

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 *Самойленка Валентина Михайловича*
(ПІБ)

академічної групи *121-17-1*
(шифр)

спеціальності *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

освітньої програми *Інженерія програмного забезпечення*
(назва освітньої програми)

на тему: *Розробка програмного забезпечення для*
повернення частини коштів від покупки в інтернет-

магазині на мові програмування JavaScript у середовищі розробки IntelliJ IDEA

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>проф. Корнієнко В.І.</i>			
розділів:				
спеціальний	<i>проф. Корнієнко В.І.</i>			
економічний	<i>проф. Вагонова О.Г.</i>			
Рецензент	<i>доц. Шедловський І.А.</i>			
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2021

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

« » 2021 року

ЗАВДАННЯ

на кваліфікаційну роботу

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 121-17-1 Самойленка Валентина Михайловича
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Розробка програмного забезпечення

для повернення частини коштів від покупки в інтернет-

магазині на мові програмування JavaScript у середовищі розробки

IntelliJ IDEA

затверджена наказом ректора НТУ «ДП» від « 07 » 06. 2021 р. № 317-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	<i>На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	13.05.2021 р.
Економічний	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПО й тривалості його розробки</i>	25.05.2021 р.

Завдання видав

(підпис)

проф. Корнієнко В.І.

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Самойленко В.М.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2021 р.

Термін подання кваліфікаційної роботи до ЕК: 16.06.2021 р.

РЕФЕРАТ

Пояснювальна записка: 70 с., 9 рис., 3 дод., 20 джерел.

Об'єкт розробки: плагін браузера, призначений для роботи з кешбек-сервісом.

Мета кваліфікаційної роботи: розробка програмного забезпечення плагіна браузера, призначеного для надання інструментарію для роботи з веб-сервісом для повернення кешбеку.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної області, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано вибір платформи для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування плагіну, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження застосунку, описана робота програми.

В економічному розділі визначено трудомісткість розробленого плагіну, проведений підрахунок вартості роботи по створенню застосунку та розраховано час на його створення.

Практичне значення полягає у створенні плагіна, що забезпечує роботу розширень кешбек-сервісів, і як наслідок – збільшує ефективність роботи інтернет-магазину через заохочення покупців.

Актуальність плагіна для роботи з кешбек-сервісом визначається великим попитом на подібні розробки, що заощаджують кошти і час покупців, нагадуючи про наявність кешбеку на сайті.

Список ключових слів: КЕШБЕК, СЕРВІС, БРАУЗЕР, ІНТЕРНЕТ-МАГАЗИН, САЙТ, JAVA SCRIPT.

ABSTRACT

Explanatory note: 70 pages, 9 pics, 3 apps, 20 sources.

The object of development: a browser plug-in to work with cashback service.

The purpose of the graduation project: is to provide the tools for working with a web service for returning a cashback through the development and implementation of the browser plug-in.

In introduction the current state of the problem was examined, the purpose of the qualification work and the field of its application were specified, the relevance of the topic was justified, and the problem statement was specified.

In the first section the status of the problem was reviewed, analytical overview of the methods and tools of application analysis and design was justified, the choice of plug-in architecture and design solutions used was done.

In the second section the features of a software implementation were overviewed, the background information and application functionality were provided, a description of the plug-in user interface was made.

In economic section labour input and duration of software product development was defined, the estimation of expenses for its creation was executed.

The practical significance of the project is to create a plug-in that ensures the operation of cashback service extensions, and as a result - increases the efficiency of the online store by encouraging customers.

The relevance of the plug-in to work with cashback service is determined by the high demand for such developments, that save money and time of customers, indicating the presence of cashback on the site.

Keywords: CASHBACK, SERVICE, BROWSER, ONLINE STORE, WEBSITE, JAVA SCRIPT.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ.....	10
1.1. Загальні відомості з предметної галузі.....	10
1.2. Призначення розробки та галузь застосування.....	11
1.3. Підстава для розробки.....	12
1.4. Постановка завдання.....	12
1.5. Вимоги до програми або програмного виробу.....	13
1.5.1. Вимоги до функціональних характеристик	13
1.5.2. Вимоги до інформаційної безпеки.....	13
1.5.3. Вимоги до складу та параметрів технічних засобів.....	14
1.5.4. Вимоги до інформаційної та програмної сумісності.....	15
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ.....	16
2.1. Функціональне призначення програми.....	16
2.2. Опис застосованих математичних методів.....	17
2.3. Опис використаної архітектури та шаблонів проектування.....	17
2.4. Опис використаних технологій та мов програмування.....	20
2.5. Опис структури програми та алгоритмів її функціонування.....	23
2.6. Обґрунтування та організація вхідних та вихідних даних програми.....	29
2.7. Опис роботи розробленого програмного продукту.....	30
2.7.1. Використані технічні засоби.....	30
2.7.2. Використані програмні засоби.....	31

2.7.3.	Виклик та завантаження програми.....	31
2.7.4.	Опис інтерфейсу користувача.....	31
РОЗДІЛ 3. ЕКОНОМІЧНИЙ		
РОЗДІЛ.....		
3.1.	Розрахунок трудомісткості та вартості розробки програмного продукту	
3.2.	Розрахунок витрат на створення програми.....	
ВИСНОВКИ.....		43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		44
Додаток А. Код програми.....		46
Додаток Б. Відгук керівника економічного розділу.....		69
Додаток В. Перелік файлів на диску.....		70

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

- БД - база даних;
- ЕОМ - електронно-обчислювальна машина;
- ІС - інформаційна система;
- ПЗ - програмне забезпечення;
- СУБД - система управління базами даних;
- CMS - система управління контентом;
- ІТ - інформаційні технології;
- SEO - пошукова оптимізація.

ВСТУП

Процес повернення частини коштів від покупки, зробленої в інтернет-магазині і, по суті, є одним з видів програми лояльності. В інтернет-торгівлі під терміном «кешбек» мається на увазі знижка у вигляді повернення частини вартості покупки на рахунок покупця в особистому кабінеті кешбек-сервісу. Популярна схема - коли один інтернет-магазин або сервіс платить іншому, який розмістив інформацію про нього, за всіх клієнтів, які прийшли і зробили покупку. Це свого роду партнерська програма: магазин дає невелику знижку, але отримує партнера, готового залучати більше покупців і вкладатися в маркетинг.

