайт (рис. 2.19), який буде обробляти програма.

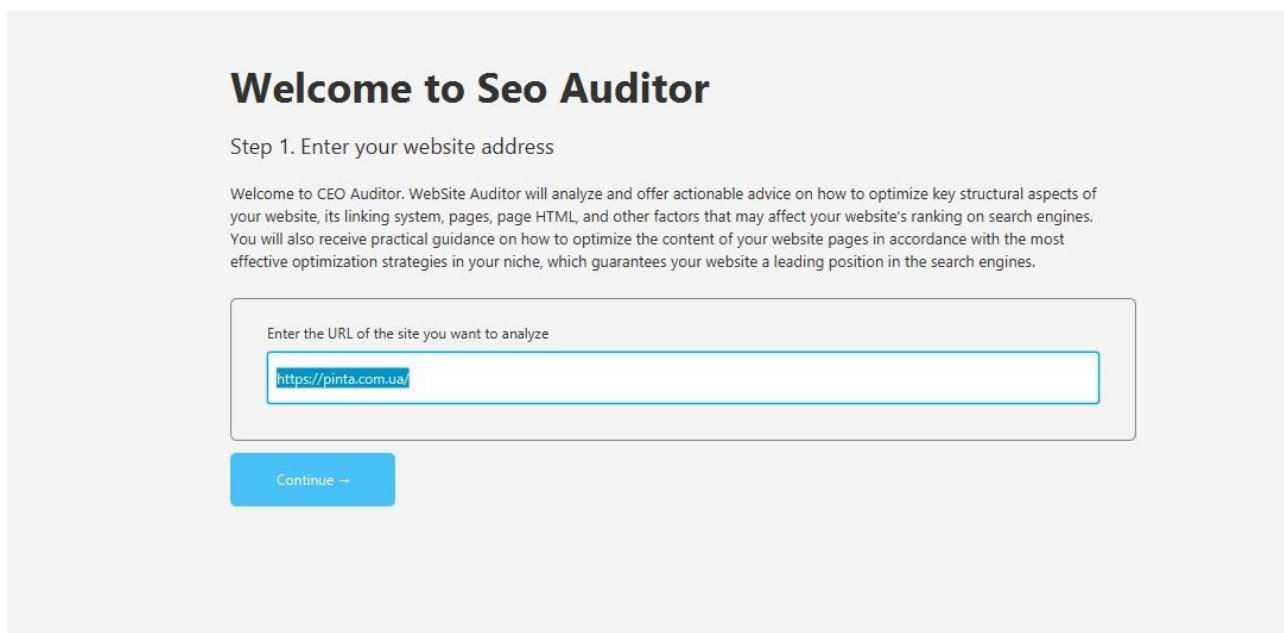


Рис. 2.19. Сторінка вводу посилання на сайт

MainController реалізує функціонал сцени із результатами, звітами, налаштуваннями (рис. 2.20).

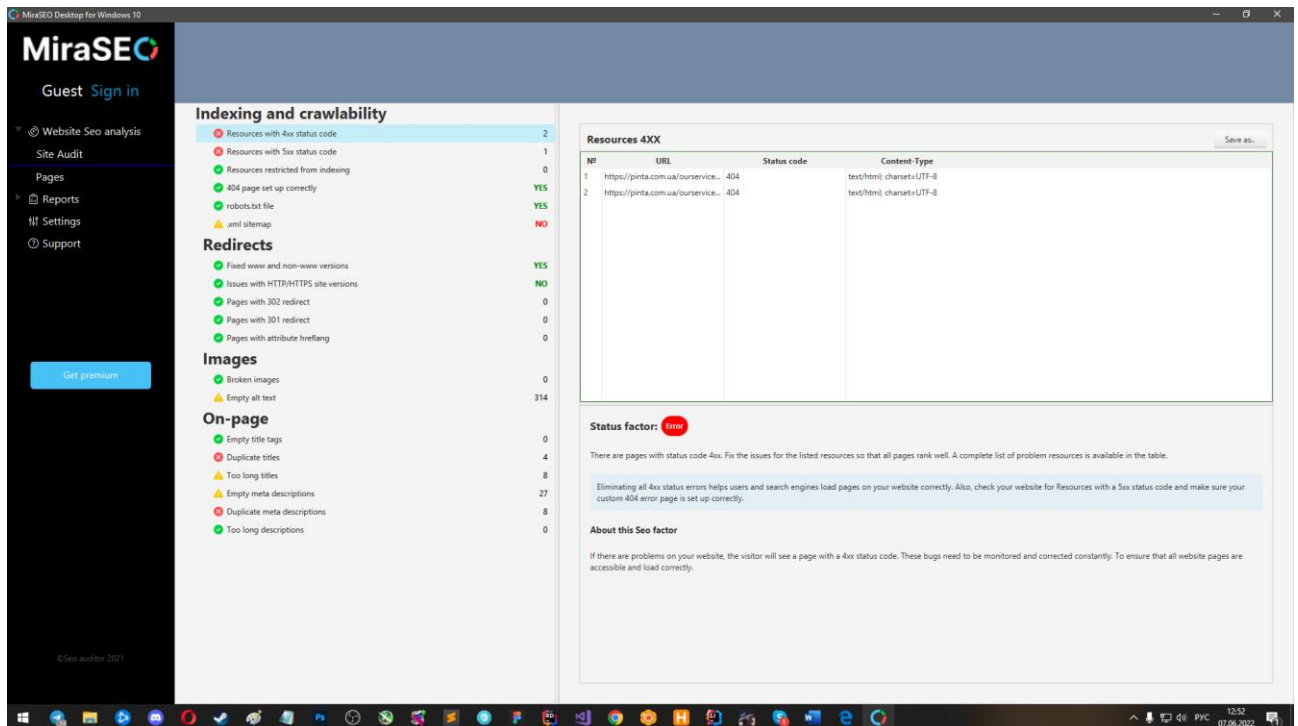


Рис. 2.20. Головна сцена

Зворотній зв'язок з клієнтами реалізований завдяки сервісу Zendesk (рис. 2.21), та йогосервісам.



Рис. 2.21. Логотип сервісу Zendesk

Zendesk - це веб-сервіс з рядом програм для підтримки клієнтів та відвідувачів.

Приклад коду запиту:

```
public void sendSupport() throws IOException,
InterruptedException {
    String name = supportName.getText();
    String email = supportEmail.getText();
    String subject = supportSubject.getText();
    String comment = supportComment.getText();

    JSONObject requester = new JSONObject();
    requester.put("name", name);
    requester.put("email", email);

    JSONObject commentBody = new JSONObject();
    commentBody.put("body", comment);

    JSONObject request = new JSONObject();
    request.put("comment", commentBody);
    request.put("subject", subject);
    request.put("requester", requester);

    JSONObject main = new JSONObject();
    main.put("request", request);

    logger.info("Result JSON: " + main);

    HttpClient httpClient =
HttpClient.newHttpClient();
```

```

HttpRequest                httpRequest                =
HttpRequest.newBuilder()

.uri (URI.create ("https://pinta.zendesk.com/api/v2/request
s.json"))
. . .

```

Пакет logic (рис. 2.22):

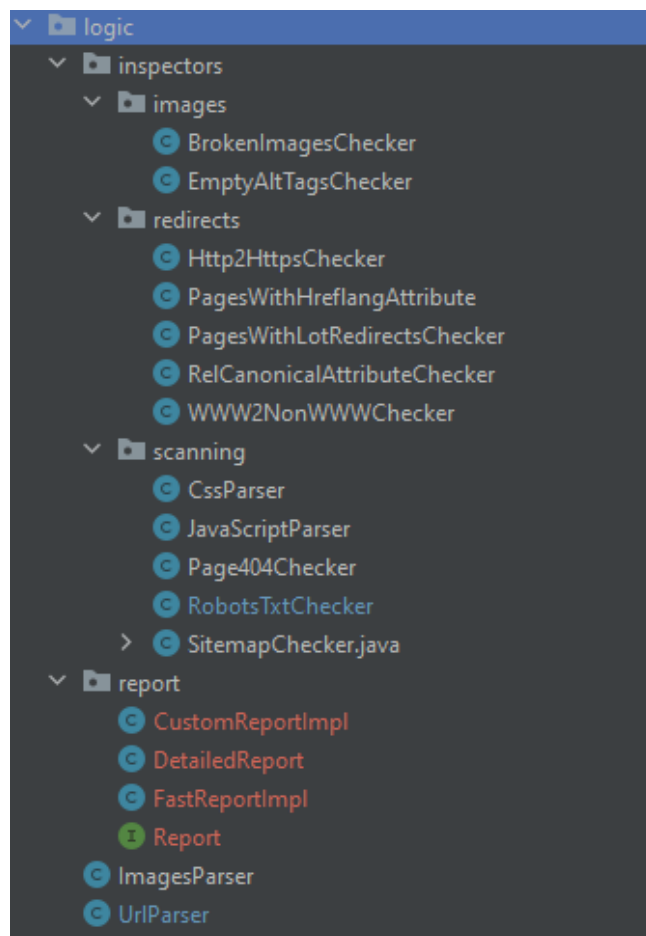


Рис. 2.22. Пакет logic

У пакетах images, redirects, scanning зібрані класи, які здійснюють пошук певних недоліків на сторінці сайту, яку на даний момент обробляє програма. Перехід від однієї сторінки до іншої виконує клас UrlParser. Суть алгоритму у переході від одної сторінки до іншої завдяки посиланням та збільшенню глибини входження. Приклад коду цього класу:

```
Elements elements = document.getElementsByTagName ("a");
```



```

        elements.stream().map(element -> element
            .absUrl("href"))
            .filter(s1 ->
urlFinderPattern.matcher(s1).matches())
            .map(this::removeHashFromURL)
            .distinct().filter(s14 ->
!AppState.skipListMap.containsKey(s14))
            .sorted()
            .peek(s13 ->
System.out.println("\tFound: " + s13))
            .forEach(s12 ->
AppState.skipListMap.putIfAbsent(s12, deep));

```

Усі необхідні нам елементи на сторінці ми знаходимо завдяки html тегам. Для цього ми використовуємо потоки вводу та виводу даних та їх методи, основним методом є метод `filter()`, який відокремлює необхідні дані зі строки. Приклад із класу який знаходить зображення:

```

public class BrokenImagesChecker implements
Callable<Boolean> {
    private final Document document;
    private final String newURL;

    public BrokenImagesChecker(Document document, String
newURL) {
        this.document = document;
        this.newURL = newURL;
    }
    @Override
    public Boolean call() throws IOException {
        Elements elements =
document.getElementsByTag("img");

```

```

        List<String> collect = elements.stream()
            .filter(element -> element.hasAttr("src")
&& element.attr("src").isEmpty())
            .map(element -> element.absUrl("src"))
            .collect(Collectors.toList());
        ImagesModel.brokenImages.addAll(collect);
        for (int i = 0; i < collect.size(); i++) {
            AppState.errors.incrementAndGet();
        }
        return true;
    }
}

```

Завантаження звітів:

Завантаження звітів може здійснюватися безпосередньо з таблиць, або у окремій вкладці з детальним звітом та швидким звітом.

Приклад коду завантаження звіту безпосередньо з таблиць:

```

private void export2XLSX() throws IOException {
    if (UserData.license == models.License.FREE) {
        Stage getPremiumWindow = new
Stage(StageStyle.DECORATED);
        Parent parent =
FXMLLoader.load(getClass().getResource("/fxml/getPremiumW
indow.fxml"), resourceBundle);
        Scene scene = new Scene(parent);
        getPremiumWindow.setScene(scene);
        getPremiumWindow.getIcons().add(new
javafx.scene.image.Image("images/icon.png"));
        getPremiumWindow.show();
    } else {

```

```

        DirectoryChooser directoryChooser = new
DirectoryChooser();
        Path pathToSave = null;

        try {
            pathToSave =
directoryChooser.showDialog(null).toPath();

        } catch (NullPointerException
NullPointerException) {

            System.out.println("Error!");
        }

        String xlsxName =
"miraseo_BrokenImages_report.xlsx";
        Path resolve = pathToSave.resolve(xlsxName);

        try {
            XSSFWorkbook workbook = new
XSSFWorkbook();
            Sheet sheet =
workbook.createSheet("list");
            FileOutputStream fileOutputStream = new
FileOutputStream(resolve.toFile());
            . . .

```

Результат (рис. 2.23):

	A	B	C
1	URL		Location
2	1	https://pinta.com.ua/marketing/ppc/test	Optional[https://pinta.com.ua/ppc]
3			

Рис. 2.23. Звіт у .xlsx форматі

Створення швидкого звіту:

```
public void exportReportToPDF() throws Exception {

    DirectoryChooser directoryChooser = new
DirectoryChooser();
    Path pathToSave = null;
    try {
        pathToSave =
directoryChooser.showDialog(null).toPath();
    } catch (NullPointerException
NullPointerException) {
        logger.error("No one output folder was
selected.");
        return;
    }

    String pdfName = "miraseo_report.pdf";
    Path resolve = pathToSave.resolve(pdfName);

    Font titleFont = new
Font(Font.FontFamily.HELVETICA, 18, Font.BOLD,
BaseColor.BLACK);
```

```

        Document document = new Document(PageSize.A4);
        PdfWriter pdfWriter =
PdfWriter.getInstance(document, new
FileOutputStream(resolveToFile()));
        pdfWriter.setPdfVersion(PdfWriter.VERSION_1_7);
        document.open();

        float[] width = {5, 1};
        PdfPTable pdfPTable = new PdfPTable(width);
. . .

```

Пакет resources:

У даному пакеті знаходяться пакети файлів, які використовуються для створення інтерфейсу (рис. 2.24).

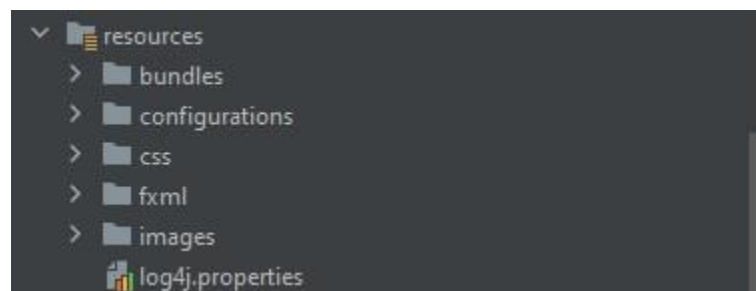


Рис. 2.24. Пакет resources

Пакет bundles містить мовні конфігурації для трьох мов української, російської та англійської, саме із цього пакету підтягуються мова при зміні її у налаштуваннях (рис. 2.25):

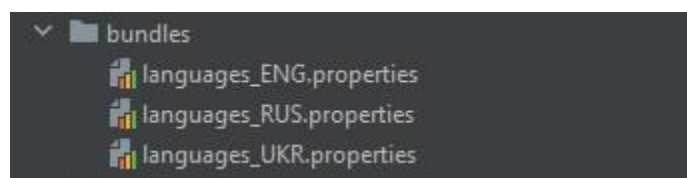


Рис. 2.25. Пакет bundles

```

miraSEO.pages_with_broken_images.ok.description=Very well! There are no broken
images on your website.

```

```
miraSEO.pages_with_broken_images.ok.description=Отлично! На Вашем сайте не  
найлены битые изображения.  
miraSEO.pages_with_broken_images.ok.description=Дуже добре! На Вашому сайті не  
знайдено биті зображення.
```

Для одного й того ж лейблу можна примінити 3 різні мови за налаштуванням.

Пакет configurations містить лише один файл у котрий записується конфігурація поточного проекту. Його вміст:

```
<?xml version="1.0" encoding="UTF-8"?>  
<launch4jConfig>  
  <dontWrapJar>>false</dontWrapJar>  
  <headerType>gui</headerType>  
  <jar>D:\Documents\IdeaProjects\seo-auditor-  
javafx\Release_files\MiraSEO-1.5.1_alpha_win.jar</jar>  
  <outfile>D:\Documents\IdeaProjects\seo-auditor-  
javafx\Release_files\MiraSEO-  
1.5.1_alpha_win.exe</outfile>  
  <errTitle></errTitle>  
  <cmdLine></cmdLine>  
  <chdir>.</chdir>  
  <priority>normal</priority>  
  <downloadUrl>http://java.com/download</downloadUrl>  
  <supportUrl></supportUrl>  
  <stayAlive>>false</stayAlive>  
  <restartOnCrash>>false</restartOnCrash>  
  <manifest></manifest>  
  <icon>D:\Documents\IdeaProjects\seo-auditor-  
javafx\src\main\resources\images\icon.ico</icon>  
  <jre>  
    <path></path>  
    <bundledJre64Bit>>false</bundledJre64Bit>
```

```

<bundledJreAsFallback>>false</bundledJreAsFallback>
  <minVersion>11.0.13</minVersion>
  <maxVersion></maxVersion>
  <jdkPreference>preferJre</jdkPreference>
  <runtimeBits>64</runtimeBits>
</jre>
<versionInfo>
  <fileVersion>1.5.1.0</fileVersion>
  <txtFileVersion>1.5.1.a</txtFileVersion>
  <fileDescription>Website structure analysis and
fast content optimization.</fileDescription>
  <copyright>Pinta Webware Copyright©</copyright>
  <productVersion>1.5.1.0</productVersion>
  <txtProductVersion>1.5.1.a</txtProductVersion>
  <productName>MiraSEO</productName>
  <companyName>Pinta Webware</companyName>
  <internalName>MiraSEO</internalName>
  <originalFilename>MiraSEO-
1.5.1_alpha_win.exe</originalFilename>
  <trademarks></trademarks>
  <language>ENGLISH_US</language>
</versionInfo>
</launch4jConfig>

```

Наступний пакет CSS містить стилі для різних елементів інтерфейсу програми (рис. 2.26):

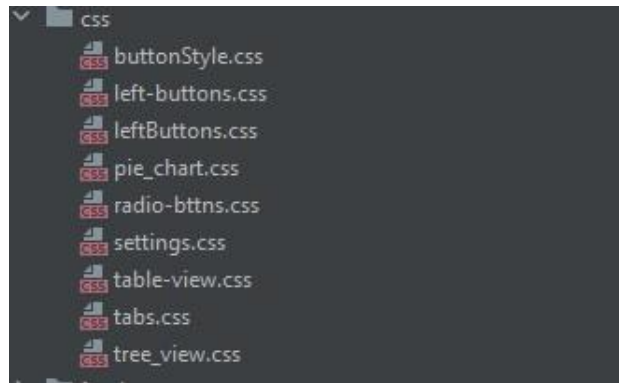


Рис. 2.26. Пакет css

Пакет images містить загалом маленькі .png зображення, для відображення статусів, іконок та флажків для мов (рис. 2.27).

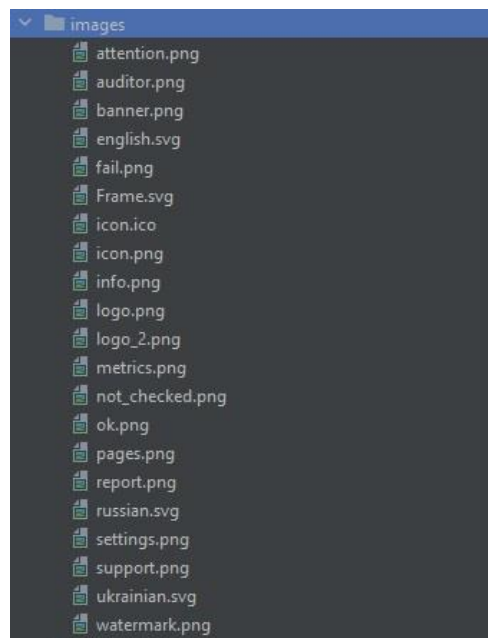


Рис. 2.27. Пакет images

Мабудь, найважливішим пакетом ресурсів, є пакет з fxml файлами, які реалізують сцени інтерфейсу користувача (рис. 2.28).

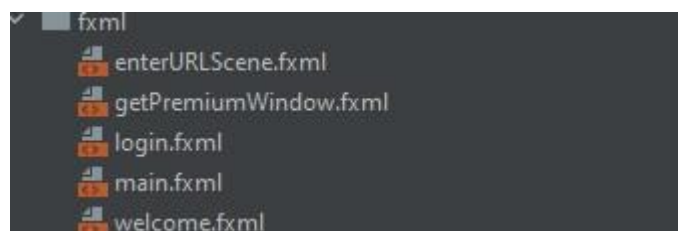


Рис. 2.28. Пакет fxml

Приклад fxml файлу enterURLScene.fxml:

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.Group?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Hyperlink?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.shape.SVGPath?>
<?import javafx.scene.text.Font?>

<AnchorPane fx:id="mainAnchorPane"
stylesheets="/css/left-buttons.css"
xmlns="http://javafx.com/javafx/17"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.EnterUrlController">
    <children>
```

```

    <GridPane prefHeight="580.0" prefWidth="1043.0"
AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
    <columnConstraints>
        <ColumnConstraints maxWidth="250.0"
minWidth="-Infinity" />
        <ColumnConstraints hgrow="SOMETIMES"
maxWidth="1.7976931348623157E308" minWidth="-Infinity" />
    </columnConstraints>
    <rowConstraints>
        <RowConstraints maxHeight="120.0"
minHeight="-Infinity" vgrow="SOMETIMES" />
        <RowConstraints
maxHeight="1.7976931348623157E308" minHeight="-Infinity"
vgrow="SOMETIMES" />
    </rowConstraints>
    <children>
        <AnchorPane prefHeight="150.0" style="-
fx-background-color: #738AA5;" GridPane.columnIndex="1">
            <children>
                <ImageView fitHeight="26.0"
fitWidth="136.0" layoutX="866.0" layoutY="40.0"
pickOnBounds="true" preserveRatio="true"
AnchorPane.leftAnchor="25.0" AnchorPane.topAnchor="50.0">
                    <image>
                        <Image
url="@../images/logo_2.png" />
                    </image>
                </ImageView>
            </children>
        </AnchorPane>
    </children>
</GridPane>
. . .

```

Для підключення усіх бібліотек ми використовуємо pom.xml файл, який автоматично створює компілятор проектів Maven. До цього файлу ми додаємо залежності Maven які реалізують підключення необхідних нам бібліотек.

Файл pom.xml:

```
<project>
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.pinta</groupId>
  <artifactId>MiraSEO</artifactId>
  <version>1.5.1_alpha</version>
  <name>MiraSEO</name>
  <packaging>jar</packaging>
  <organization>
    <name>Pinta Webware</name>
    <url>https://pinta.com.ua</url>
  </organization>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <junit.version>5.8.1</junit.version>

  <maven.compiler.source>11</maven.compiler.source>

  <maven.compiler.target>11</maven.compiler.target>

  <maven.compiler.release>11</maven.compiler.release>

  <project.outputDirectory>Release_files</project.outputDirectory>
```

```
<os.name>win</os.name>
<javafx.version>18</javafx.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.controlsfx</groupId>
    <artifactId>controlsfx</artifactId>
    <version>11.1.1</version>
  </dependency>

  <dependency>
    <groupId>org.openjfx</groupId>
    <artifactId>javafx-swing</artifactId>
    <classifier>${os.name}</classifier>
    <version>${javafx.version}</version>
  </dependency>

  <dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>5.2.2</version>
  </dependency>

  <dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>5.2.2</version>
  </dependency>
```

```
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-csv</artifactId>
  <version>1.9.0</version>
</dependency>
```

```
<dependency>
  <groupId>com.opencsv</groupId>
  <artifactId>opencsv</artifactId>
  <version>5.6</version>
</dependency>
```

```
<dependency>
  <groupId>org.openjfx</groupId>
  <artifactId>javafx-web</artifactId>
  <classifier>${os.name}</classifier>
  <version>${javafx.version}</version>
</dependency>
```

```
<dependency>
  <groupId>org.openjfx</groupId>
  <artifactId>javafx-fxml</artifactId>
  <classifier>${os.name}</classifier>
  <version>${javafx.version}</version>
</dependency>
```

```
<dependency>
  <groupId>org.json</groupId>
  <artifactId>json</artifactId>
  <version>20220320</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.jsoup</groupId>
```

```
  <artifactId>jsoup</artifactId>
```

```
  <version>1.14.3</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>log4j</groupId>
```

```
  <artifactId>log4j</artifactId>
```

```
  <version>1.2.17</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>com.github.oshi</groupId>
```

```
  <artifactId>oshi-core</artifactId>
```

```
  <version>6.1.5</version>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.apache.httpcomponents</groupId>
```