Кешбек-сервіс підключається до партнерської програми інтернет-магазину і привертає клієнта в магазин для покупки. За зроблену клієнтом покупку сервіс отримує комісійну винагороду. Більшу частину цієї винагороди кешбек-сервіс віддає клієнту. В інтернеті є багато сервісів, які консолідують кешбек різних торгових майданчиків і дозволяють користувачеві таким чином повертати гроші швидше (за свою комісію).

Останнім часом великої популярності набирають кешбек-портали. Вони являють собою навігаційний майданчик, який перенаправляє покупця на сайти профільних інтернет-магазинів, які укладають з кешбек-агрегатором партнерські угоди. Кешбек-портали часто надають покупцям додатковий сервіс з пошуку відповідного варіанту товару, порівняно цін, знижкових програм та акцій в різних інтернет-магазинах, пропонують кілька зручних варіантів оплати через онлайн-системи, пропонують інші супутні послуги.

При проведенні операції купівлі товару через кешбек-портал клієнт отримує повернення певного відсотка вартості товару саме від кешбек-маркету, а не від продавця цього товару (партнерського інтернет-магазину).

В даний час кешбек в інтернеті є вже досить поширеною моделлю заохочення покупців. Таким чином, актуальною є розробка плагіна браузера, призначеного для роботи з кешбек-сервісом.

Завдання кваліфікаційної роботи та об'єкт її діяльності безпосередньо пов'язані зі спеціальністю 121 «Інженерія програмного забезпечення» та відповідає узагальненій тематиці кваліфікаційних робіт і переліку зазначених компетенцій.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

Плагіни - це невеликі програмні коди, що доповнюють функціонал основного застосунка, для якого вони створюються. Простіше кажучи - це надбудови, доповнення або виправлення, завдяки яким вихідна програма стає краще (функціональніша) з точки зору конкретного користувача. Якщо всі програми спочатку випускати з повним пакетом різних функцій, вони стануть дуже великими і будуть займати зайве місце, витратити ресурси комп'ютерів, що знизить їх швидкодію. Завдяки плагинам цього не відбувається, бо кожен користувач отримує базовий функціонал, а все інше він зможе отримати, скачавши або встановивши потрібне йому розширення. Використання модульної схеми в роботі програм, де кожен плагін можна підключити або відключити, не викликаючи порушень в роботі основних функцій, дозволяє поліпшувати програми не тільки силами розробників, але і силами будь-яких інших програмістів.

Доповнення створюються для різних категорій програм, але найбільш популярні плагіни для застосунків, що працюють з інтернетом, таких як веб-браузери, сайти (точніше для їх систем управління контентом), онлайн-ігри. Деякі кешбек-сервіси також надають своїм клієнтам спеціальні розширення для браузера - кешбек-плагін. Проаналізувавши асортимент подібного ПО в Інтернет-просторі України, можна виділити такі основні програмні продукти:

- payBack (для Google Chrome, Mozilla Firefox, Opera).
- Megabonus (для Google Chrome, Mozilla Firefox, Opera, Safari).
- Letyshops (для Google Chrome).
- ePN (для Google Chrome, Mozilla Firefox, Opera).
- Smarty.Sale (для Google Chrome, Mozilla Firefox, Opera).

Деякі з перерахованих плагінів призначені виключно для нагадування про можливість отримання кешбеку в інтернет-магазині (Smarty.Sale). Інші показують, який відсоток можна отримати з того чи іншого товару інтернет-магазину, не залишаючи сайт магазину (Megabonus, Letyshops, ePN). Деякі кешбек-плагіни дозволяють створювати спеціальне кешбек-посилання, яким можна ділитися з друзями і отримувати кешбек зі своїх покупок і покупок друзів, а також з покупок всіх тих, кому було надане дане кешбек-посилання (payBack).

Якщо користувач користується декількома кешбек-сервісами, то розширення від одного з них може конкурувати з іншими; найбільш ефективним варіантом є використання розширення від єдиного сервісу.

Але попри велику кількість різновидів та опцій, всі кешбек-плагіни успішно виконують одну спільну задачу – заощаджують кошти і час покупців, нагадуючи про наявність кешбеку на сайті. Таким чином, актуальною видається розробка плагіна браузера, призначеного для роботи з кешбек-сервісом.

1.2. Призначення розробки та галузь застосування

Як об'єкт впровадження розроблюваного плагіна може розглядатися будь-який кешбек-сервіс або кешбек-портал, незалежно від кількості магазинів, з якими він підтримує співробітництво, та територіальної належності. Потенційні споживачі продукту, що розробляється, - користувачі кешбек-сервісу.

Розроблений застосунок призначений для:

- нагадування про можливість отримання кешбеку в інтернет-магазині;
- спрощення роботи з кешбек-сервісом;
- заощадження коштів і часу покупців при отриманні кешбеку.

1.3. Підстава для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу.

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

- освітня програма спеціальності 121 «Інженерія програмного забезпечення»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 317-с від 07.06.2021 р;
- завдання на кваліфікаційну роботу на тему: «Розробка програмного забезпечення для повернення частини коштів від покупки, зробленої в інтернет-магазині на мові програмування JavaScript у середовищі розробки IntelliJ IDEA».

1.4. Постановка завдання

Завданням кваліфікаційної роботи є розробка програмного забезпечення плагіна браузера, призначеного для надання інструментарію для роботи з веб-сервісом для повернення кешбеку.

Програма, що розробляється, повинна реалізувати наступні функції:

- повідомлення про можливість отримання кешбеку на сайтах, що переглядаються браузером;
- активація кешбек-сервісу;
- відображення сайтів для потенційної активації кешбек-сервісу у результатах пошукових систем.

Для досягнення поставленої мети необхідно:

- вивчити предметну область розв'язуваної задачі;
- провести порівняльний аналіз можливостей аналогічних застосунків;
- вибрати раціональну структуру і параметри програми;

- розробити структуру вхідних і вихідних даних для проєктованого програмного забезпечення;
- написати програмний код застосунку;
- розробити рекомендації щодо застосування програми.

Дана програма повинна бути створена із використанням фундаментальних шаблонів проєктування.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Для досягнення поставлених цілей програмне забезпечення, що розробляється, повинно підтримувати виконання наступних дій:

- читання інформації з бази даних кешбек-сервісу;
- внесення і редагування інформації у веб-сховище браузера;
- підтримання різних статусів активності сервісу;
- обмін даними з сайтом кешбек-сервісу у фоновому режимі;
- обробка результатів роботи пошукових систем;
- видача повідомлень через показ спливаючих вікон.

Для виконання перерахованих вище функцій у застосунку повинні бути реалізовані:

- можливість інтеграції в веб-браузер;
- наявність типової конфігурації, що забезпечує можливість швидкого введення застосунку в експлуатацію;
- програмно-апаратна переносимість.

1.5.2 Вимоги до інформаційної безпеки

Для уникнення некоректної роботи програми необхідно реалізувати:

- контроль вхідних даних;
- обробку виняткових ситуацій;
- виведення повідомлень про помилки;
- можливість повторного введення даних;
- можливість безперервної роботи протягом не менше 120 годин (5 діб);
- платформну незалежність;
- вірогідність виникнення не більше 2 логічних помилок на 1000 операторів за 1 рік експлуатації;
 - забезпечення неушкодженого стану даних, що зберігаються в базі даних сервісу, у випадку відмови плагіна.

1.5.3 Вимоги до складу та параметрів технічних засобів

Для нормального функціонування програми необхідно, щоб обчислювальна машина, на якій буде функціонувати веб-орієнтований застосунок, відповідала наступним вимогам:

- ✓ процесор класу Intel Xeon з тактовою частотою не менш 2.4 ГГц;
- ✓ не менше 2 GB оперативної пам'яті;
- ✓ рідкокристалічний монітор з діагоналлю не менше 17 ";
- ✓ 20 Гб вільного місця на жорсткому диску;
- ✓ доступ до мережі Internet;
- ✓ клавіатура;
- ✓ маніпулятор "миша".

Наведені вище технічні характеристики є рекомендованими, тобто при наявності технічних засобів не нижче зазначених, розроблений програмний виріб буде функціонувати відповідно до вимог щодо надійності, швидкості обробки даних і безпеки, висунутими замовником.

1.5.4 Вимоги до інформаційної та програмної сумісності

Для нормального функціонування програми необхідно, щоб програмне забезпечення обчислювальної машини, на якій буде функціонувати веб-орієнтований застосунок, відповідало наступним вимогам:

- ✓ операційна система сімейства Windows (7 та більш сучасні), Linux;
- ✓ веб-браузер Firefox / Google Chrome / Opera / Edge.

Застосунок має бути реалізовано на мові програмування JavaScript у середовищі розробки IntelliJ IDEA.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

В рамках даного проекту була виконана розробка програмного забезпечення плагіна браузера, призначеного для надання інструментарію для роботи з веб-сервісом для повернення кешбеку.

Призначення розробленого застосунку:

- сповіщення про можливість отримання кешбеку в інтернет-магазині;
- спрощення роботи з кешбек-сервісом;
- заощадження коштів і часу покупців при отриманні кешбеку.

Розроблена програма реалізує наступні функції:

- повідомлення про можливість отримання кешбеку на сайтах, що переглядаються браузером;
- активація кешбек-сервісу;
- відображення сайтів для потенційної активації кешбек-сервісу у результатах пошукових систем.

Для досягнення поставленої задачі розроблене програмне забезпечення підтримує виконання таких операцій:

- читання інформації з бази даних кешбек-сервісу;
- внесення і редагування інформації у веб-сховище браузера;
- підтримання різних статусів активності сервісу;
- обмін даними з сайтом кешбек-сервісу у фоновому режимі;
- обробка результатів роботи пошукових систем;
- видача повідомлень через показ спливаючих вікон.

Для виконання перерахованих вище функцій у застосунку реалізовані:

- можливість інтеграції в веб-браузер;

- наявність типової конфігурації, що забезпечує можливість швидкого введення застосунку в експлуатацію;
- програмно-апаратна переносимість.

2.2. Опис застосованих математичних методів

Оскільки особливості предметної галузі розв'язуваної задачі не передбачають застосування математичних методів, при розробці плагіна для роботи з кешбек-сервісом математичні методи не використовувалися.

2.3. Опис використаної архітектури та шаблонів проектування

Мікросервісна архітектура - це підхід, коли єдиний додаток будується як сукупність невеликих, самодостатніх, незалежних, не тісно зв'язаних сервісів, що взаємодіють між собою за допомогою легких механізмів як то HTTP, gRPC, AMQP. Ці сервіси побудовані навколо бізнес-потреб (кожен відповідальний за конкретний процес) та розгортаються незалежно з використанням повністю автоматизованого середовища. Існує абсолютний мінімум централізованого управління цими сервісами. Самі по собі сервіси можуть бути написані на різних мовах і використовувати різні технології зберігання даних.

Мікросервісна архітектура абсолютно протилежна за принципом монолітній, яка є також однією з найрозповсюдженіших для побудови складних програмних додатків. Тож архітектура має такі переваги:

- Кожен сервіс може бути розгорнутий незалежно від інших. Тож якщо ви змінюєте щось в одному з них, ви можете розгорнути ці зміни, не чіпаючи інших мікросервісів, які можуть продовжувати працювати.
- Кожен сервіс може бути масштабований окремо, тобто зміни в одному сервісі повністю ізольовані.
- Неполадки у мікросервісі не повинні зламати весь додаток. Швидше за все, вони не вплинуть суттєво на роботу додатку, особливо великого.

– Усуваються будь-які довгострокові зобов'язання щодо технології. При розробці нового сервісу ви можете вибрати новий стек технологій. Будь-який сервіс у системі можна замінити. Його можна переписати з нуля в межах прийняттого часу та бюджету без необхідності перебудовувати всю систему.

– Мікросервіси, як правило, краще організовані, оскільки кожен мікросервіс має дуже специфічну роботу і не займається роботою інших сервісів. Тож потенційно легші для розуміння, підтримки і тестування.

– Відокремлені сервіси легше перекомпонувати і переналаштувати, щоб виконувати задачі різних додатків.

На відміну від монолітної архітектури, у мікросервісній дуже легко змінювати окремі сервіси, без необхідності робити зміни у інших. У перспективі розробки, мікросервіси можуть залишатися досить простими, адже кожен з них має свою область відповідальності, а у монолітній архітектурі все знаходиться у єдиному середовищі і кодова база програмного артефакту з кожною ітерацією розробки ставатиме все більш складною та заплутанною. Та найголовніше – якщо у одному з мікросервісів станеться якась проблема виконання, ймовірно роботу сервера буде припинено, проте решта сервісів продовжить працювати, і користувач може не помітити, що якась частина функціоналу припинила працювати. У монолітній архітектурі будь-яка поламка означає повне припинення усього програмного продукту.