```
  <artifactId>httpmime</artifactId>
```

```
  <version>4.5.13</version>
```

```
</dependency>
```

```
. . .
```

Остання частина `<build>...</build>` необхідна для компіляції проекту у `.jar` файл з подальшим перетворенням у `.exe`. У цій частині вказується директорія, куди буде схований білд, версія білду, операційна система, назва білду.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

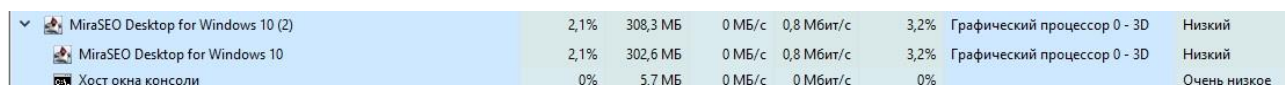
У якості вхідних даних додаток приймає дані користувача, які він надає під час реєстрації, посилання на сайт, який буде аналізуватися, дані з сайту, токен авторизації користувача. У якості вихідних даних користувач отримує звіти у різних форматах, результат роботи програми також виводиться до таблиць, блоків та діаграм.

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

Додаток завдяки адаптивності може бути відкритий та запущений на усіх основних операційних системах. Також додаток адаптований під використання на пристроях різної конфігурації.

Додаток не навантажує систему та є надійним й стабільним (рис. 2.30).



MiraSEO Desktop for Windows 10 (2)	2,1%	308,3 МБ	0 МБ/с	0,8 Мбит/с	3,2%	Графический процессор 0 - 3D	Низкий
MiraSEO Desktop for Windows 10	2,1%	302,6 МБ	0 МБ/с	0,8 Мбит/с	3,2%	Графический процессор 0 - 3D	Низкий
Хост окна консоли	0%	5,7 МБ	0 МБ/с	0 Мбит/с	0%		Очень низкое

Рис. 2.30. Пікове навантаження системи додатком

Як ми бачимо, проблем із навантаженістю нема.

2.6.2. Використані програмні засоби

Для використання додатку необхідно мати встановлену Java на пристрої (не нижче 11.0.13 версії). Користувач може мати будь-яку операційну систему на своєму пристрої (Windows, Linux), найважливішою умовою для роботоспроміжності додатку є наявність стабільного інтернет-з'єднання.

2.6.3. Виклик та завантаження програми

Для виклику додатку на пристрої, необхідно у IDE реалізувати запуск стартового методу та дочекатися збірки проекту, виконати запуск .jar файлу через системну консоль, або конвертувати .jar файл у .exe файл за допомогою програми **launch4j**.

На даний момент додаток знаходиться у стадії розробки, тому він доступний лише розробникам(мені).

2.6.4 Опис інтерфейсу додатку

Після запуску додатку користувач потрапляє на початкову сторінку додатку, що зображена на (рис. 2.31):

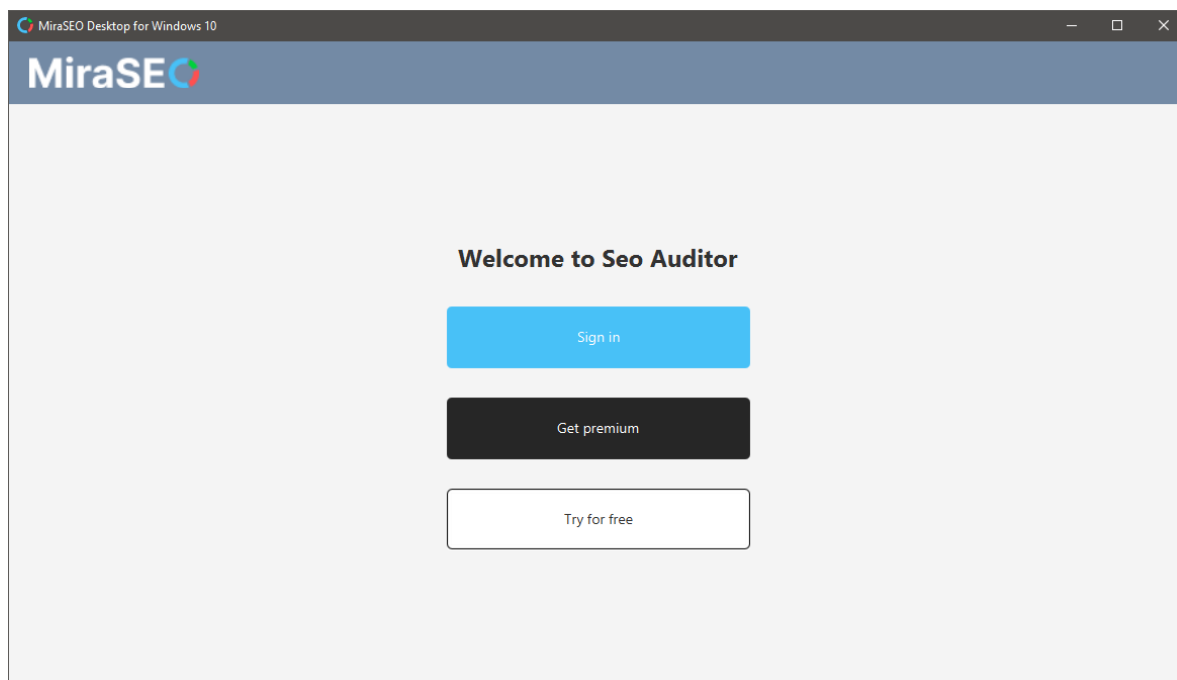


Рис. 2.31. Початкова сторінка

Користувач може обрати один з трьох варіантів: увійти зі своєю обліковим записом, купити преміум версію додатку або спробувати безкоштовно але з деякими обмеженнями, які стосуються звітів.

Після натискання на кнопку Sign in ми потрапляємо на форму вводу паролю та логіну (рис. 2.32):

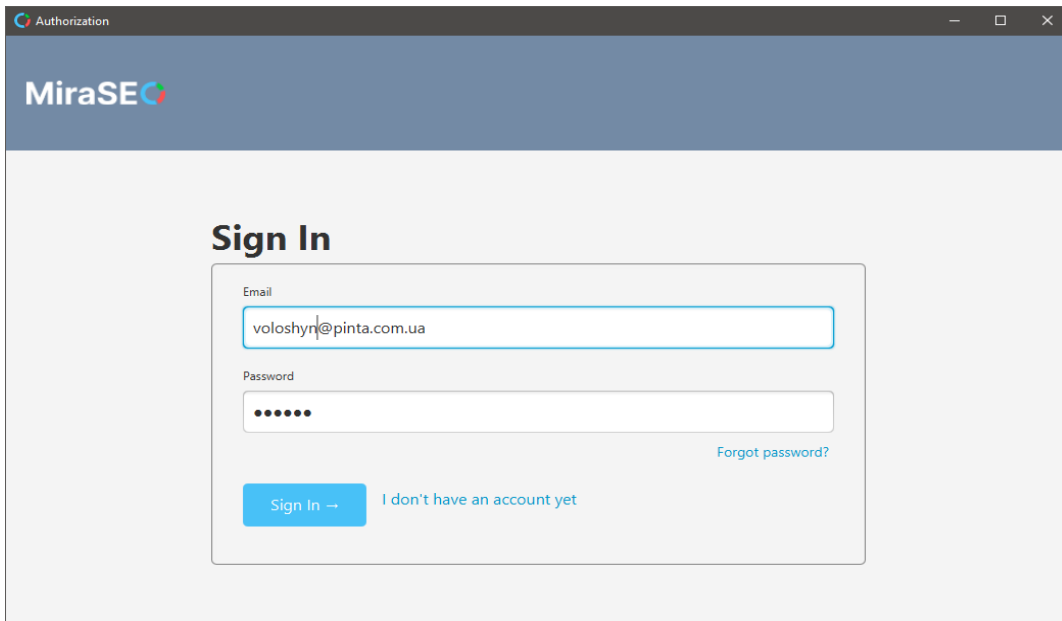


Рис. 2.32. Сторінка авторизації

Якщо ви не маєте свого акаунту, можна зареєструватися натиснувши на «I don't have an account yet» (рис. 2.33):

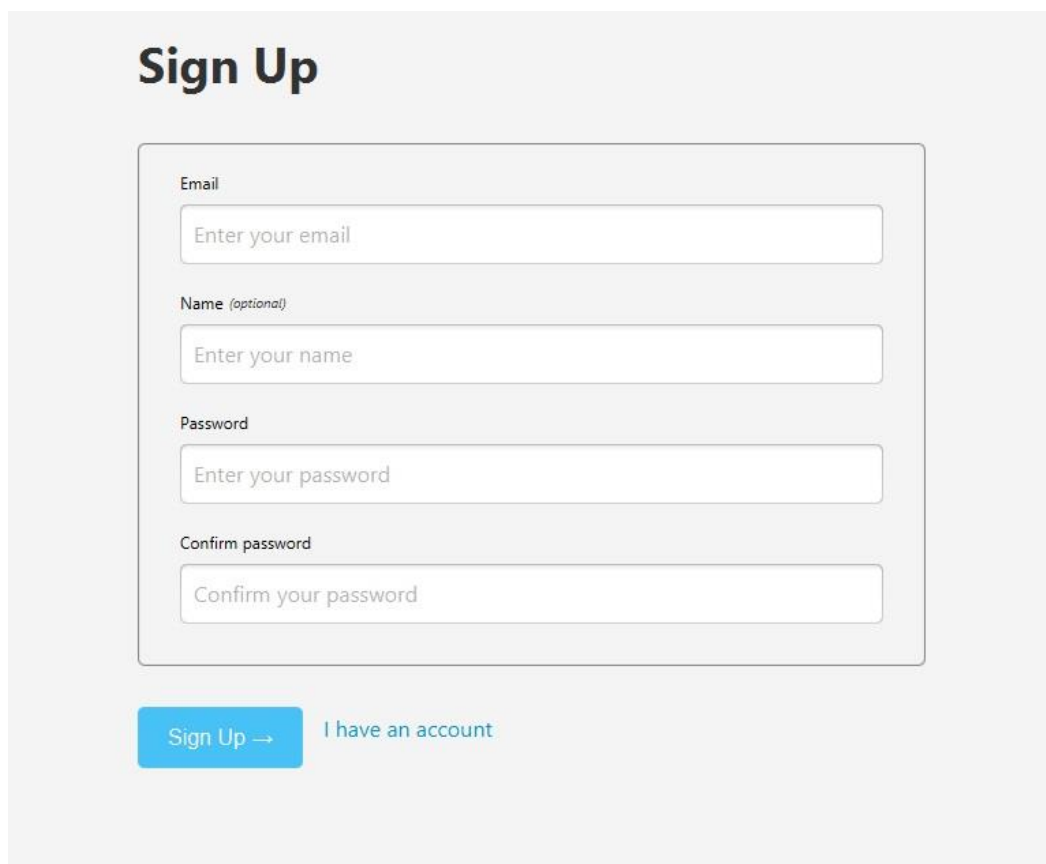
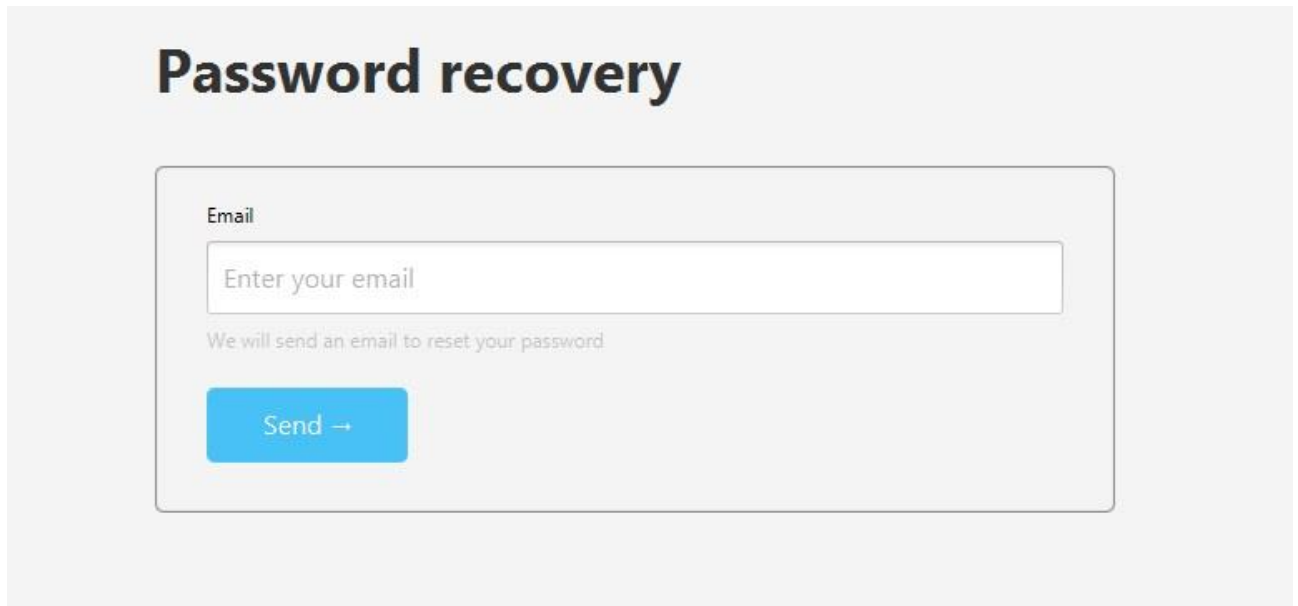


Рис. 2.33. Сцена реєстрації

Вводите усі необхідні дані, відправляються на сервер, який хранить і обробляє інформацію про користувачів, та створюєте нового користувача додатку, після створення дані також відправляються до юази даних, яка містить інформацію про логін, пароль та запити користувача.

Якщо ви забули праоль (рис. 2.34), то натисніть на «Forgot password?»



Password recovery

Email

Enter your email

We will send an email to reset your password

Send →

Рис. 2.34. Сцена сбросу паролю

Вводите свою пошту, та на неї надходить токен для зміни паролю.

Повернемося до стартової сторінки. Після натискання на кнопку “Get Premium”, відкривається сцена з декількома версіями підписки (рис. 2.35):

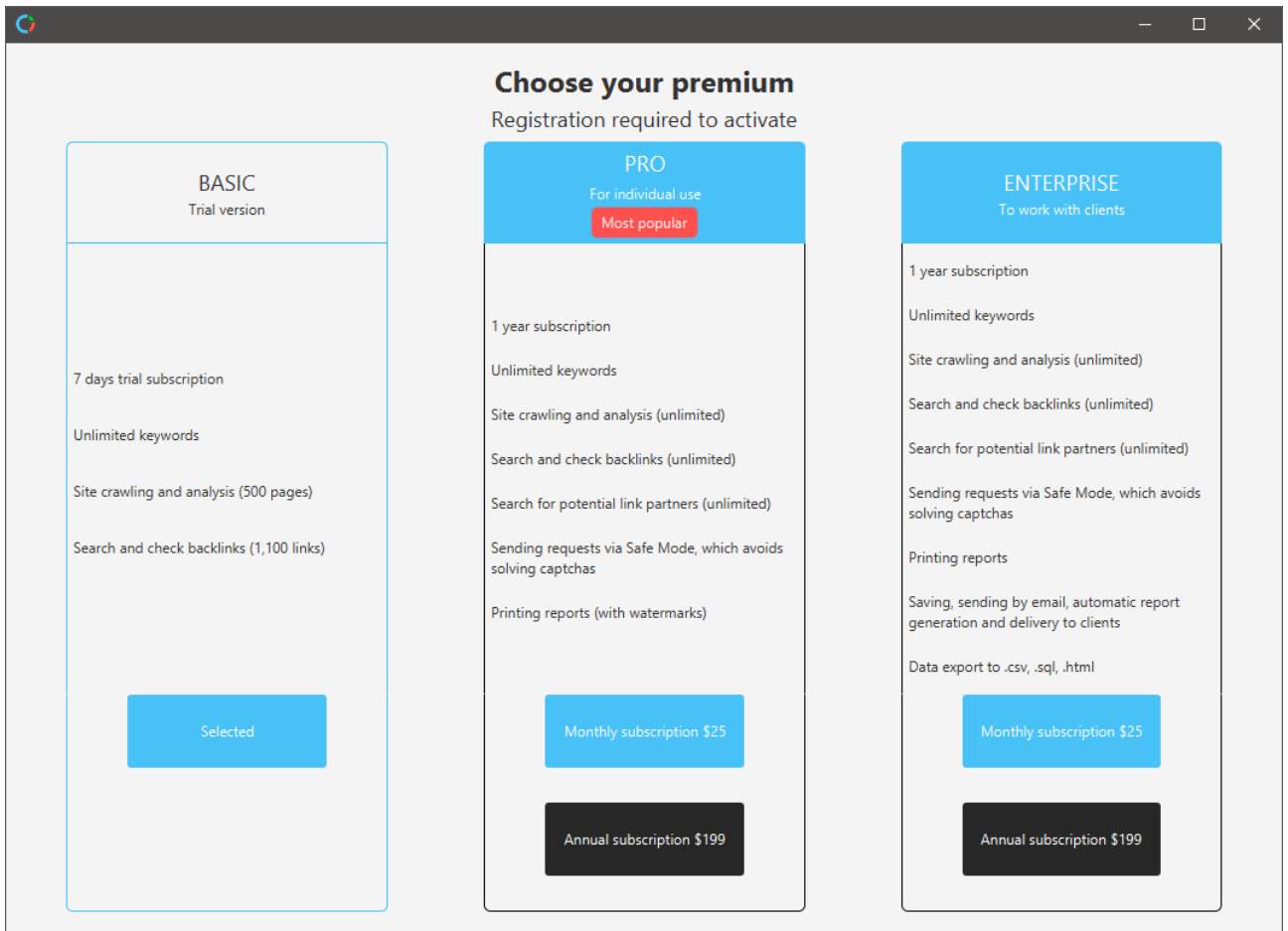


Рис. 2.35. Сцена придбання підписки

Після натискання на певну кнопку, відкриється сайт додатку (рис. 2.36):

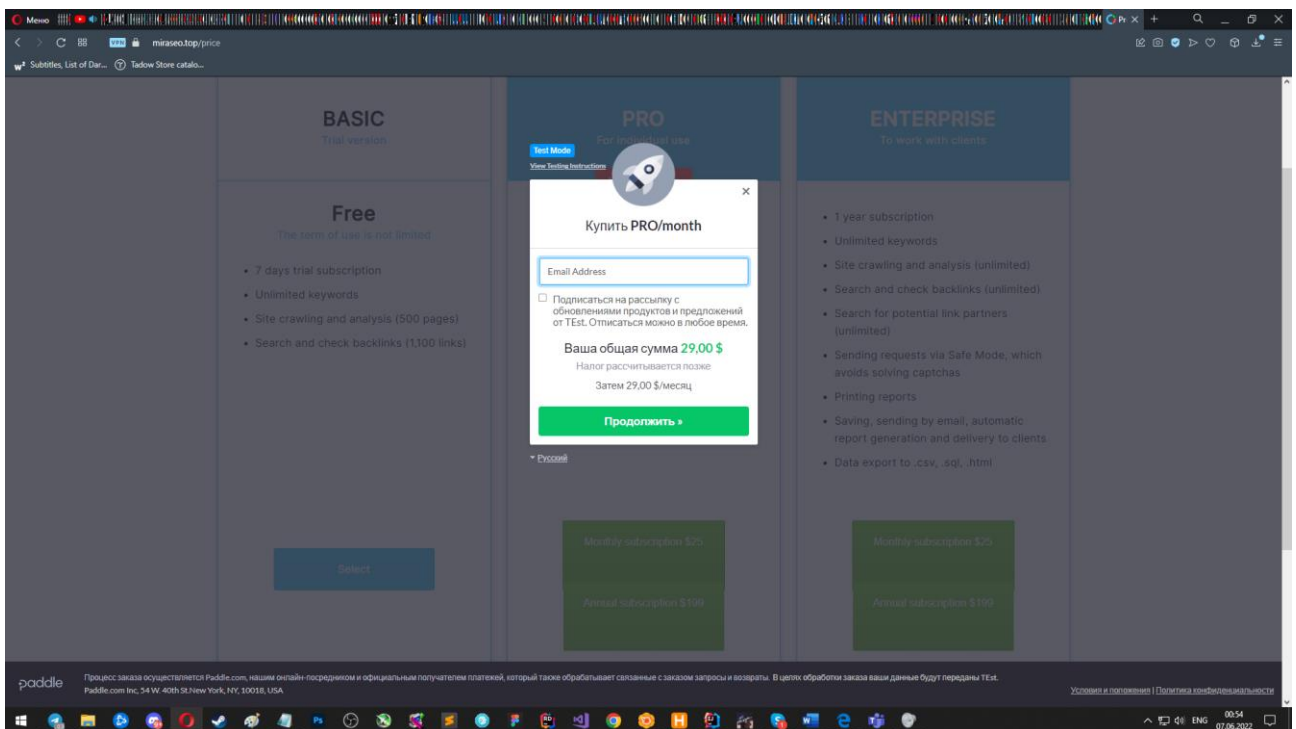


Рис. 2.36. Сайт додатку

Та натиснувши на останню кнопку «Try for free», ми одразу потрапляємо на сцену, де ми вводимо посилання на сайт (рис. 2.37), котрий необхідно проаналізувати.

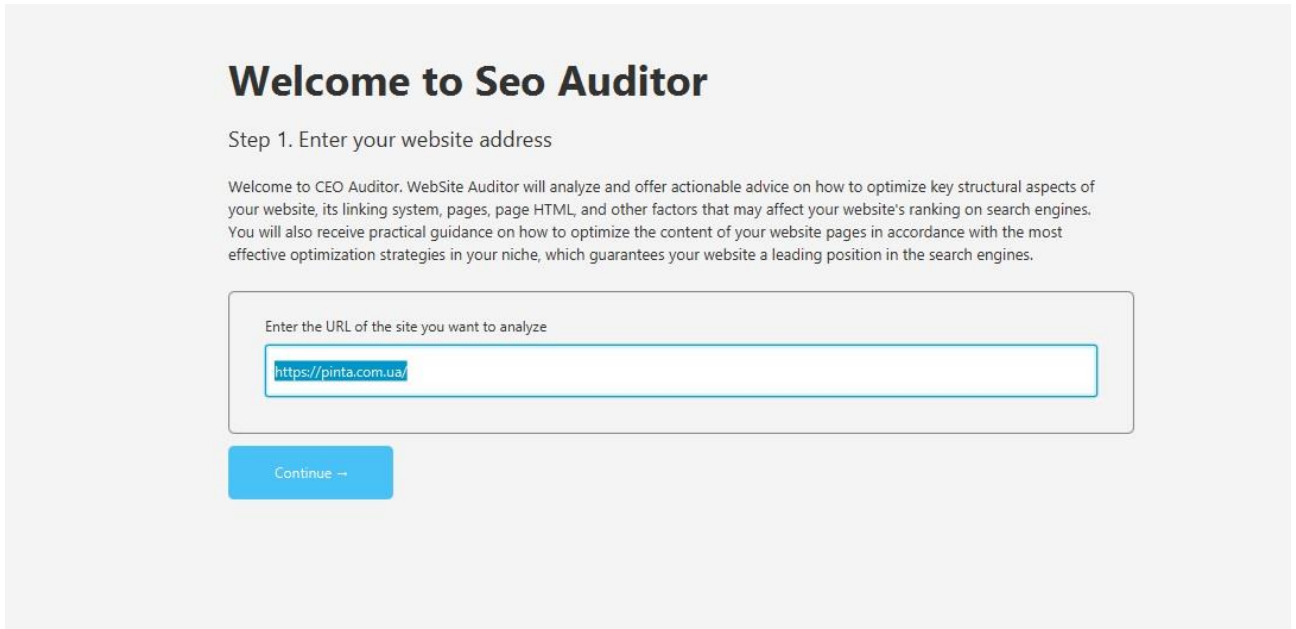


Рис. 2.37. Ввід посилання на сайт

Після того, як ми ввели посилання, та натиснули кнопку «Continue» - починається парсинг сайту, та нас перекидує на іншу сцену:

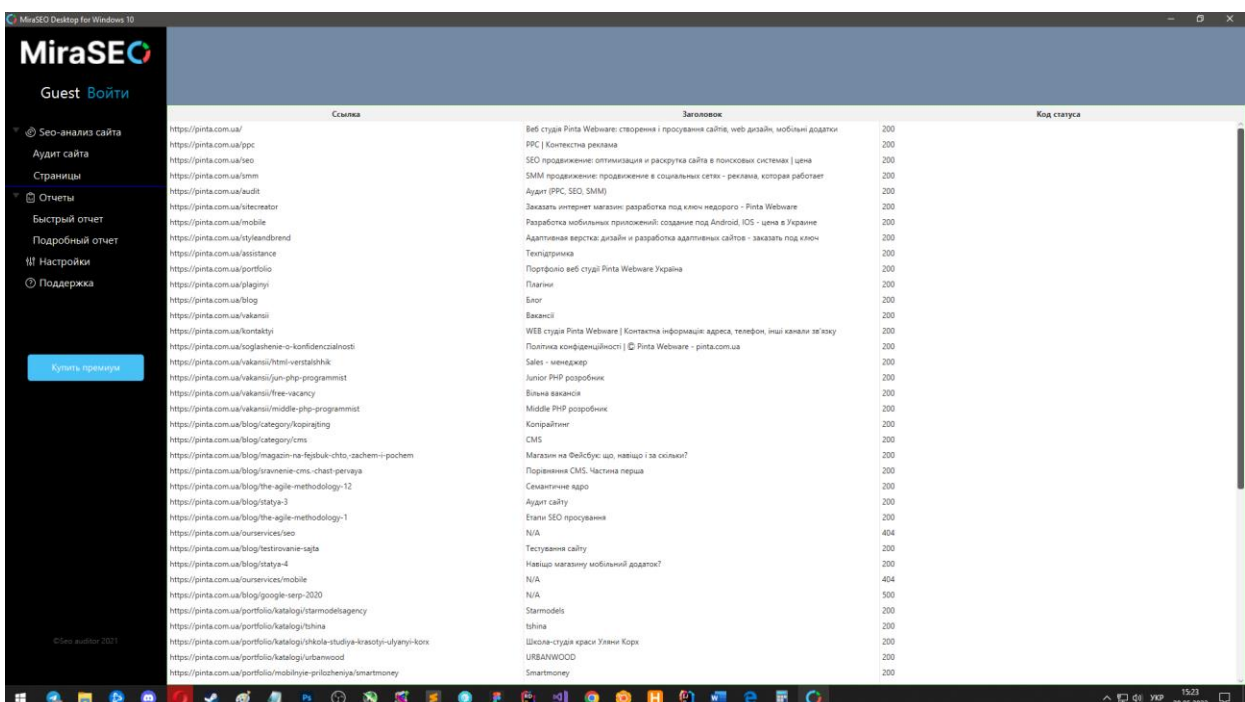


Рис. 2.38. Головна сцена

Код написаний за допомогою бібліотеки log4j, виводить до консолі усю необхідну інформацію про кожну оброблену сторінку (рис. 2.39):