Тим не менш, мікросервісна архітектура – не ідеал, та має свої недоліки, серед яких:

- Мікросервіси успадковують усі проблеми розподілених систем (складність розподілених транзакцій, остаточна узгодженість, CAP теорема).
- Значні накладні витрати на інфраструктуру, моніторинг і операційні дії.
- Ускладнене налагодження, трасування.
- Незалежність сервісів призводить до дублювання коду (утиліти, робота з БД, об'єкти транспортування даних, тощо).

- Проблеми зі стабільністю мережевого зв'язку між сервісами, мережеві затримки, маршалінг/демаршалінг даних.
- Ускладнене тестування і розгортання.

Серед вищезгаданих проблем однією з найбільших є стабільність мережевого зв'язку. Звісно, при розгортанні додатку у середовищі реального використання, інтернет-мережа має бути налаштована найбільш так, щоб бути стійкою до перебоїв, але проблеми трапляються навіть у найбільш налагодженій мережі. Тож, коли для опрацювання запиту користувача необхідна взаємодія багатьох окремих сервісів, треба запевнитись, що усі необхідні запити між сервісами буде виконано правильно, швидко та без помилок.

Отже мікросервісна архітектура є більш сприятливою для системи, що розроблялась.

Ще один шаблон, що використовувався у розробленій системі – це шаблон Model-View-Controller (MVC). Він використовується для розділення обов'язків у програмі.

Model – представляє модуль, відповідальний за бізнес-логіку програми, тобто данні та їх обробку. У випадку розробленої системи можна вважати, що цей компонент шаблону представлений back-end сервером та базою даних.

View – представляє візуалізацію даних, може мати елементи для управління системою, тобто інтерфейс користувача. Він може перетворювати отриману від контролера інформацію у таку, що потім відобразить користувачу. У розробленій системі цей компонент – front-end сервер, що містить додаток для користувачів. Коли користувач відкриває сторінку додатку, додаток отримує від back-end сервера дані, видозмінить їх та відобразить користувачу у браузері.

Controller – стоїть між відображення та моделлю. Він контролює потік даних від відображення у модель, та навпаки. Він отримує від представлення дані, що потрібно обробити, передає їх моделі для обробки, та повертає відповідну відповідь представленню.

Схема роботи шаблону MVC наведено на рис. 2.2. На ньому видно, яким чином компоненти шаблону взаємодіють одне з одним та як вони відокремлені одне від одного.

2.4. Опис використаних технологій та мов програмування

Застосунок був реалізований на мові програмування JavaScript у середовищі розробки IntelliJ IDEA. В інтерфейсі було використано бібліотеку jQuery.

JavaScript (JS) - динамічна, об'єктно-орієнтована мова програмування, реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, дозволяє коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, управляти браузером, асинхронно обмінюватися даними з сервером, змінювати структуру і зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні сервера (подібно таким мовам програмування, як Java і C#), розробки ігор, стаціонарних і мобільних додатків, сценаріїв в прикладному ПО (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів, і тому подібне.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Крім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну і частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічну і слабку типізацію, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

JavaScript має ряд властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, - функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання (closures) - що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою Сі має наступні корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;
- функції як об'єкти першого класу;
- обробка виключень;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, керуючих логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто, успадкований від С), але спрощений у порівнянні з ним, щоб зробити мову сценаріїв легким для вивчення. Так, наприклад, декларація змінної не містить її типу, властивості теж типів, а декларація функції може стояти в тексті програми після неї.

jQuery є бібліотекою Javascript і служить відмінним інструментарієм з відкритим кодом для створення динамічних веб-додатків. Бібліотека jQuery надає спеціалізовані елементи управління (календарі, вкладки, гармошки, ползункові регулятори і ін.), функції для створення численних спецефектів (загасання, ковзання, анімації), а також операції перетягування.

Крім того, jQuery підтримує технологію Ajax, яка дозволяє звертатися до веб-сервера з браузера користувача, не приводячи до перезавантаження цілої сторінки, а тільки її окремої частини.

Так історично склалося, що браузерери по-різному обробляють код, написаний на Javascript. Відсутність єдиного стандарту призвело до того, що програмістам доводиться писати зайві рядки коду для сумісності з усіма популярними браузерами. Бібліотека jQuery розроблялася таким чином, щоб її

можна було використовувати незалежно від браузера. jQuery надає розробнику базовий набір функцій, який буде працювати у всіх браузерах.

Бібліотека jQuery - це бібліотека Javascript, яка поставляється у вигляді JS-файлу. Завантажити останню версію бібліотеки jQuery можна з офіційного сайту за адресою www.jquery.com. Існує кілька варіантів для завантаження бібліотеки: стандартна або мінімізована. Мінімізована версія (наприклад, `jquery-1.4.2.min.js`) - це версія файлу, в якій вилучені всі розриви рядків і інші символи, щоб зменшити вагу файлу. Можна сказати, що мінімізована версія призначена для браузера, а не для людини. Якщо потрібно заглянути у нутроці бібліотеки і дізнатися, як вона влаштована, необхідно завантажити повну версію (назва файлу без суфікса `min`). У повній версії вже присутні відступи і розриви рядків, тому код бібліотеки можна вивчати без проблем.

IntelliJ IDEA — комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP, JavaScript) від компанії JetBrains. Система поставляється у вигляді обмеженої по функціональності безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition».

IntelliJ IDEA являє собою високотехнологічний комплекс тісно інтегрованих інструментів програмування, що включає інтелектуальний редактор вихідних текстів з розвиненими засобами автоматизації, потужні інструменти рефакторинга коду, вбудовану підтримку технологій J2EE, механізми інтеграції з середовищем тестування Ant/JUnit і системами управління версіями, унікальний інструмент оптимізації та перевірки коду Code Inspection, а також інноваційний візуальний конструктор графічних інтерфейсів.

IntelliJ IDEA надає розробнику такі інструменти:

- розумне автодоповнення, інструменти для аналізу якості коду, зручна навігація, розширені рефакторингом і форматування для Java, Groovy, Scala, Clojure і Erlang;
- професійний набір інструментів для розробки Android-додатків;

- підтримка JavaFX 2.0, інтеграція з SceneBuilder; Дизайнер інтерфейсу для Swing;
- інтеграція з автоматизованими інструментами побудови та управління проектом, включаючи Maven, Gradle, Ant та іншими;
- інструменти для тестування з підтримкою JUnit, TestNG, Spock, ScalaTest і spec2;
- інтеграція з системами управління версіями, включаючи Git, Subversion, Mercurial і CSV.