```
2022-06-09 17:14:53 INFO 65 UrlParser -> "https://pinta.com.ua/vakansii/jun-php-programmist"
2022-06-09 17:14:53 INFO 99 UrlParser -> Status code: 200
2022-06-09 17:14:53 INFO 100 UrlParser -> Content type: text/html; charset=UTF-8
2022-06-09 17:14:53 INFO 116 UrlParser -> Title: "Junior PHP розробник"
2022-06-09 17:14:53 INFO 117 UrlParser -> Description: "Детальніше"
```

Рис. 2.39. Інформація у консолі

Ми одразу бачимо список посилань, які поповнюються, це сторінки які вже оброблені програмою.

Після завершення обробки сайту, усі результати структуровано виводяться до таблиць (рис. 2.40), відповідно кожному критерію пошуку:

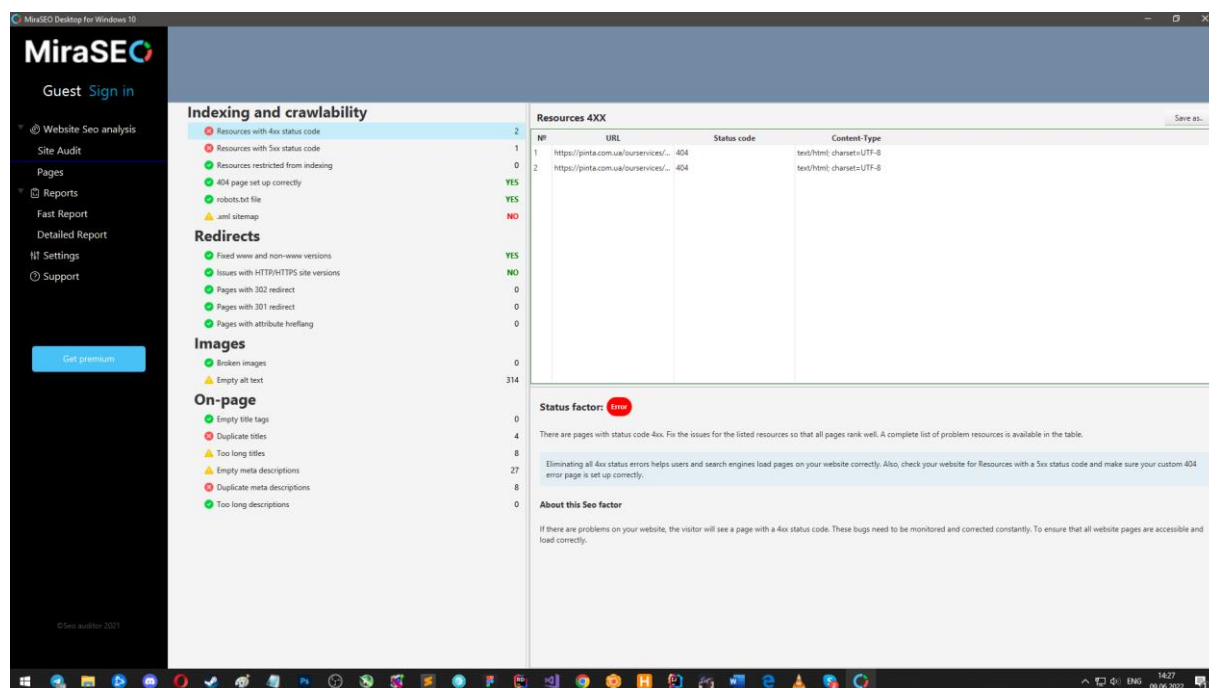


Рис.2.40. Вкладка з критеріями

Розберемося за що кожний критерій відповідає:

1) Ресурси з кодом стану 4XX (рис. 2.41):

- 400 (Bad Request):

У повідомленні запити є синтаксична помилка;

- 401 (Unauthorised):

Для надсилання запити потрібна інформація про аутентифікацію Http;

- 403(Forbidden):

У доступі до запитаного ресурсу відмовлено;

- 404 (Not Found):

Ресурс не можна знайти на сервері.

Resources 4XX			
№	URL	Status code	Content-Type
1	https://pinta.com.ua/ourservices/seo	404	text/html; charset=UTF-8
2	https://pinta.com.ua/ourservices/mobile	404	text/html; charset=UTF-8

Рис. 2.41. Таблиця результатів 4XX

- 2) Ресурси з кодом стану 5XX (рис. 2.42):

- 500 (Internal Server Error):

Сервер надсилає виняток під час виконання запити;

- 503(Server Unavailable):

Сервер тимчасово не може опрацювати запит.

Resources 5XX			
№	URL	Status code	Content-Type
1	https://pinta.com.ua/blog/google-serp-2020	500	text/html; charset=UTF-8

Рис. 2.42. Таблиця результатів 5XX

3) Сторінки, закриті від індексування – це сорінки до яких використаний тег <noindex></noindex>, він вказує на те, що сторінки до якої він приміняється не враховуються пошуковою оптимізацією.

4) 404 сторінка налаштована коректно – це означає що вона стилізована та хоча б містить посилання на початкову сторінку (рис. 2.43).

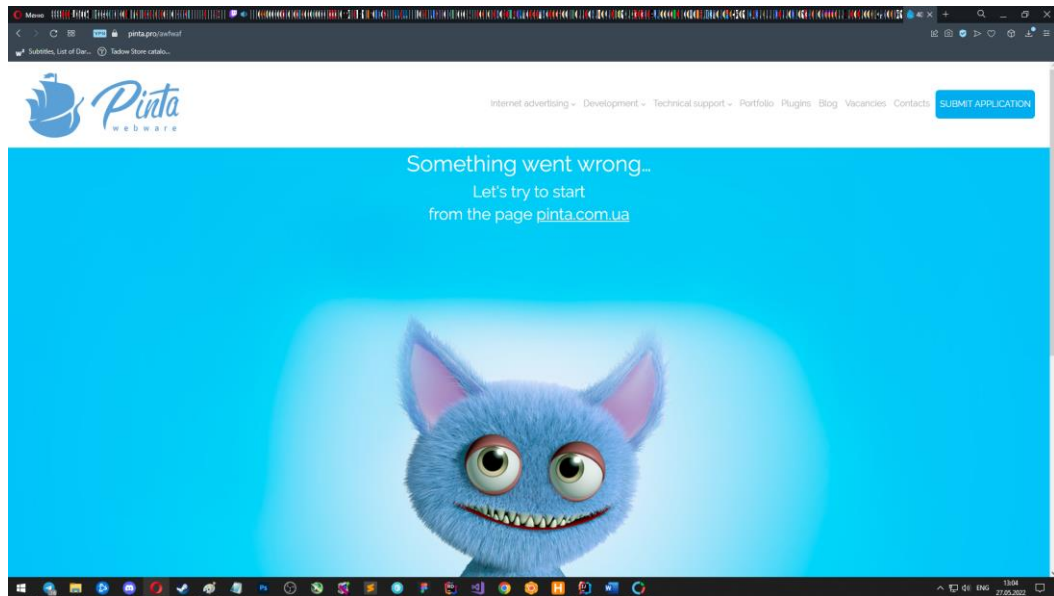


Рис. 2.43. Коректно налаштована сторінка 404

5) Файл robots.txt – параметр перевіряє наявність цього файлу. Файл robots.txt використовується роботами пошукових систем для отримання параметрів що до сайту.

6) .xml карта сайту – або (sitemap), містить список URL-адрес сторінок вашого сайту, як довідник, а також деяку іншу інформацію для пошукових систем.

7) Виправлена версія з www та без www – перевіряє чи вказується пріоритетний варіант доменного імені сайту.

8) Проблеми з версіями HTTP/HTTPS - HTTP/HTTPS версії вашого сайту можуть індексуватися пошуковими системами я сторінки що дублюються. Для того що би виправити це, встановлюється пріоритетна версія (HTTP/HTTPS)/

9) Сторінки із перенаправленням 302(Redirect 302) – цей параметр виводить усі сторінки які мають 302 перенаправлення. 302 redirect – це тимчасовий редирект. Він не передає посилальну вагу і повідомляє пошукові системи, що сторінка тимчасово переміщена на іншу URL адресу.

10) Сторінки із перенаправленням 301(Redirect 301) – цей параметр виводить усі сторінки які мають 302 перенаправлення. Переадресація (редирект) 301 вказує на те, що сторінку остаточно переміщено з одного місця до іншого.

Частина 301 відноситься до коду стану HTTP перенаправленої сторінки. Простіше кажучи, 301 редирект повідомляє браузеру: Ця сторінка була переміщена назавжди (рис. 2.44).

Redirect 301			Сохранить как..
№	URL	Redirect to	
1	https://pinta.com.ua/marketing/ppc/test	https://pinta.com.ua/ppc	

Рис. 2.44. Таблиця результатів 301 перенаправлення

11) Биті зображення – битим зображенням вважається будь-яке зображення, яке неможливо завантажити. Помилки, які заважають відобразити зображення, можуть бути різні – помилки з кодом 4XX, 5XX, некоректний URL у тегу , тощо (рис. 2.45).

Broken Images				Сохранить как..
Page	Image URL	HTTP Status Code	Alt text	
https://pinta.com.ua/vakansii/jun-php-programmist	pinta.com.ua/	200		
https://pinta.com.ua/vakansii/free-vacancy	pinta.com.ua/	200		
https://pinta.com.ua/vakansii/middle-php-programmist	pinta.com.ua/	200		

Рис. 2.45. Таблиця результатів битих зображень

12) Порожній alt текст – цей критерій відображає зображення з порожнім alt текстом. alt текст – короткий опис зображення з використанням цільових слів для пошукової оптимізації (рис. 2.46).

Empty Alt Attribute			Сохранить как..
№	Title	Image URL	
1	Веб студія Pinta Webware: створення і просування сайтів, we...	https://pinta.com.ua/assets/img/home/fon/services-1.png	
2	Веб студія Pinta Webware: створення і просування сайтів, we...	https://pinta.com.ua/assets/img/home/fon/services-2.png	
3	Веб студія Pinta Webware: створення і просування сайтів, we...	https://pinta.com.ua/assets/img/home/fon/services-3.png	
4	Веб студія Pinta Webware: створення і просування сайтів, we...	https://pinta.com.ua/assets/img/home/fon/services-4.png	
5	Веб студія Pinta Webware: створення і просування сайтів, we...	https://pinta.com.ua/assets/img/home/fon/services-5.png	
6	Веб студія Pinta Webware: створення і просування сайтів, we...	https://pinta.com.ua/assets/img/home/fon/services-6.png	
7	Веб студія Pinta Webware: створення і просування сайтів, we...	https://pinta.com.ua/assets/img/home/fon/services-7.png	

Рис. 2.46. Таблиця результатів порожнього alt атрибуту

13) Порожні теги заголовків – критерій відображає порожні теги заголовків, якщо вони є. Використання тегу заголовка допомагає пошуковим системам та користувачам отримати представлення про наповнення сторінки.

14) Заголовки що повторюються – відображає заголовки які повторюються. Заголовки що повторюються можуть розцінюватися пошуковими системами як інформація що дублюється на сторінках та є негативним фактором при ранжуванні вашого сайту.

15) Надто довгі заголовки – відображає надто довгі заголовки. У результатах видачі будуть відображатися не більше 70 символів, тому не варто додавати надто довгий заголовок (рис. 2.47).

Long Titles			
№	URL	Title	Title length
1	https://pinta.com.ua/	Веб студія Pinta Webware: створення і просування сайтів, web дизайн, мобільні додатки	85
2	https://pinta.com.ua/seo	SEO продвижение: оптимизация и раскрутка сайта в поисковых системах цена	74
3	https://pinta.com.ua/smm	SMM продвижение: продвижение в социальных сетях - реклама, которая работает	75
4	https://pinta.com.ua/sitecreator	Заказать интернет магазин: разработка под ключ недорого - Pinta Webware	71