Унікальні можливості JetBrains IntelliJ IDEA позбавляють програміста від вантажу рутинної роботи, допомагають своєчасно усунути помилки і підвищити якість коду, значно збільшуючи продуктивність розробника.

2.5. Опис структури програми та алгоритмів її функціонування

Для створення плагіну для роботи з кешбек- сервісом був використаний шаблон проектування «Делегування».

Делегування (англ. Delegation) - основний шаблон проектування, в якому об'єкт зовні виражає деяку поведінку, але в реальності передає відповідальність за виконання цієї поведінки пов'язаному об'єкту (рис. 2.1). Шаблон делегування є фундаментальною абстракцією, на основі якої реалізовано інші шаблони - композиція (також звана агрегацією), домішки (mixins) і аспекти (aspects).

Делегат надає механізм, який забезпечує проектування програмного забезпечення з мінімальною взаємозалежністю між компонентами. Делегування застосовується в разі необхідності багаторазового використання характеристик поведінки і розширення класу.

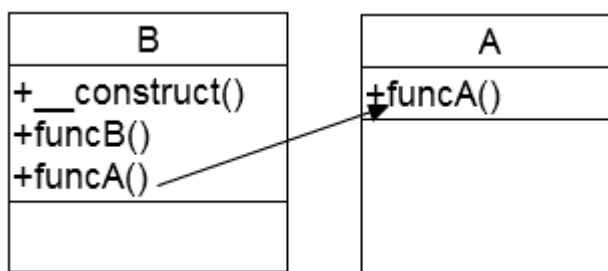


Рис. 2.1. Класова модель делегування

Делегування застосовується у ситуації, коли один і той же суб'єкт може грати комбінації ролей:

- спадкування забезпечує статичний характер відносин (спадкування). Тип об'єкта не змінюється з часом;
- модель вимагає динамічного характеру відносин (role-played-by). Об'єкт може делегувати поведінку різних об'єктів в різні моменти часу.

Суть використання даного шаблону полягає у тому, що об'єкт делегує відповідальність за виконання ролі деякого об'єкту.

Переваги делегування:

- необхідність подання об'єкта різними підкласами в різні моменти часу;
- проблема приховування успадкованих методів;
- поведінка, успадковане від суперкласу, не може бути з легкістю змінено.

Недоліки:

- реалізація делегування вимагає більше коду.

Для вирішення задачі створення плагіну, який взаємодіє із кешбек-сервісом, було використано модель делегування подій. Кожен об'єкт-джерело події посилає повідомлення об'єктам-слухачам. Об'єкт-джерело, як правило, не вирішує, що робити з цією подією, а делегує відповідальність об'єктам-слухачам. У ролі об'єкт-джерела виступає кешбек-сервіс, у якості об'єкта-слухача –

розроблений плагін. Фрагмент діаграми класів, яка ілюструє взаємодію кешбек-сервісу та плагіна, показано на рис. 2.2.

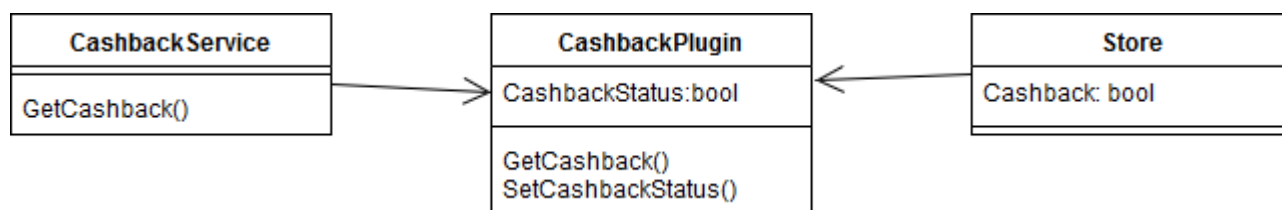


Рис. 2.2. Реалізація делегування у системі «кешбек-сервіс – плагін»

Для зберігання даних про активовані кешбеки використовується документна база даних, що є частиною браузера. Зберігання даних безпосередньо в браузері має безліч переваг, основне з яких це швидкий і незалежний від мережі доступ до такої бази даних. На даний момент для цього є три активних методи:

- куки;
- веб-сховище (DOM-сховище);
- IndexedDB.

Для зберігання даних плагіну кешбеку був обраний варіант веб-сховища (Інтернет-сховища).

Інтернет-сховище або DOM-сховище - це програмні методи і протоколи веб-застосунків, які використовуються для зберігання даних в веб-браузері. Інтернет-сховище являє собою постійне сховище даних, схоже на куки, але зі значно розширеною ємністю і без зберігання інформації в заголовку запиту HTTP. Існують два основних типи веб-сховища: локальне сховище (localStorage) і сесійне сховище (sessionStorage), провідні себе аналогічно постійним і сесійним кукам відповідно.

Інтернет-сховище підтримує набагато більше місця на диску в порівнянні з куки, якому доступно всього 4 Кбайта, що приблизно в 1000 разів менше ніж у веб-сховища (5 Мбайт на домен в Mozilla Firefox, Google Chrome, Opera, і Edge). На відміну від куки, які можуть бути доступні як на сервері, так і на стороні

клієнта, веб-сховище потрапляє виключно до компетенції сценаріїв (скриптів) на стороні клієнта. Дані інтернет-сховища не передаються на сервер при кожному запиті HTTP, і веб-сервер не може безпосередньо записати в інтернет-сховище.

Інтернет-сховище пропонує дві різних області: локальна пам'ять і сесійне сховище, які відрізняються за своїми масштабами і часу життя. Дані розміщуються в окреме для кожного домена локальне сховище (воно є для всіх скриптів з домену, який спочатку додав дані) і зберігаються після закриття браузера. Сесія зберігається за принципом одна сторінка-одне вікно і обмежується життям даного вікна, тобто для кожного відкритого вікна створюється нова сесія, яка припиняє своє існування при закритті вікна і не залежить від домену, що відкрив її. Збереження сесії призначене для надання окремих екземплярів одного і того ж веб-застосунку для роботи в різних вікнах, не заважаючи один одному.

Інтернет-сховище в даний час надає програмний інтерфейс краще, ніж куки. Інтерфейс являє собою асоціативний масив моделі даних, де ключі і значення є рядками.

Переваги веб-сховища:

- Пропонує більш простий і інтуїтивний інтерфейс зберігання даних.
- Безпечніше для зберігання даних на клієнті.
- Дозволяє зберігати більше даних, ніж куки.

Недоліки веб-сховища:

- Дозволяє зберігати тільки рядки.

Локальна пам'ять і сесійне сховище є різновидами Web Storage API, спеціального API для зберігання даних в браузері в форматі ключ-значення. Браузери надають дані API для використання для взаємодії з DOM. Сучасні фреймворки JavaScript використовують API DOM, щоб повідомити браузеру, що відобразатиметься на сторінці.

DOM - це відображення документа у вигляді множини об'єктів, з якими можна взаємодіяти з використанням JavaScript. Таке кількість місць називають деревом об'єктів DOM. Кожен елемент DOM дерева називається нодом (вузол).

Таким чином, з допомогою такої структури, можна повністю взаємодіяти з кожним елементом HTML сторінки.

У загальному випадку, кожен HTML-тег має свій, спеціальний JavaScript-об'єкт, який входить в DOM дерево. Взаємодіючи з цим об'єктом, можна фактично змінити HTML-код або взаємодію з ним.

Доступ до DOM надається з допомогою документа глобального об'єкта. Об'єкт документа репрезентує вхідну точку в структуру DOM (рис.2.3).

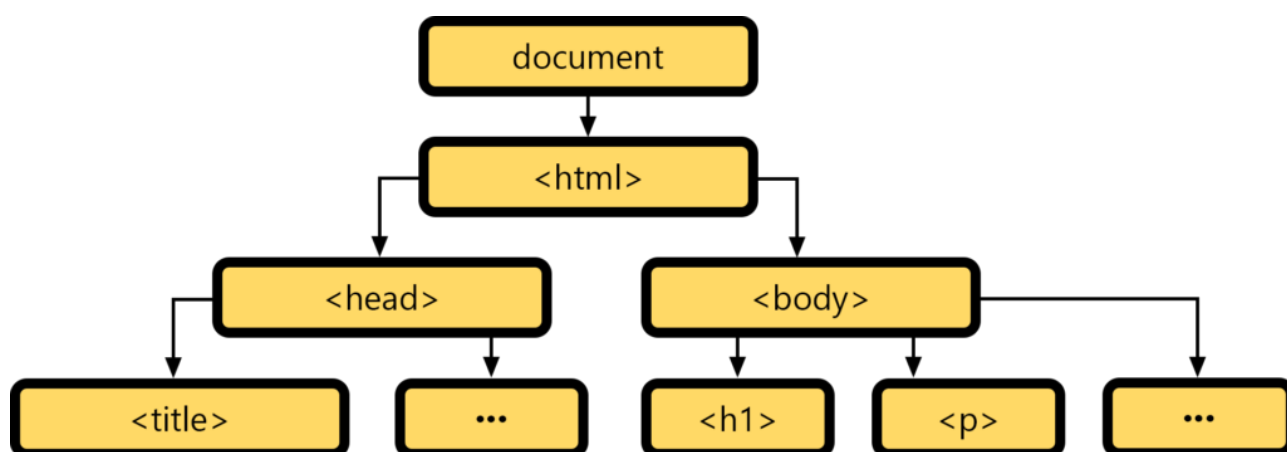


Рис. 2.3. Фрагмент базової структури DOM-дерева

Знайти елемент в DOM дереві не завжди є остаточною метою при роботі з ним. Часто, як правило необхідно знайти дочірні, сусідні або батьківський елементи щодо якогось елементу. Для такої навігації по DOM дереву існують спеціальні властивості для навігації.

При навігації по DOM дереву можна отримати доступ як до нодів будь-якого типу (element, text, comment і т.д.), так і виключно до нодів з типом element, що найчастіше і буває необхідно.

Об'єкт window, крім того, що є глобальним об'єктом в JavaScript, містить в собі безліч властивостей, методів і об'єктів для роботи з поточною сторінкою і вікном браузера (рис.2.4).