5	https://pinta.com.ua/mobile	Разработка мобильных приложений: создание под Android, IOS - цена в Украине	75
6	https://pinta.com.ua/styleandbrand	Адаптивная верстка: дизайн и разработка адаптивных сайтов - заказать под ключ	77
7	https://pinta.com.ua/kontaktyi	WEB студія Pinta Webware Контактна інформація: адреса, телефон, інші канали зв'язку	85

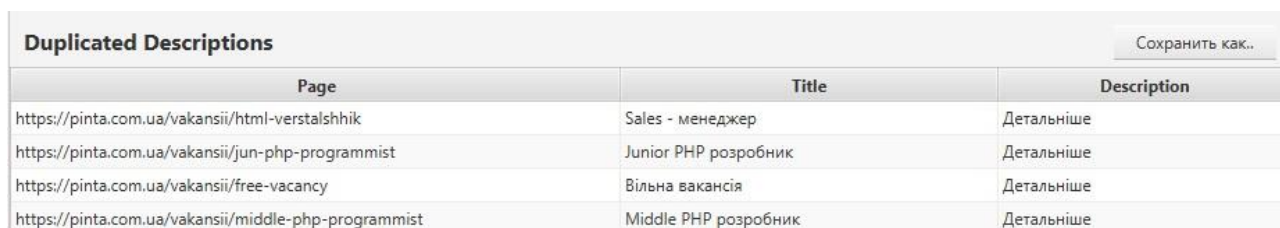
Рис. 2.47. Таблиця результатів з довгими заголовками

16) Порожні мета-описи – відображає усі порожні мета-описи що знайшла програма. Мета-описи повідомляють пошуковим системам короткі описи наповнення сторінки. Також мета-описи можуть бути використані для формування сніпета у результатах видачі. Якщо мета-опис сторінки порожнє, пошукова система сгенерує його сама. Таким чином користувачу буде відображатися у пошуковій видачі поламаний або незрозумілий опис, котрий не дозволить відкрити ваш сайт (рис. 2.48).

Empty Descriptions		
№	URL	Title
1	https://pinta.com.ua/assistance	Техпідтримка
2	https://pinta.com.ua/plaginyi	Плагіни
3	https://pinta.com.ua/vakansii	Вакансії
4	https://pinta.com.ua/blog/magazin-na-fejsbuk-chto,-zachem-i-pochem	Магазин на Фейсбук: що, навіщо і за скільки?
5	https://pinta.com.ua/blog/sravnienie-cms.-chast-pervaya	Порівняння CMS. Частина перша
6	https://pinta.com.ua/blog/the-agile-methodology-12	Семантичне ядро

Рис. 2.48. Таблиця результатів з порожніми описами

17) Мета описи що повторюються – відображає у таблиці усі повторення мета-описів. Мета-описи напряму впливають на ранжування. Але необхідно використовувати унікальні описи, котрі не мають ключових слів. Це дозволяє збільшити видимість сайту та завлїкати більшу кількість користувачів (рис. 2.49).



Duplicated Descriptions			Сохранить как..
Page	Title	Description	
https://pinta.com.ua/vakansii/html-verstalshhik	Sales - менеджер	Детальніше	
https://pinta.com.ua/vakansii/jun-php-programmist	Junior PHP розробник	Детальніше	
https://pinta.com.ua/vakansii/free-vacancy	Вільна вакансія	Детальніше	
https://pinta.com.ua/vakansii/middle-php-programmist	Middle PHP розробник	Детальніше	

Рис. 2.49. Таблиця результатів дубльованих описів

18) Надто довгі мета-описи – відображає усі занадто довгі мета-описи. Якщо мета-опис надто довгий (більше 320 символів), у видачі буде відображатися лише тільки його частина. Та ця інформація не завжди є зрозумілою та структурованою для користувача.

Второю головною вкладкою є вкладка звітів (рис. 2.50):

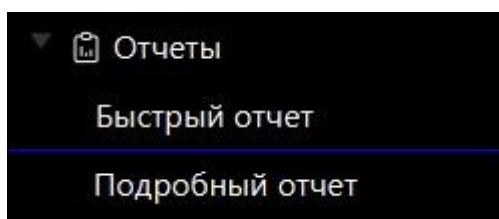


Рис. 2.50. Вкладка «Звіти»

У ній знаходяться 2 вкладки:

- Швидкий звіт;
- Детальний звіт.

Швидкий звіт генерує нову сцену із інформацією про кількість недоліків, скріншотом сайту, графіком (рис. 2.51):

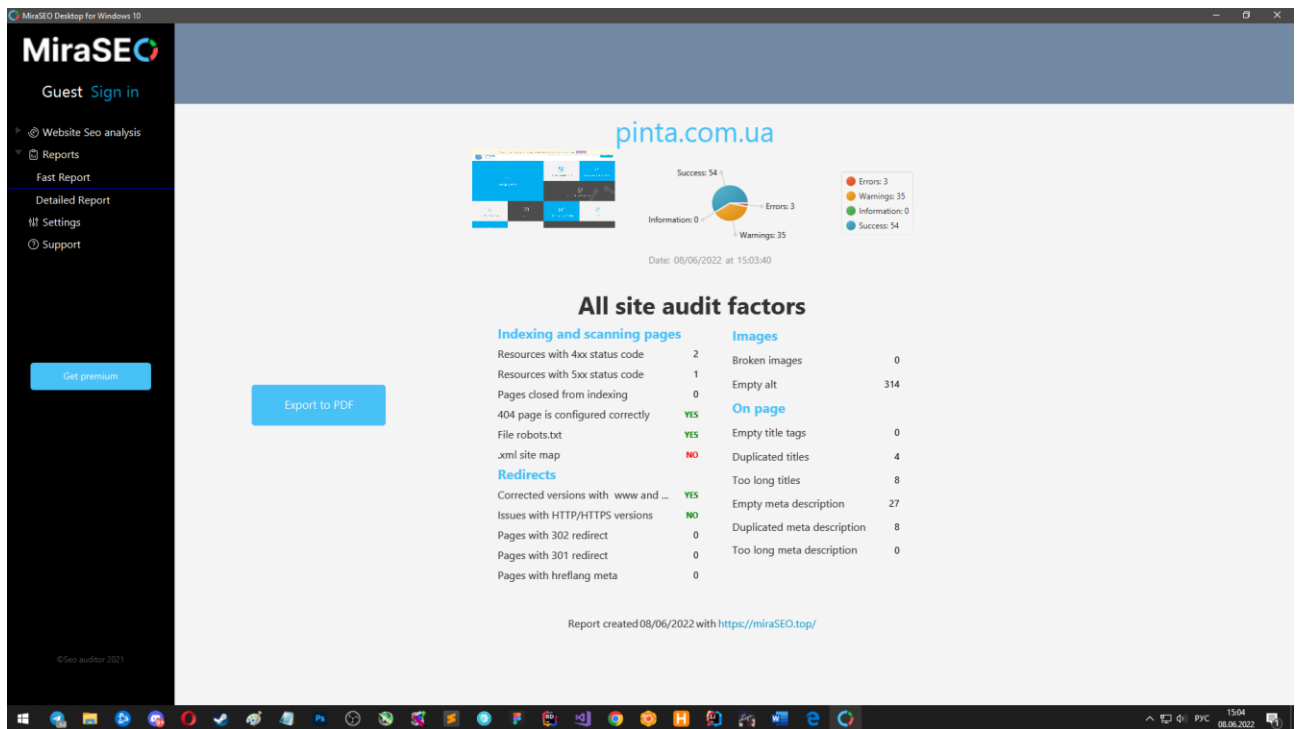


Рис. 2.51. Вкладка швидкого звіту

Також тут існує кнопка завантаження цього звіту у pdf форматі на локальну машину користувача, котрий він може використовувати у майбутньому у зручному форматі.

Створення цього звіту реалізовано за допомогою бібліотеки ІТЕХТ. У коді ми створюємо таблицю та додаємо до неї клітинки, до котрих заноситься інформація зі змінних.

Для користувачів з безкоштовною версією, звіти створюються з вотермаркою та впливає сцена після створення звіту, що пропонує придбати покращену версію додатку, яка дозволяє завантажувати звіти у всіх доступних форматах та без вотермаки.

Результат перевірки сайту pinta.com.ua (рис. 2.52):

Report pinta.com.ua



All site audit factors

Indexing and scanning pages

Resources with 4xx status code	2
Resources with 5xx status code	1
Pages closed from indexing	0
404 page is configured correctly	Yes
File robots.txt	Yes
.xml site map	Yes

Redirects

Corrected versions with www and without www	Yes
Issues with HTTP/HTTPS versions	Yes
Pages with 302 redirect	0
Pages with 301 redirect	0
Pages with hreflang meta	0

Images

Broken images	0
Empty alt	314

On Page

Empty title tags	0
Duplicated titles	4
Too long titles	8
Empty meta description	27
Duplicated meta description	8
Too long meta description	0

Report created 09/06/2022 with <https://miraseo.top/>

Рис. 2.52. Готовий швидкий звіт

Детальний звіт є компіляцією усіх таблиць та блоків, які знаходяться у вкладці «Аудит сайту», та предствляє собою довгий скроллер (рис. 2.53):

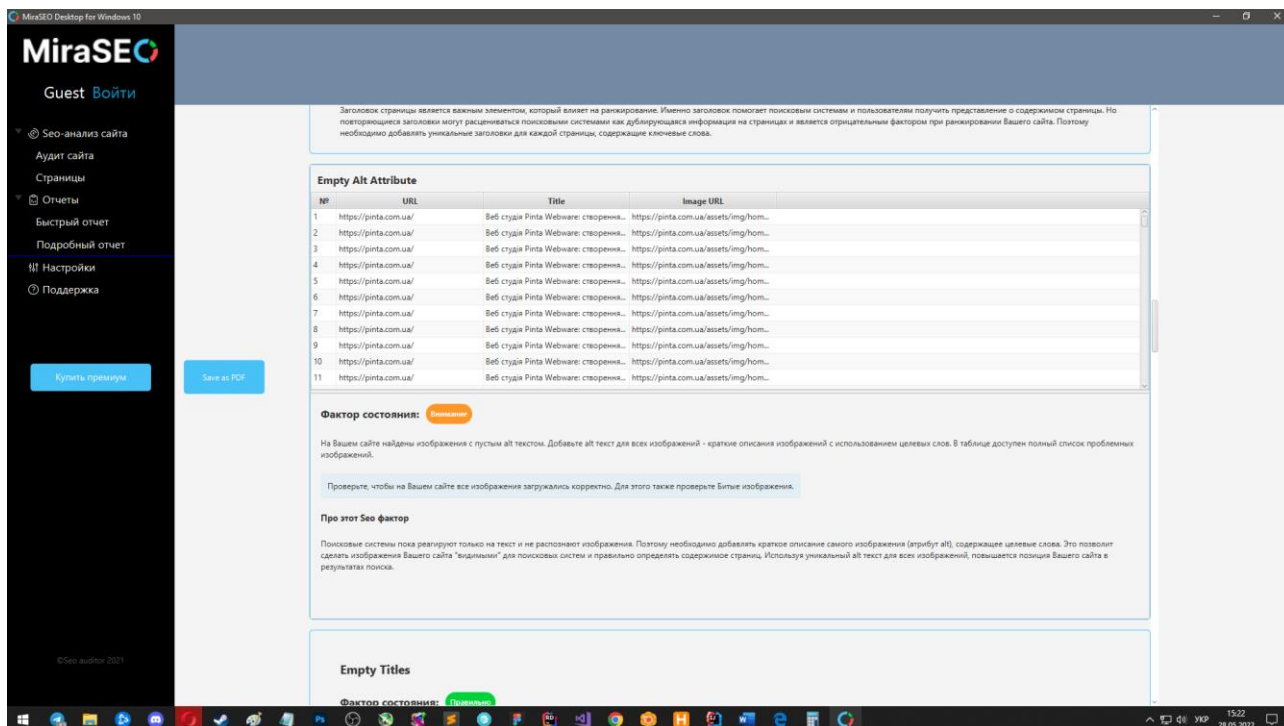


Рис. 2.53. Вкладка детального звіту

Така таблиця виводиться для безкоштовних користувачів, після завантаження звітів (рис. 2.54):

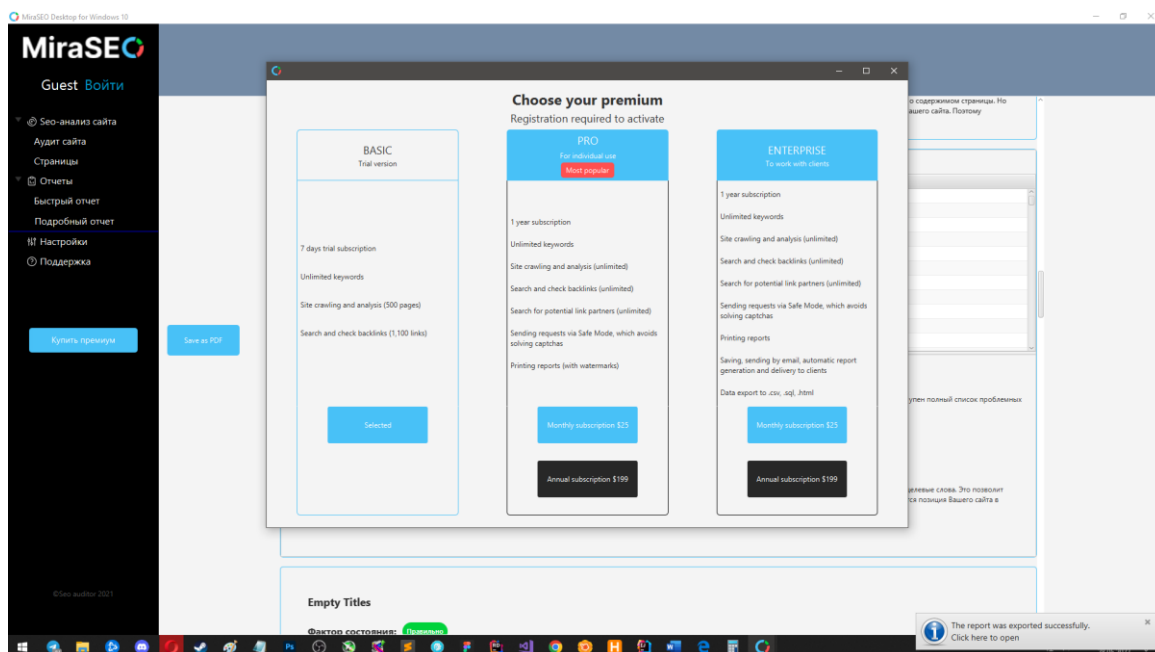


Рис. 2.54. Вікно, яке впливає для безкоштовних користувачів

Також, після завантаження звітів у обрану дирикторію, справа внизу впливає оповіщення про те що усе гаразд або ні якщо виникла помилка (рис. 2.55):

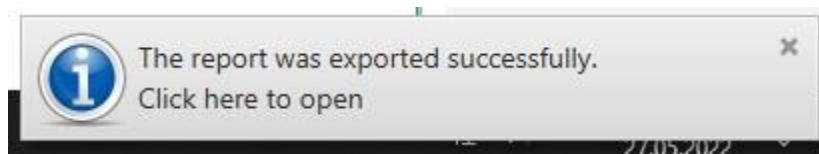


Рис. 2.55. Оповіщення

Також можна завантажити і цей звіт у PDF форматі натиснувши кнопку зліва від таблиці. Зовнішній вид цього звіту (рис. 2.56):

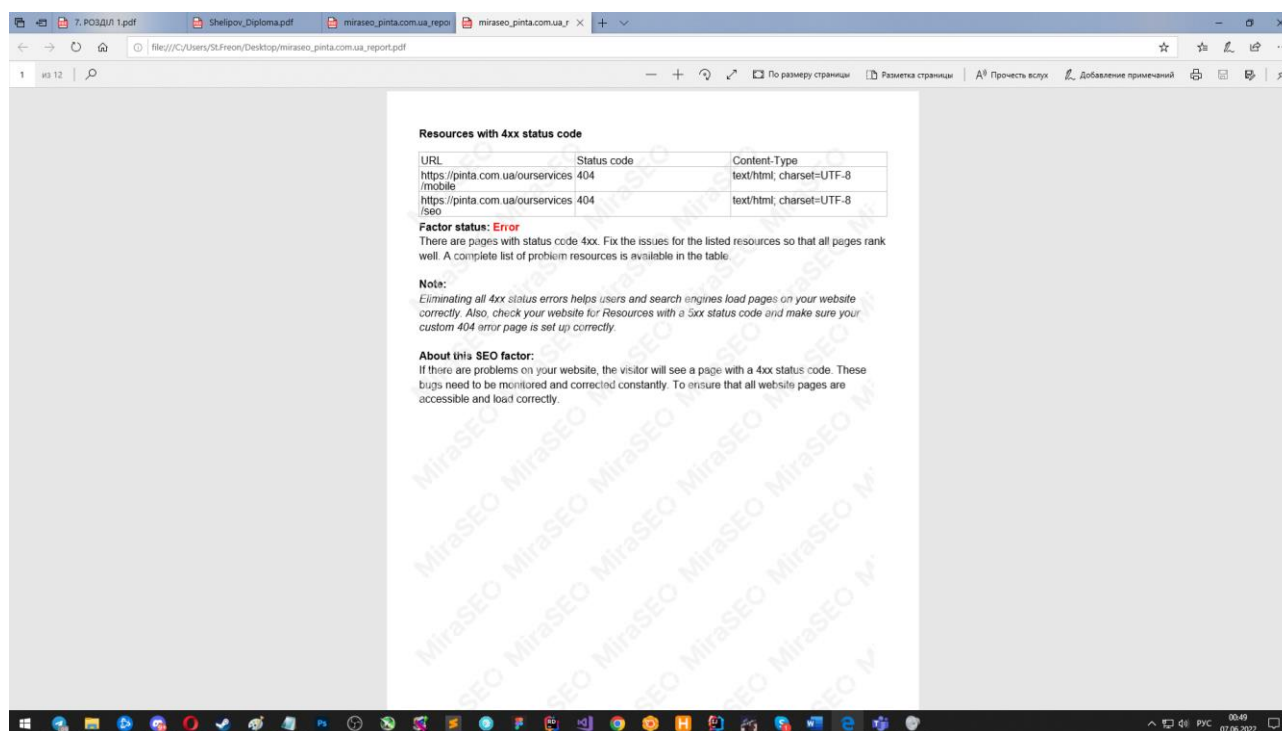


Рис.2.56. Приклад детального звіту

Та 19 інших листів з інформацією, згідно усім критеріям що реалізує програма у звітах.

Завантаження звітів також може проходити безпосередньо певної таблиці, наприклад таблиця з результатами пошуку 301 перенаправлень (рис. 2.57):

Redirect 301			Сохранить как..
№	URL	Redirect to	
1	https://pinta.com.ua/marketing/ppc/test	https://pinta.com.ua/ppc	

Рис. 2.57. Таблица з результатами 301 перенаправлення

Для завантаження безпосередньо таблиці можна вибрати 2 формати - .xlsx та .csv (рис. 2.58):



Рис. 2.58. Меню вибору формату завантаження звіту

Результат завантаження звіту у .xlsx форматі (рис. 2.59):

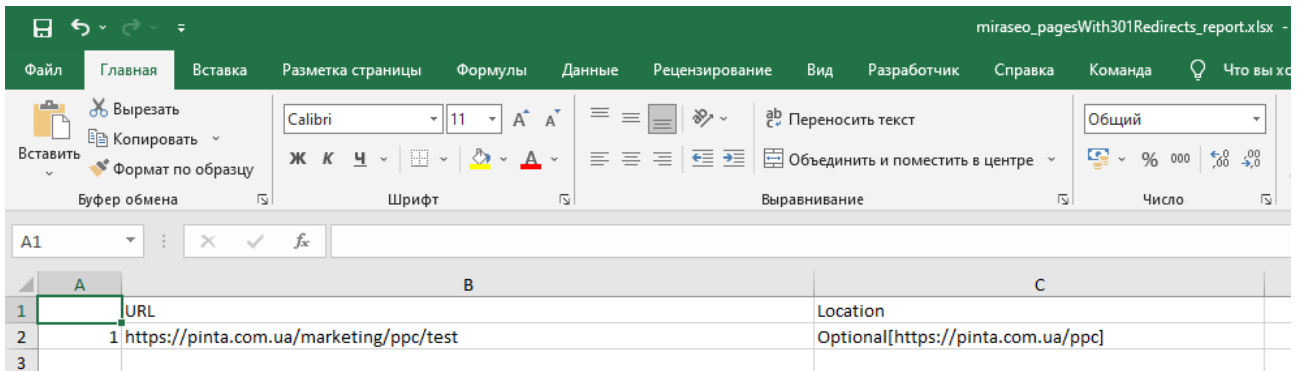


Рис. 2.59. Приклад файлу відкритого у .xlsx форматі

Результат завантаження звіту у .csv форматі (рис. 2.60):

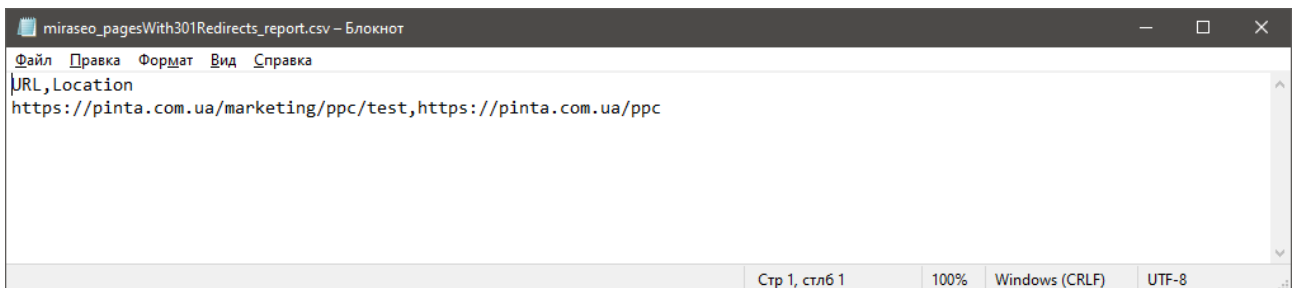


Рис. 2.60. Приклад файлу відкритого у .csv форматі

Для користувача доступна вкладка Налаштування (рис. 2.61):

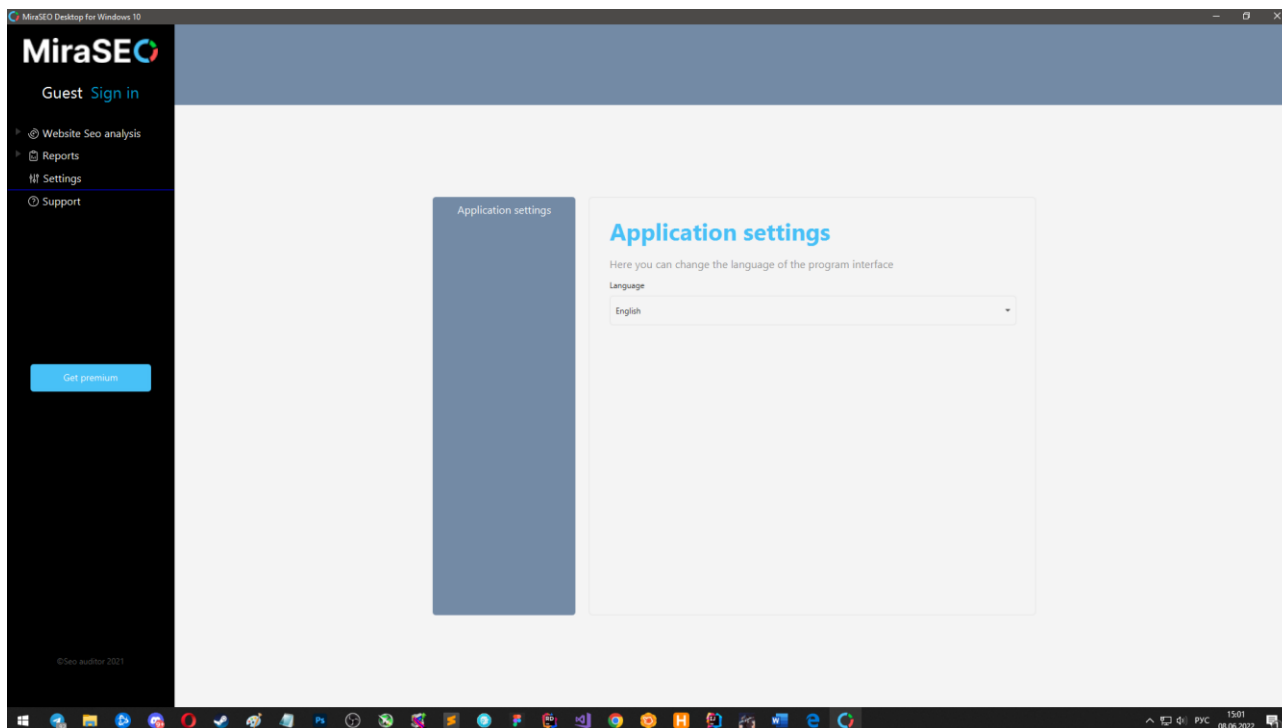


Рис. 2.61. Вкладка з налаштуваннями

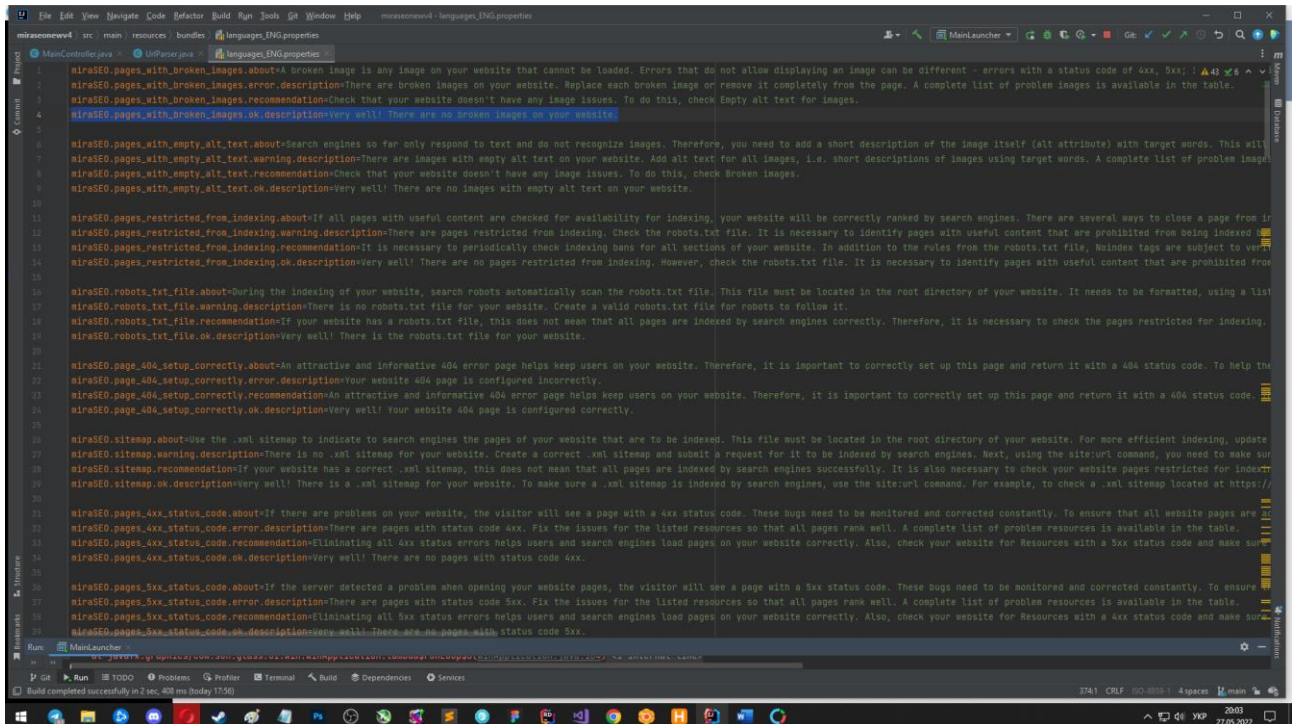
У ній знаходяться такі пункти:

- Налаштування застосунку.

У налаштуваннях додатку знаходиться функція зміни мови додатку українська/російська/англійська.

Зміна мов виконується завдяки підключенню створених мовних пакетів, а її збереження при наступному запуску програми реалізовано за допомогою механізму серіалізації.

Приклад коду ресурсу з англійською мовою (рис. 2.62):