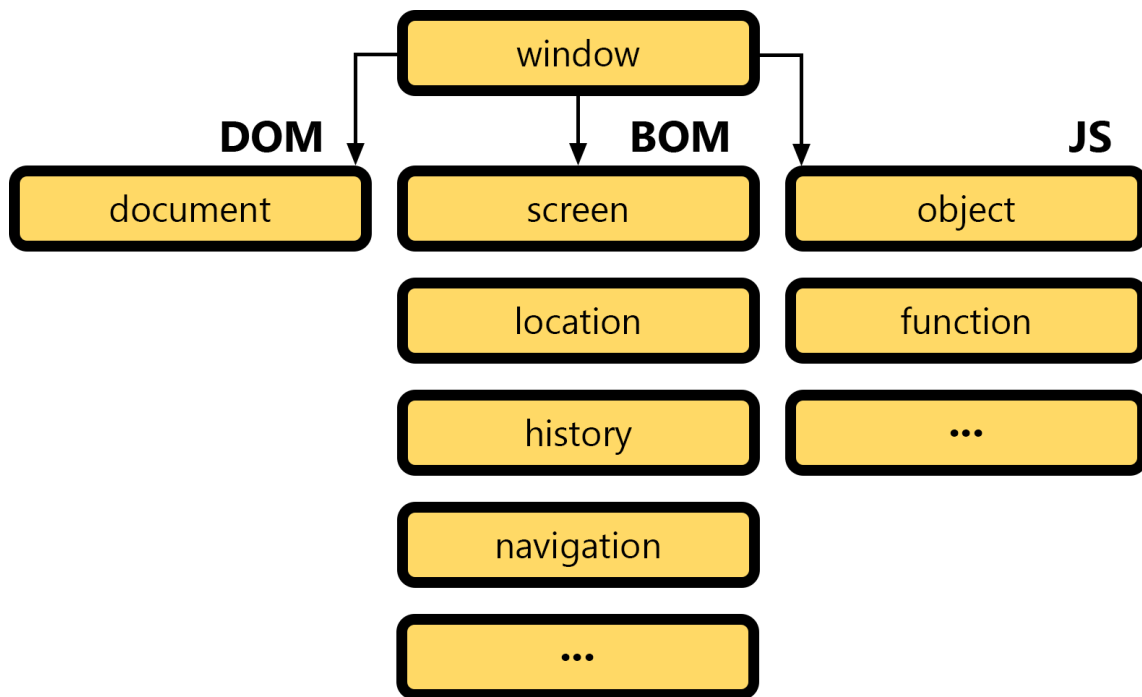


Рис. 2.4. Схема зв'язку об'єктних моделей документа та браузера

Як видно зі схеми на рис. 2.4, об'єкт `window` служить сполучною ланкою для взаємодії JavaScript з браузером. Об'єктна модель документа (Document Object Model - DOM), щоб виконувати завдання зі структурою документа, об'єктна модель браузера (Browser Object Model - BOM) - це збірне поняття, яке об'єднує безліч об'єктів, що містять в собі властивості і методи для роботи з браузером. В JavaScript не існує вбудованих властивостей і методів для роботи DOM і BOM. Іншими словами, сам по собі JavaScript не вміє працювати ні з HTML сторінкою, ні з браузером, це можливо лише завдяки глобальному об'єкту `window`, який надає набір необхідних властивостей і методів, завдяки яким JavaScript може всебічно взаємодіяти з браузером.

Для роботи з веб-сховищем (Web Storage), яке розроблений плагін використовує для локального зберігання даних в браузері, існують два об'єкти, які належать об'єкту `window`:

- `localStorage` - дозволяє зберігати дані локально без обмежень за часом зберігання і можуть бути видалені тільки за допомогою JavaScript;
- `sessionStorage` - дозволяє зберігати дані локально на протязі однієї сесії (до тих пір, поки не буде закрита вкладка браузера).

Методи для роботи з localStorage і sessionStorage однакові. Нижче вони перелічені на прикладі localStorage:

- localStorage.setItem () - додавання значення в сховище;
- localStorage.getItem () - отримання значення з сховища по ключу;
- localStorage.removeItem () - видалити значення по ключу;
- localStorage.clear () - очистити сховище.

Приведений нижче фрагмент коду ілюструє використання перелічених методів:

```
localStorage.setItem('site', 'jsexpert.net');  
// додали дані, які можна отримати по ключу site  
let site = localStorage.getItem('site');  
console.log(site); // 'jsexpert.net'  
localStorage.removeItem('site');  
localStorage.clear();
```

Дані встановлюються в форматі ключ значення.

Якщо при додаванні даних в сховищі значення з таким ключем вже буде існувати, то значення буде перезаписано. Зберігання однакових ключів неприпустимо.

Так як localStorage і sessionStorage є звичайними об'єктами, то для додавання, отримання або видалення значень, можна використовувати синтаксис звичний для роботи з об'єктами:

```
localStorage['site'] = 'jsexpert.net' // встановлення  
значення  
localStorage['site']; // отримання значення  
delete localStorage['site'] // видалення значення
```

2.6. Обґрунтування та організація вхідних та вихідних даних програми

Програмне забезпечення отримує вхідні дані шляхом обміну даними з сайтом кешбек-сервісу та обробки результатів роботи пошукових систем.

Вхідні дані:

- списки адрес інтернет-магазинів, з якими працює кешбек-сервіс;

- вміст адресного рядку браузера;
- вміст сторінок результатів пошуку пошукових систем;
- посилання на отримання кешбеку зі сторінки інтернет-магазину.

Вихідні дані:

- значення статусу активності сервіса;
- посилання на отримання кешбеку;
- записи до веб-сховища браузера;
- повідомлення у спливаючих вікнах.

2.7. Опис роботи розробленого програмного продукту

2.7.1. Використані технічні засоби

Для серверних технічних засобів рекомендована конфігурація, що забезпечує цілодобову роботу програми з резервуванням даних:

- ✓ процесор класу Intel ® Xeon з тактовою частотою 2.4GHz;
- ✓ шина даних - 1066MHz,
- ✓ кеш другого рівня - 2048 КБ;
- ✓ оперативна пам'ять 2 x DIMM DDR2-800 1024 Мб;
- ✓ жорсткі диски 3x 250 Гб SATA 2 16 Мб буфер, 7200 RPM;
- ✓ рідкокристалічний монітор з діагоналлю не менше 17 ";
- ✓ доступ до мережі Internet;
- ✓ клавіатура;
- ✓ маніпулятор "миша".

Наведені вище технічні характеристики є рекомендованими, тобто при наявності технічних засобів не нижче зазначених, розроблений програмний виріб буде функціонувати відповідно до вимог щодо надійності, швидкості обробки даних і безпеки, висунутими замовником.

2.7.2. Використані програмні засоби

Проект реалізований на мові програмування JavaScript, тому для коректної роботи плагіна необхідний довільний веб-браузер з підтримкою технології JavaScript.

Необхідні програмні засоби:

- ✓ ОС сімейства Windows 7 та більш нові, Linux;
- ✓ веб-браузер Firefox / Google Chrome / Opera / Edge.

2.7.3. Виклик та завантаження програми

Розроблений плагін для роботи з кешбек-сервісом використовується для інтеграції з кешбек-порталом toprecash.net. Після першого підключення плагіна до обраного браузера для подальшого використання застосунку достатньо запустити браузер.

2.7.4. Опис інтерфейсу користувача

Для початку роботи з плагіном необхідно увійти в особистий розділ користувача, клікнувши на кнопку «Авторизуватися» на головній сторінці кешбек-сервісу (рис. 2.5). Для авторизації користувача необхідно ввести емейл і пароль у відповідні поля вводу центрального блоку сторінки.