```
miraseo.pages_with_broken_images.about<A broken image is any image on your website that cannot be loaded. Errors that do not allow displaying an image can be different - errors with a status code of 4xx, 5xx: >
miraseo.pages_with_broken_images.error.description<There are broken images on your website. Replace each broken image or remove it completely from the page. A complete list of problem images is available in the table.
miraseo.pages_with_broken_images.recommendation<Check that your website doesn't have any image issues. To do this, check Empty alt text for images.
miraseo.pages_with_broken_images.ok.description<Very well! There are no broken images on your website.

miraseo.pages_with_empty_alt_text.about<Search engines do not only respond to text and do not recognize images. Therefore, you need to add a short description of the image itself (alt attribute) with target words. This will
miraseo.pages_with_empty_alt_text.warning.description<There are images with empty alt text on your website. Add alt text for all images, i.e. short descriptions of images using target words. A complete list of problem images
miraseo.pages_with_empty_alt_text.recommendation<Check that your website doesn't have any image issues. To do this, check Broken Images.
miraseo.pages_with_empty_alt_text.ok.description<Very well! There are no images with empty alt text on your website.

miraseo.pages_restricted_from_indexing.about<If all pages with useful content are checked for availability for indexing, your website will be correctly ranked by search engines. There are several ways to close a page from
miraseo.pages_restricted_from_indexing.warning.description<There are pages restricted from indexing. Check the robots.txt file. It is necessary to identify pages with useful content that are prohibited from being indexed
miraseo.pages_restricted_from_indexing.recommendation<It is necessary to periodically check indexing bans for all sections of your website. In addition to the rules from the robots.txt file, Noindex tags are subject to verification
miraseo.pages_restricted_from_indexing.ok.description<Very well! There are no pages restricted from indexing. However, check the robots.txt file. It is necessary to identify pages with useful content that are prohibited from

miraseo.robots_txt_file.about<During the indexing of your website, search robots automatically scan the robots.txt file. This file must be located in the root directory of your website. It needs to be formatted, using a list
miraseo.robots_txt_file.warning.description<There is no robots.txt file for your website. Create a valid robots.txt file for robots to follow it.
miraseo.robots_txt_file.recommendation<If your website has a robots.txt file, this does not mean that all pages are indexed by search engines correctly. Therefore, it is necessary to check the pages restricted for indexing.
miraseo.robots_txt_file.ok.description<Very well! There is the robots.txt file for your website.

miraseo.page_404_setup_correctly.about<An attractive and informative 404 error page helps keep users on your website. Therefore, it is important to correctly set up this page and return it with a 404 status code. To help the
miraseo.page_404_setup_correctly.error.description<Your website 404 page is configured incorrectly.
miraseo.page_404_setup_correctly.recommendation<An attractive and informative 404 error page helps keep users on your website. Therefore, it is important to correctly set up this page and return it with a 404 status code.
miraseo.page_404_setup_correctly.ok.description<Very well! Your website 404 page is configured correctly.

miraseo.sitemap.about<Use the .xml sitemap to indicate to search engines the pages of your website that are to be indexed. This file must be located in the root directory of your website. For more efficient indexing, update
miraseo.sitemap.warning.description<There is no .xml sitemap for your website. Create a correct .xml sitemap and submit a request for it to be indexed by search engines. Next, using the site:url command, you need to make sure
miraseo.sitemap.recommendation<If your website has a correct .xml sitemap, this does not mean that all pages are indexed by search engines successfully. It is also necessary to check your website pages restricted for indexing
miraseo.sitemap.ok.description<Very well! There is a .xml sitemap for your website. To make sure a .xml sitemap is indexed by search engines, use the site:url command. For example, to check a .xml sitemap located at https://

miraseo.pages_4xx_status_code.about<If there are problems on your website, the visitor will see a page with a 4xx status code. These bugs need to be monitored and corrected constantly. To ensure that all website pages are
miraseo.pages_4xx_status_code.error.description<There are pages with status code 4xx. Fix the issues for the listed resources so that all pages rank well. A complete list of problem resources is available in the table.
miraseo.pages_4xx_status_code.recommendation<Eliminating all 4xx status errors helps users and search engines load pages on your website correctly. Also, check your website for Resources with a 5xx status code and make sure
miraseo.pages_4xx_status_code.ok.description<Very well! There are no pages with status code 4xx.

miraseo.pages_5xx_status_code.about<If the server detected a problem when opening your website pages, the visitor will see a page with a 5xx status code. These bugs need to be monitored and corrected constantly. To ensure
miraseo.pages_5xx_status_code.error.description<There are pages with status code 5xx. Fix the issues for the listed resources so that all pages rank well. A complete list of problem resources is available in the table.
miraseo.pages_5xx_status_code.recommendation<Eliminating all 5xx status errors helps users and search engines load pages on your website correctly. Also, check your website for Resources with a 4xx status code and make sure
miraseo.pages_5xx_status_code.ok.description<Very well! There are no pages with status code 5xx.
```

Рис. 2.62. Приклад коду ресурсу з мовами

У цих файлах прописаний текст для кожної фрази, лейблу, кнопки, тексту, тощо.

Змінений інтерфейс програми (з російської на англійську) (рис. 2.63).

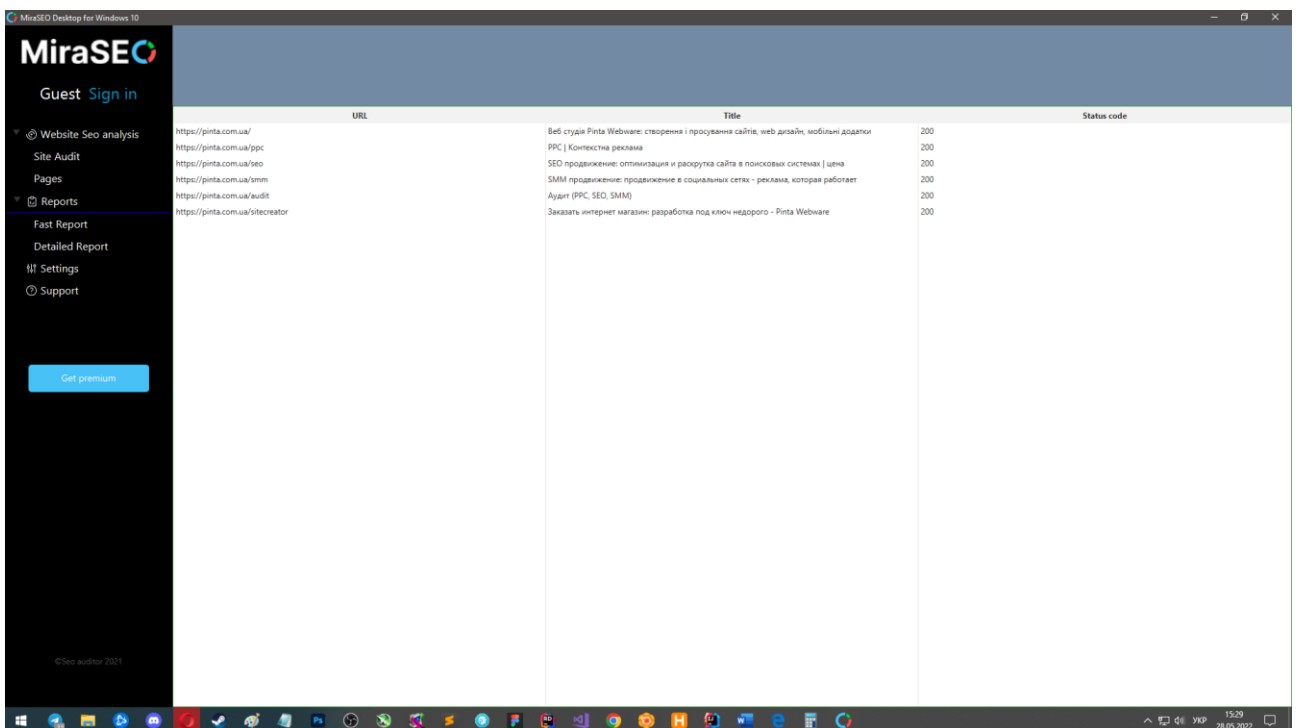


Рис. 2.63. Змінена мова інтерфейсу після приміненого налаштування

Для зміни мови додатку необхідно перезавантажити програму.

Якщо користувач зустрівся з будь-якими проблемами, або має зауваження, пропозиції, тощо.

То він може перейти до вкладки «Підтримка» (рис. 2.64):

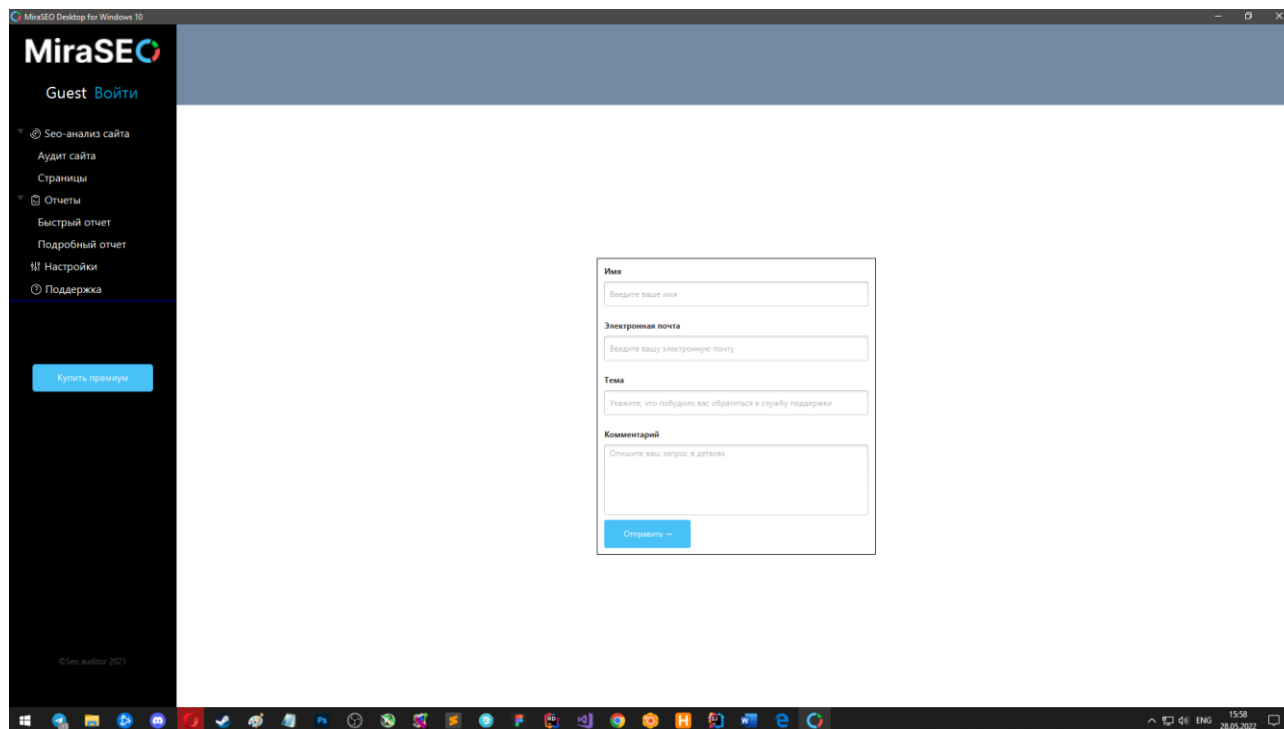


Рис. 2.64. Вкладка з обратним зв'язком

Для того, щоб зробити запит до підтримки додатку, необхідно заповнити усі необхідні поля вводу:

- Ім'я користувача;
- Електронна пошта;
- Тема запиту;
- Опис проблеми у вигляді коментарю.

Запит відправляється на акаунт компанії, який зареєстрований у Zendesk:

Оператор, який займається цим акаунтом, побачить що надійшов запит на підтримку, відповідь на нього як тільки зможе. Та користуючись даними для зворотнього зв'язку, яку надав користувач додатку – відповідь йому згідно теми запиту.

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

Вихідні дані розробки програмного забезпечення:

- а) передбачувана кількість операторів – 1500;
- б) коефіцієнт складності програми – 1,75;
- в) коефіцієнт кореляції програми в ході її розробки – 0,06;
- г) середня годинна заробітня плата програміста, грн/год – 45;
- д) коефіцієнт кваліфікації програміста, обумовлений від стажу – 1,1;
- е) вартість машино-годин ЕОМ, грн/год – 4,2;
- ж) Коефіцієнт збільшення витрат праці в наслідок недостатнього опису задачі (1,2...1,5) - 1,2;

3.1. Визначення тромісткості розробки програмного забезпечення

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_{\delta}, \text{ людино-годин,} \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 60 людино-годин);

t_u – витрати праці на дослідження алгоритму рішення задачі,

t_a – витрати праці на розробку блок-схеми алгоритму,

t_n – витрати праці на програмування по готовій блок-схемі,

t_{oml} – витрати праці на налагодження програми на ЕОМ,

t_{δ} – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = qC(1+p), \text{ людино-годин,} \quad (3.2)$$

де q – передбачуване число операторів,

C – коефіцієнт складності програми,

p – коефіцієнт корекції програми в ході її розробки.

$$Q = 1500 * 1,75 * (1 + 0,06) = 2\,782,5$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t = \frac{Q * B}{(75..85) * k}, \text{ людино-години,} \quad (3.3)$$

де B , яке дорівнює 1,2 - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі,

k , яке дорівнює 1,2, - коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності.

$$t = \frac{2782,5 * 1,2}{82 * 1,1} = 37,02, \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t = \frac{Q}{(20..25) * k}; \quad (3.4)$$

$$t = \frac{2782}{22 * 1,1} = 114,95, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t = \frac{q}{(20..25)*k}; \quad (3.5)$$

$$t = \frac{2782}{21*1,2} = 110,39; \text{ людино-годин};$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t = \frac{q}{(4..5)*k}; \quad (3.6)$$

$$t = \frac{2782}{5*1,1} = 505,81, \text{ людино-годин},$$

– за умови комплексного налагодження завдання;

$$t^k_{oml} = 1,4*t_{oml}; \quad (3.7)$$

$$t^k_{oml} = 1,4*505,81 = 708,134, \text{ людино годин.}$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \quad (3.8)$$

де t_{∂} – трудомісткість підготовки матеріалів і рукопису:

$$t = \frac{q}{(15..20)*k}; \quad (3.9)$$

$$t = \frac{2782}{20*1,1} = 126,45, \text{ людино-годин}$$

де $t_{\partial o}$ – трудомісткість редагування, печатки й оформлення документації:

$$t_{до} = 0,75 * t_{оп}; \quad (3.10)$$

$$t_{до} = 0,75 * 126,45 = 94,83, \text{ людино-годин.}$$

$$t_{\partial} = 126,45 + 94,83 = 221,28, \text{ людино-годин.}$$

Отримаємо трудомісткість розробки програмного забезпечення:

$$t_{\partial} = 60 + 37,02 + 114,95 + 110,39 + 505,81 + 221,28 = 1049,45, \text{ людино-годин.}$$

У результаті ми розрахували, що в загальній кількості необхідно 1116,86 людино-годин для розробки даного програмного забезпечення.

3.2 Розрахунок витрат на створення програми

Витрати на створення ПЗ $K_{ПО}$ включають витрати на заробітну плату виконавця програми $Z_{ЗП}$ і витрат машинного часу, необхідного для налагодження програми на ЕОМ.

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн.} \quad (3.11)$$

де $Z_{ЗП}$ – заробітна плата виконавців, яка визначається за формулою:

$$Z_{ЗП} = t * C_{ПР}, \text{ грн.} \quad (3.12)$$

де t – загальна трудомісткість людино-годин,

$C_{ПР}$ – середня годинна заробітня плата програмісті, грн/година. Моя середня заробітня плата Java-розробника становить 250\$(1.5\$ в годину)? Що у перерахунку на грн/год становить 45 грн/год.

$$Z_{3П} = 45 * 1049,45 = 47\,225,25, \text{ грн.}$$

Z_{MB} - вартість машинного часу, необхідно для налагодження програми на ЕОМ:

$$Z_{MB} = t_{омл} * C_{МЧ}, \text{ грн.} \quad (3.13)$$

де $t_{омл}$ – трудомісткість налагодження програми на ЕОМ, год.

$C_{МЧ}$ - вартість машино-години ЕОМ, грн/год., дорівнює 4,2.

$$Z_{MB} = 505,81 * 4,2 = 2\,124,4, \text{ грн.}$$

$$K_{ПО} = 47\,225,25 + 2\,124,4 = 49\,349,65 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k * F_p}, \text{ мес,} \quad (3.14)$$

де B_k – число виконавців,

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні $F_p = 176$ годин).

$$T = \frac{1049,45}{1 * 176} = 5,96 \text{ міс.}$$

Висновки: час розробки даного програмного забезпечення складає 1049,45 людино-годин. Таким чином, очікувана тривалість розробки складе 5,96 місяця при 40 годинному робочому тижні (місячний фонд робочого часу 176 годин), а витрати на створення програмного забезпечення складатимуть 49 349,65 грн.

ВИСНОВКИ

Метою даної кваліфікаційної роботи є створення десктопного додатку, який буде знаходити недоліки у реалізації аспектів SEO для її подальшого виправлення.

Додаток може бути використаний власниками веб-ресурсів, компаніями які просувають свій продукт або заінтересованими людьми для пошуку недоліків на сайтах, які призводять до пониження ранжування сайтів. Також додаток може бути використаний в якості основного інструменту для оптимізації, тими хто не володіє великою кількістю ресурсів.

Цей продукт адаптований під усі актуальні десктопні операційні системи, платформи (Windows, MacOS, Linux). Не є вибагливим до конфігурації та працює стабільно. Надає можливість створити персональний акаунт, придбати покращену версію додатку, проаналізувати обраний веб-ресурс, та отримати звіт у зручному форматі.

Для досягнення поставленої мети вирішено наступні питання:

- аналіз предметної області;
- уточнення вимог для характеристик та параметрів інформаційної системи;
- уточнення вимог для продуктивності додатку;
- розробка архітектури компонентів додатку.

Для створення додатку, були використані такі основні програмні засоби:

- Java – мова програмування використана для розробки додатку;
- JavaFX – набір інструментів для реалізації графічної складової додатку;
- SceneBuilder – інструмент для конструювання інтерфейсу;
- Maven – конструктор проекту;
- Jsoup – бібліотека для роботи із HTML-файлами;
- GitLab – репозиторій для проекту;
- ITEXT – бібліотека для роботи з PDF файлами;
- Apache POI – бібліотека для роботи з .XLSX файлами;

- Log4j – бібліотека для логування;
- MongoDB – для зберігання даних про користувача.

Згідно проведеного аналізу та поставленого завдання були вирішені такі основні задачі та вповнені вимоги до додатку:

- функціональність;
- стабільність роботи;
- швидкість обробки даних;
- зрозумілий та зручний інтерфейс;
- можливість реєстрації користувача;
- додаткові функції для преміум користувача;
- завантаження звітів роботи додатку;
- перегляд результатів безпосередньо у самому додатку.

Визначено трудомісткість розробленої інформаційної системи (1049,45 людино-годин), проведений підрахунок вартості роботи по створенню програми (49 349,65 грн.) та розраховано час на його створення (5,96 міс).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Forrester: 84% Of U.S. Adults Now Use The Web Daily [Електронний ресурс]/ URL : <https://techcrunch.com/2012/12/19/forrester-84-of-u-s-adults-now-use-the-web-daily-50-own-smartphones-tablet-ownership-doubled-to-19-in-2012/>.
2. Пошукова оптимізація [Електронний ресурс]/URL: https://ua.wikipedia.org/wiki/Пошукова_оптимізація.
3. Цілі та завдання seo просування сайту [Електронний ресурс] / URL: <https://meduza.org/stati/czeli-i-zadachi-seo/>.
4. Цілі та завдання seo просування сайту [Електронний ресурс] / URL: <https://meduza.org/stati/czeli-i-zadachi-seo/>.
5. What is Black Hat SEO? [Електронний ресурс] / URL: <https://elit-web.com/blog/chto-takoe-black-hat-seo>.
6. Internal ranking factors - how to make your site understandable for search engines [Електронний ресурс] / URL: <https://netpeak.net/ru/blog/kak-sdelat-sayt-ponyatnym-dlya-poiskovykh-sistem-seo-dlya-novichkov>.
7. Каталог сайтів для підвищення рейтингу сайту [Електронний ресурс] / URL: https://catalog.tools/blog/chto_takoe_katalog_sajtov/2022-01-24-59.
8. Семантичне ядро: як зібрати та покращити SEO-просування [Електронний ресурс] / URL: <https://roistat.com/rublog/semanticheskoe-yadro/>.
9. Eric Enge, Stephan Spencer, and Jessie C. Stricchiola SEO: The Art of SEO: Mastering Search Engine Optimization Third Edition. O`Reilly Media, Inc., 2017. 162p.
10. Розширюйте бізнес із Google Ads [Електронний ресурс] / URL: https://ads.google.com/intl/ua_ua/home/.
11. SEO-гайд: оптимізація сайту у 2021 році [Електронний ресурс] / URL: <https://promo.altera-media.com/information/expert/seo-gajd-optimizaciya-sajta-v-2021-godu/>.
12. Що таке пошуковий робот і як він працює [Електронний ресурс] / URL: <https://netpeaksoftware.com/ru/blog/chto-takoe-poiskoviy-robot-i-kak-on-rabotaet>

13. Як створити карту сайту - посібник з SEO-картографії [Електронний ресурс] / URL: <https://seranking.com/ru/blog/kak-sozdat-kartu-sajta/>.
14. SEF формат url [Електронний ресурс] / URL: <https://imt-academy.com/a/sef-format-url/>.
15. Maven [Електронний ресурс] / URL: <https://blog.skillfactory.ru/glossary/maven/>.
16. Що таке JavaFX і як він допоможе при створенні програм [Електронний ресурс] / URL: <https://highload.today/chto-takoe-javafx-i-kak-on-pomozhet-pri-sozdanii-prilozhenij/>
17. jsoup – Короткий посібник [Електронний ресурс] / URL: <https://coderlessons.com/tutorials/java-tehnologii/uznaite-jsoup/jsoup-kratkoe-rukovodstvo>.
18. IText [Електронний ресурс] / URL: <https://ru.wikipedia.org/wiki/IText>.
19. IText – обзор [Електронний ресурс] / URL: <https://coderlessons.com/tutorials/raznoe/uznaite-itext/itext-obzor>.
20. Apache POI – Короткий посібник [Електронний ресурс] / URL: <https://coderlessons.com/tutorials/java-tehnologii/izuchite-apache-poi/apache-poi-kratkoe-rukovodstvo>.
21. jsoup – короткий посібник [Електронний ресурс] / URL: <https://coderlessons.com/tutorials/java-tehnologii/uznaite-jsoup/jsoup-kratkoe-rukovodstvo>.
22. GitLab [Електронний ресурс] / URL: <https://itglobal.com/ru-ru/company/glossary/gitlab/>.
23. launch4j [Електронний ресурс] / URL: <http://launch4j.sourceforge.net/>.

КОД ПРОГРАМИ

BrokenImagesScene.java

```
package controllers.rigthside;

import models.AppState;
import models.UserData;
import models.states.BrokenImages;

import java.awt.*;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.ResourceBundle;

import javafx.fxml.FXMLLoader;
import javafx.geometry.Insets;
import javafx.geometry.Orientation;
import javafx.geometry.Pos;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.Menu;
import javafx.scene.control.MenuBar;
import javafx.scene.control.MenuItem;
import javafx.scene.control.SplitPane;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.stage.DirectoryChooser;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
import javafx.util.Duration;
import org.apache.commons.csv.CSVFormat;
```

```

import org.apache.commons.csv.CSVPrinter;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.controlsfx.control.Notifications;

public class BrokenImagesScene {

    private final boolean exportButtonVisible;

    public BrokenImagesScene(AnchorPane rightSide,
ResourceBundle resourceBundle, boolean
exportButtonVisible) {
        this.rightSide = rightSide;
        this.resourceBundle = resourceBundle;
        this.exportButtonVisible = exportButtonVisible;

        if (AppState.brokenImages.size() > 0) {
            renderWithTable();
        } else {
            render();
        }
    }

    private void render() {
        rightSide.setPadding(new Insets(30));

        VBox vBox = new VBox(20);
        HBox hBox = new HBox(15);

        Label criterion = new Label("Broken Images");
        criterion.setFont(Font.font("System",
FontWeight.BOLD, 18));

        Label statusFactor = new
Label(resourceBundle.getString("panel.pageAudit.recommend
ations.factorStatus.label.text"));
        statusFactor.setFont(Font.font("System",
FontWeight.BOLD, 16));
        statusFactor.setWrapText(true);

        Label status = new
Label(resourceBundle.getString("ui.ok_status"));
        status.setFont(Font.font("System",
FontWeight.BOLD, 11));

```

```

        status.setWrapText(true);
        status.setStyle("-fx-background-color: #00D23B;"
            + "-fx-background-radius: 100px;"
            + "-fx-padding: 7 7 7 7;"
            + "-fx-text-fill: white;");
        hBox.getChildren().addAll(statusFactor, status);
        hBox.setAlignment(Pos.CENTER_LEFT);
        hBox.setSpacing(5);

        Label successText = new
Label(resourceBundle.getString("miraSEO.pages_with_broken
_images.ok.description"));
        successText.setWrapText(true);

        Label remark = new
Label(resourceBundle.getString("miraSEO.pages_with_broken
_images.recommendation"));
        remark.setWrapText(true);
        remark.setStyle("-fx-background-color: rgba(72,
193, 247, 0.1);" +
            "-fx-padding: 10px");

        Label aboutSEOFactor = new
Label(resourceBundle.getString("panel.pageAudit.recommend
ations.aboutFactor.label.text"));
        aboutSEOFactor.setWrapText(true);
        aboutSEOFactor.setFont(Font.font("System",
FontWeight.BOLD, 13));

        Label description = new
Label(resourceBundle.getString("miraSEO.pages_with_broken
_images.about"));
        description.setWrapText(true);
        description.setMaxWidth(rightSide.getMaxWidth());

        vBox.getChildren().addAll(criterion, hBox,
successText, remark, aboutSEOFactor, description);
        AnchorPane.setTopAnchor(vBox, 15.0);
        AnchorPane.setRightAnchor(vBox, 15.0);
        AnchorPane.setLeftAnchor(vBox, 15.0);
        rightSide.getChildren().add(vBox);
    }

    private void renderWithTable() {
        SplitPane splitPane = new SplitPane();

```

```

splitPane.orientationProperty().setValue(Orientation.VERTICAL);
    AnchorPane.setLeftAnchor(splitPane, 0.0);
    AnchorPane.setTopAnchor(splitPane, 0.0);
    AnchorPane.setRightAnchor(splitPane, 0.0);
    AnchorPane.setBottomAnchor(splitPane, 0.0);

    AnchorPane up = new AnchorPane();
    MenuBar menuBar = new MenuBar();

    Label label = new Label("Broken Images");
    label.setFont(Font.font("System",
FontWeight.BOLD, 16));
    up.getChildren().add(label);
    AnchorPane.setTopAnchor(label, 10.0);
    AnchorPane.setLeftAnchor(label, 10.0);

    Menu menu = new
Menu(resourceBundle.getString("ui.download_file"));
    menu.setVisible(exportButtonVisible);
    MenuItem menuItem = new
MenuItem(resourceBundle.getString("ui.download_exel_file"
));
    menuItem.setOnAction(actionEvent -> {
        switch (UserData.license) {
            case FREE:
                try {
                    licenseChecker();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                break;
            case PRO:
            case ENTERPRISE:
                try {
                    export2XLSX();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                break;
        }
    });
}

```



```

        MenuItem menuItem1 = new
MenuItem(resourceBundle.getString("ui.download_csv_file")
);
        menuItem1.setOnAction(actionEvent -> {
            switch (UserData.license) {
                case FREE:
                    try {
                        licenseChecker();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    break;
                case PRO:
                case ENTERPRISE:
                    export2CSV();
                    break;
            }
        });
        menu.getItems().add(menuItem);
        menu.getItems().add(menuItem1);
        menuBar.getMenus().add(menu);
        up.getChildren().add(menuBar);

        AnchorPane.setRightAnchor(menuBar, 10.0);
        AnchorPane.setTopAnchor(menuBar, 10.0);

        TableView<BrokenImages> tableView =
drawTableView();

        tableView.setColumnResizePolicy(TableView.CONSTRAINED_RES
IZE_POLICY);
        up.getChildren().add(tableView);
        AnchorPane.setLeftAnchor(tableView, 0.0);
        AnchorPane.setTopAnchor(tableView, 40.0);
        AnchorPane.setRightAnchor(tableView, 0.0);
        AnchorPane.setBottomAnchor(tableView, 0.0);

        AnchorPane down = new AnchorPane();
        drawText(down);
        splitPane.getItems().addAll(up, down);
        rightSide.getChildren().add(splitPane);
    }

    private void drawText(AnchorPane down) {
        down.setPadding(new Insets(15));
    }

```

```

    VBox vbox = new VBox(20);
    HBox hbox = new HBox(15);

    Label statusFactor = new
Label(resourceBundle.getString("panel.pageAudit.recommend
ations.factorStatus.label.text"));
    statusFactor.setFont(Font.font("System",
FontWeight.BOLD, 16));
    statusFactor.setWrapText(true);

    Label status = new
Label(resourceBundle.getString("ui.error_status"));
    status.setFont(Font.font("System",
FontWeight.BOLD, 11));
    status.setWrapText(true);
    status.setStyle("-fx-background-color: red;" +
        "-fx-background-radius: 100px;" +
        "-fx-padding: 7 7 7 7;" +
        "-fx-text-fill: white;");
    hbox.getChildren().addAll(statusFactor, status);
    hbox.setAlignment(Pos.CENTER_LEFT);
    hbox.setSpacing(5);

    Label warningText = new
Label(resourceBundle.getString("miraSEO.pages_with_broken
_images.error.description"));
    warningText.setWrapText(true);

    Label remark = new
Label(resourceBundle.getString("miraSEO.pages_with_broken
_images.recommendation"));
    remark.setWrapText(true);
    remark.setStyle("-fx-background-color: rgba(72,
193, 247, 0.1);" +
        "-fx-padding: 10px");

    Label aboutSEOFactor = new
Label(resourceBundle.getString("panel.pageAudit.recommend
ations.aboutFactor.label.text"));
    aboutSEOFactor.setWrapText(true);
    aboutSEOFactor.setFont(Font.font("System",
FontWeight.BOLD, 13));

```

```

        Label description = new
Label(resourceBundle.getString("miraSEO.pages_with_broken
_images.about"));
        description.setWrapText(true);
        description.setMaxWidth(down.getMaxWidth());

        vbox.getChildren().addAll(hBox, warningText,
remark, aboutSEOFactor, description);
        AnchorPane.setTopAnchor(vBox, 0.0);
        AnchorPane.setRightAnchor(vBox, 0.0);
        AnchorPane.setLeftAnchor(vBox, 0.0);
        down.getChildren().add(vBox);
    }

    private void licenseChecker() throws IOException {
        if (UserData.license == models.License.FREE) {
            Stage getPremiumWindow = new
Stage(StageStyle.DECORATED);
            Parent parent =
FXMLLoader.load(getClass().getResource("/fxml/getPremiumW
indow.fxml"), resourceBundle);
            Scene scene = new Scene(parent);
            getPremiumWindow.setScene(scene);
            getPremiumWindow.getIcons().add(new
javafx.scene.image.Image("images/icon.png"));
            getPremiumWindow.show();
        }
    }

    private void export2XLSX() throws IOException {
        if (UserData.license == models.License.FREE) {
            Stage getPremiumWindow = new
Stage(StageStyle.DECORATED);
            Parent parent =
FXMLLoader.load(getClass().getResource("/fxml/getPremiumW
indow.fxml"), resourceBundle);
            Scene scene = new Scene(parent);
            getPremiumWindow.setScene(scene);
            getPremiumWindow.getIcons().add(new
javafx.scene.image.Image("images/icon.png"));
            getPremiumWindow.show();
        } else {
            DirectoryChooser directoryChooser = new
DirectoryChooser();
            Path pathToSave = null;

```

```

        try {
            pathToSave =
directoryChooser.showDialog(null).toPath();

        } catch (NullPointerException
NullPointerException) {

            System.out.println("Error!");
        }

        String xlsxName =
"miraseo_BrokenImages_report.xlsx";
        Path resolve = pathToSave.resolve(xlsxName);

        try {
            XSSFWorkbook workbook = new
XSSFWorkbook();
            Sheet sheet =
workbook.createSheet("list");
            FileOutputStream fileOutputStream = new
FileOutputStream(resolve.toFile());

            Row row2 = sheet.createRow(0);
            row2.createCell(1).setCellValue("URL");
            row2.createCell(2).setCellValue("Title");
            row2.createCell(3).setCellValue("Title
Length");

            for (int i = 0; i <
AppState.brokenImages.size(); i++) {
                Row row = sheet.createRow(i + 1);
                row.createCell(0).setCellValue(i +
1);

                row.createCell(1).setCellValue(AppState.brokenImages.get(
i).getPage());

                row.createCell(2).setCellValue(AppState.brokenImages.get(
i).getImageURL());

                row.createCell(3).setCellValue(AppState.brokenImages.get(
i).getPageStatusCode());
            }
        }
    }
}

```

```
row.createCell(4).setCellValue(AppState.brokenImages.get(i).getAltText());
```

```
    }  
    workbook.write(fileOutputStream);  
    fileOutputStream.close();  
} catch (IOException exception) {  
    System.out.println("Error!");  
}
```

```
Notifications
```

```
    .create()  
    .text("The report was exported  
successfully.\nClick here to open")  
    .hideAfter(Duration.seconds(5.0))  
    .onAction(event -> {  
        Desktop desktop =  
Desktop.isDesktopSupported() ? Desktop.getDesktop() :  
null;  
        if (desktop != null &&  
desktop.isSupported(Desktop.Action.BROWSE)) {  
            try {  
                desktop.open(new  
File(String.valueOf(resolve.toFile())));  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        } else {  
            try {  
                new ProcessBuilder("xdg-  
open", resolve.toString()).start();  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
    })  
    .showInformation();  
    }  
}
```

```
private void export2CSV() {  
    DirectoryChooser directoryChooser = new  
DirectoryChooser();  
    Path pathToSave = null;
```

```

        try {
            pathToSave =
directoryChooser.showDialog(null).toPath();
        } catch (NullPointerException
NullPointerException) {
            System.out.println("Error!");
        }

        String xlsxName =
"miraseo_brokenImages_report.csv";
        Path resolve = pathToSave.resolve(xlsxName);

        try {
            try (
                BufferedWriter writer =
Files.newBufferedWriter(resolve);
                CSVPrinter csvPrinter = new
CSVPrinter(writer, CSVFormat.DEFAULT
                    .withHeader("URL", "Title",
"Title Length", "ALT Text"));
            ) {
                for (int i = 0; i <
AppState.brokenImages.size(); i++) {

csvPrinter.printRecord(AppState.brokenImages.get(i).getPa
ge(), AppState.brokenImages.get(i).getImageURL(),
AppState.brokenImages.get(i).getPageStatusCode(),
AppState.brokenImages.get(i).getAltText());

                }

                csvPrinter.flush();
            }
        } catch (IOException exception) {
            System.out.println("Error!");
        }

        Notifications
            .create()
            .text("The report was exported
successfully.\nClick here to open")
            .hideAfter(Duration.seconds(5.0))
            .onAction(event -> {

```

```

        Desktop desktop =
Desktop.isDesktopSupported() ? Desktop.getDesktop() :
null;
        if (desktop != null &&
desktop.isSupported(Desktop.Action.OPEN)) {
            try {
                desktop.open(new
File(String.valueOf(resolve.toFile())));
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            try {
                new ProcessBuilder("xdg-
open", resolve.toString()).start();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    })
    .showInformation();
}

private TableView<BrokenImages> drawTableView() {
    TableView<BrokenImages> tableView = new
TableView<>();

    TableColumn<BrokenImages, String> pageColumn =
new TableColumn<>("Page");
    pageColumn.setCellValueFactory(new
PropertyValueFactory<>("page"));

    TableColumn<BrokenImages, String> urlColumn = new
TableColumn<>("Image URL");
    urlColumn.setCellValueFactory(new
PropertyValueFactory<>("imageURL"));

    TableColumn<BrokenImages, String>
httpStatusCodeColumn = new TableColumn<>("HTTP Status
Code");
    httpStatusCodeColumn.setCellValueFactory(new
PropertyValueFactory<>("pageStatusCode"));

    TableColumn<BrokenImages, String> altTextColumn =
new TableColumn<>("Alt text");

```

```

        altTextColumn.setCellValueFactory(new
PropertyConnectionFactory<>("altText"));

        tableView.getColumns().addAll(pageColumn,
urlColumn, httpStatusCodeColumn, altTextColumn);

tableView.getItems().addAll(AppState.brokenImages);

        return tableView;
    }
}

```

LoginController.java

```

package controllers;

import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TabPane;
import javafx.scene.control.TextField;
import javafx.scene.image.Image;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import models.AppState;
import models.Consts;
import models.License;
import models.UserData;
import org.apache.http.HttpEntity;
import org.apache.http.client.methods.HttpPost;
import
org.apache.http.entity.mime.MultipartEntityBuilder;
import org.apache.log4j.Logger;
import org.controlsfx.control.Notifications;
import org.json.JSONObject;
import oshi.SystemInfo;
import oshi.hardware.HardwareAbstractionLayer;

import java.io.IOException;
import java.io.InputStream;
import java.net.URI;
import java.nio.charset.StandardCharsets;

```



```

import java.util.ResourceBundle;
import java.util.UUID;

public class LoginController {
    public AnchorPane mainAnchorPane;
    public PasswordField confirmPasswordRegistration;
    public PasswordField newPassword;
    public PasswordField passwordAuthorization;
    public PasswordField passwordRegistration;
    public TabPane loginTabPane;
    public TextField emailAuthorization;
    public TextField emailRecovery;
    public TextField emailRegistration;
    public TextField recoverToken;
    public TextField userName;

    private final Logger log =
Logger.getLogger(LoginController.class);
    private ResourceBundle resourceBundle;
    public void gotoAuthorization() {
        loginTabPane.getSelectionModel().select(0);
    }

    public void gotoRegistration() {
        loginTabPane.getSelectionModel().select(1);
    }

    public void gotoRecoverPassword() {
        loginTabPane.getSelectionModel().select(2);
    }

    public void signIn() throws IOException {
        String emailAuthorization =
this.emailAuthorization.getText().trim();
        String passwordAuthorization =
this.passwordAuthorization.getText().trim();

        if (emailAuthorization.isEmpty() &&
passwordAuthorization.isEmpty()) {
            Notifications.create().title("Empty required
data").text("The email and password cannot be
empty!").showWarning();
            return;
        }
    }
}

```

```

        if (emailAuthorization.isEmpty()) {
            Notifications.create().title("Empty
email").text("The email cannot be empty!").showWarning();
            return;
        }

        if (passwordAuthorization.isEmpty()) {
            Notifications.create().title("Empty
password").text("The password cannot be
empty!").showWarning();
            return;
        }

        log.info("Signing in...");

        HttpEntity requestBody =
MultipartEntityBuilder.create().addTextBody("login",
emailAuthorization).addTextBody("password",
passwordAuthorization).build();
        HttpPost postRequest = new HttpPost();
        postRequest.setURI(URI.create(Constants.API_LOGIN));
        postRequest.setEntity(requestBody);
        org.apache.http.HttpResponse response =
AppState.httpClient.execute(postRequest);

        log.info("Status: " + response.getStatusLine());

        InputStream responseContent =
response.getEntity().getContent();
        String responseMessage = new
String(responseContent.readAllBytes(),
StandardCharsets.UTF_8);
        log.info("Response message: " + responseMessage);

        JSONObject jsonObject = new
JSONObject(responseMessage);

        if (jsonObject.getBoolean("status")) {
            String token = getAuthToken(responseMessage);
            UserData.setToken(token);
            UserData.setEmail(emailAuthorization);
            UserData.setUUID(getUUID());
        }

```

```

Notifications.create().title("Successful").text("You
signed in successfully!").showInformation();

        Stage startStage = (Stage)
mainAnchorPane.getScene().getWindow();
        startStage.close();

        Stage stage = new Stage();
        FXMLLoader fxmlLoader = new
FXMLLoader(getClass().getResource("/fxml/enterURLScene.fx
ml"), resourceBundle);
        Parent root = fxmlLoader.load();
        EnterUrlController enterURLController =
fxmlLoader.getController();

enterURLController.setResourceBundle(resourceBundle);

        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.setTitle(Constants.APPLICATION_NAME);
        stage.getIcons().add(new
Image("images/icon.png"));
        stage.setMaximized(true);
        UserData.signedIn = true;
        stage.show();

    } else {
        System.err.printf("%s\n\n",
jsonObject.getString("message"));

Notifications.create().text(jsonObject.getString("message
")).showWarning();
    }
}

private String getUUID() {
    SystemInfo systemInfo = new SystemInfo();
    HardwareAbstractionLayer hardware =
systemInfo.getHardware();
    String hardwareUUID =
hardware.getComputerSystem().getHardwareUUID().toLowerCase();

    if (hardwareUUID.equals("unknown")) {

```

```

        String processorID =
hardware.getProcessor().getProcessorIdentifier().getProce
ssorID();
        return
UUID.nameUUIDFromBytes(processorID.getBytes()).toString()
;
    }

    return hardwareUUID;
}

public void signUp() throws IOException {
    String email =
emailRegistration.getText().trim();
    String password =
passwordRegistration.getText().trim();
    String passwordConfirm =
confirmPasswordRegistration.getText().trim();
    String userName = this.userName.getText().trim();

    if (checkData(email, password, passwordConfirm))
{
        log.info("Registration...");

        MultipartEntityBuilder requestBody =
MultipartEntityBuilder
            .create()
            .addTextBody("login", email)
            .addTextBody("password", password);

        if (!userName.isEmpty()) {
            requestBody.addTextBody("fullName",
userName);
        }
        HttpEntity build = requestBody.build();
        HttpPost post = new HttpPost();
        post.setURI(URI.create(Constants.API_REGISTRY));
        post.setEntity(build);
        org.apache.http.HttpResponse response =
AppState.httpClient.execute(post);
        log.info("Request status: " +
response.getStatusLine());
        org.apache.http.HttpResponse httpResponse =
AppState.httpClient.execute(post);
        log.info("Status: " + httpResponse);
    }
}

```

```

        InputStream responseContent =
httpResponse.getEntity().getContent();

        String responseMessage =
            new
String(responseContent.readAllBytes(),
StandardCharsets.UTF_8);

        log.info("Response message: " +
responseMessage + '\n');

        JSONObject jsonObject = new
JSONObject(responseMessage);

        if (jsonObject.getBoolean("status")) {

Notifications.create().title("Successful!").text("You was
registered successfully!").showInformation();

setUserLicense(jsonObject.getString("license"));

loginTabPane.getSelectionModel().select(0);
        } else {

System.err.println(jsonObject.getString("error") + '\n');

Notifications.create().text(jsonObject.getString("error")
).showWarning();
        }
    }

private void setUserLicense(String license) {
    switch (license) {
        case "free":
            UserData.license = License.FREE;
            break;
        case "pro":
            UserData.license = License.PRO;
            break;
        case "enterprise":
            UserData.license = License.ENTERPRISE;
            break;
    }
}

```

```

    }

    public void recoverPasswordWithEmail() throws
IOException {
        String emailRecovery =
this.emailRecovery.getText().trim();

        if (emailRecovery.isEmpty()) {
            Notifications.create().title("Empty
email").text("The email cannot be empty!").showWarning();
            return;
        }

        HttpEntity httpEntity =
MultipartEntityBuilder.create().addTextBody("email",
emailRecovery).build();
        HttpPost post = new HttpPost();
        post.setEntity(httpEntity);

post.setURI(URI.create(Constants.API_RESET_PASSWORD));

        org.apache.http.HttpResponse httpResponse =
AppState.httpClient.execute(post);

        System.out.printf("Recovering password with email
-> [%s]\n", emailRecovery);

        log.info("Status code: " +
httpResponse.getStatusLine());

        InputStream responseContent =
httpResponse.getEntity().getContent();

        String responseMessage = new
String(responseContent.readAllBytes(),
StandardCharsets.UTF_8);
        log.info("Response message: " + responseMessage);

        JSONObject jsonObject = new
JSONObject(responseMessage);

        if (jsonObject.getBoolean("status")) {
            log.info("A verification token was sent to
your email." + '\n');
        }
    }
}

```

```

        Notifications.create().text("Next information
was sent to your email.").showInformation();
        loginTabPane.getSelectionModel().select(3);
    } else {
        System.err.println("User with this email
doesn't exist!" + '\n');
        Notifications.create().title("Invalid
email").text("User with this email doesn't
exist!").showWarning();
    }
}

```

```

    public void recoverPasswordWithToken() throws
IOException {
        String newPassword =
this.newPassword.getText().trim();
        String recoverToken =
this.recoverToken.getText().trim();

        if (newPassword.isEmpty() &&
recoverToken.isEmpty()) {
            Notifications.create().title("Empty required
data").text("The password and token cannot be
empty!").showWarning();
            return;
        }

        if (newPassword.isEmpty()) {
            Notifications.create().title("Empty
password").text("The password cannot be
empty!").showWarning();
            return;
        }

        if (recoverToken.isEmpty()) {
            Notifications.create().title("Empty
token").text("The token cannot be empty!").showWarning();
            return;
        }

        if (newPassword.length() < 6) {
            Notifications.create().title("Short
password").text("The password must be at least 6
characters long!").showWarning();
            return;
        }
    }
}

```

```

    }

    HttpEntity httpEntity =
MultipartEntityBuilder.create().addTextBody("password",
newPassword).addTextBody("token", recoverToken).build();

    HttpPost httpPost = new
HttpPost(URI.create(Consts.API_NEW_PASSWORD));
    httpPost.setEntity(httpEntity);
    httpPost.setHeader(Consts.USER_AGENT,
Consts.APPLICATION_NAME);

    log.info("Recovering...");
    org.apache.http.HttpResponse httpResponse1 =
AppState.httpClient.execute(httpPost);
    log.info("Status code: " +
httpResponse1.getStatusLine());
    InputStream responseContent =
httpResponse1.getEntity().getContent();
    String responseMessage = new
String(responseContent.readAllBytes(),
StandardCharsets.UTF_8);
    log.info("Response message: " + responseMessage);
    JSONObject jsonObject = new
JSONObject(responseMessage);

    if (jsonObject.getBoolean("status")) {
        log.info("Password was reset successfully!" +
'\n');
        Notifications.create().text("Password was
reset successfully!").showInformation();
        loginTabPane.getSelectionModel().select(0);
    } else {

System.err.println(jsonObject.getString("error") + '\n');

Notifications.create().text(jsonObject.getString("error")
+ "\n" + "Please check your token!").showWarning();
    }
}

    private boolean checkData(String email, String
password, String passwordConfirm) {

```



```

        if (email.isEmpty() && password.isEmpty() &&
passwordConfirm.isEmpty()) {
            Notifications.create().title("Empty required
data").text("The email and passwords cannot be
empty!").showWarning();
            return false;
        }

        if (email.isEmpty()) {
            Notifications.create().title("Empty
email").text("The email cannot be empty!").showWarning();
            return false;
        }

        if (!email.contains("@") && !email.contains("."))
{
            Notifications.create().title("Invalid email
format").text("The email should contain '.' and '@'
characters!").showWarning();
            return false;
        }

        if (!email.contains("@") || !email.contains("."))
{
            Notifications.create().title("Invalid email
format").text("The email should contain '.' and '@'
characters!").showWarning();
            return false;
        }

        if (password.isEmpty()) {
            Notifications.create().title("Empty
password").text("The password cannot be
empty!").showWarning();
            return false;
        }

        if (password.length() < 6) {
            Notifications.create().title("Short
password").text("The password must be at least 6
characters long!").showWarning();
            return false;
        }

        if (passwordConfirm.isEmpty()) {

```

```

        Notifications.create().title("Empty
confirmation password").text("The confirmation password
cannot be empty!").showWarning();
        return false;
    }

    if (passwordConfirm.length() < 6) {
        Notifications.create().title("Short
confirmation password").text("The confirmation password
must be at least 6 characters long!").showWarning();
        return false;
    }

    if (!passwordConfirm.equals(password)) {
        Notifications.create().title("Different
passwords").text("The password and confirmation password
are different!").showWarning();
        return false;
    }
    return true;
}

private String getAuthToken(String json) {
    JSONObject jsonObject = new JSONObject(json);
    return jsonObject.has("token") ?
jsonObject.getString("token") : null;
}

public void setResourceBundle(ResourceBundle
resourceBundle) {
    this.resourceBundle = resourceBundle;
}
}

```

Весь інший код можна знайти в файлах проекту.

ВІДГУК

керівника економічного розділу

на кваліфікаційну роботу бакалавра

на тему: «Створення десктопного додатку для SEO-оптимізації сайтів»

студента групи 122-18-1

Волошина Дмитра Андрійовича

Керівник економічного розділу

доцент каф. ПЕП та ПУ, к.е.н

Л. В. Касьяненко

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_Волошин.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом_Волошин.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація Волошин.ppt	Презентація кваліфікаційної роботи