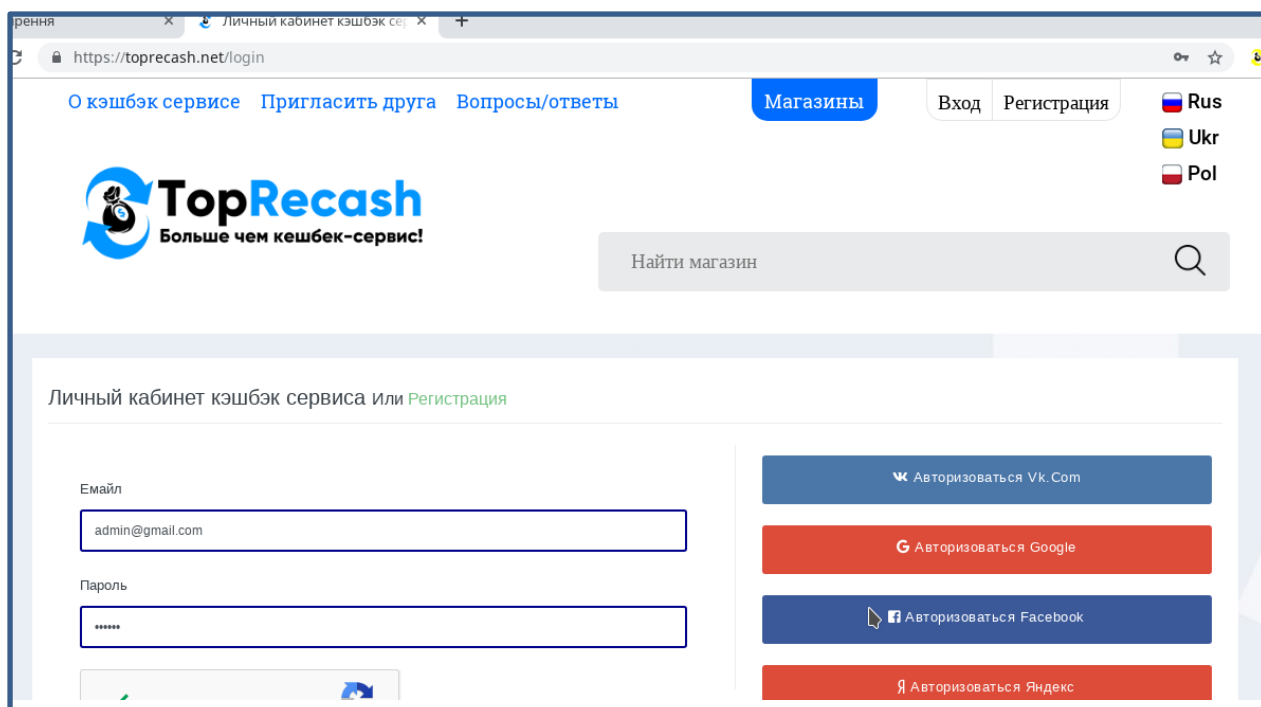


Рис. 2.5. Головна сторінка кешбек-сервісу

Нижче під блоком авторизації на головній сторінці розміщений список нових магазинів, з якими співпрацює сервіс, та їх пропозиціями щодо розміру кешбеку (рис. 2.6).

Після успішної авторизації клієнт переходить до сторінки «Особистий кабінет», яка містить такі розділи: «Магазини», «Мої покупки» та «Запросити друга». У верхній частині сторінки знаходиться рядок поточного стану балансу клієнта.

На сторінці «Магазини» розміщується перелік усіх магазинів, з якими співпрацює даний сервіс, включаючи їх логотип, назву, обсяг кешбеку, та посилання для переходу у обраний магазин.

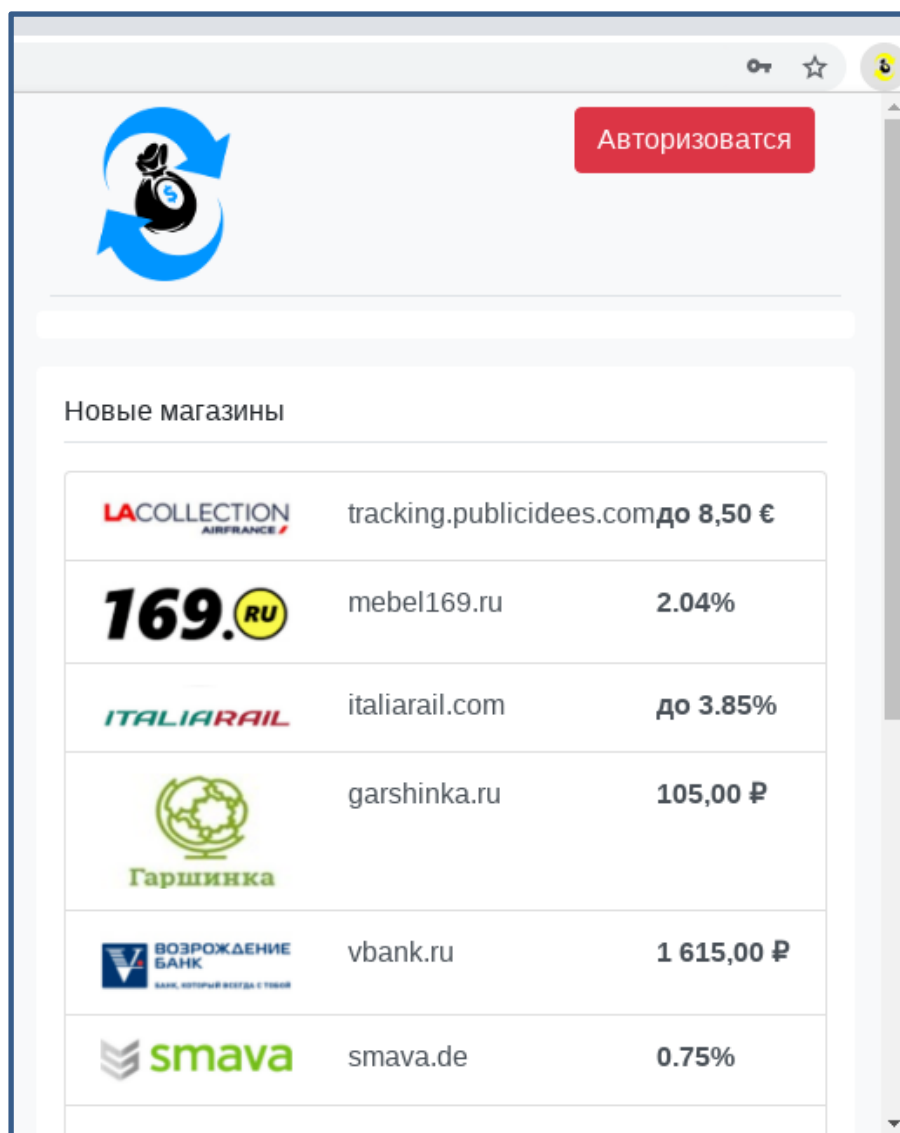


Рис. 2.6. Нові магазини

На сторінці «Мої покупки» розташований список замовлень користувача, відсортований по даті здійснення покупки. Опис замовлення складається з назви магазину, у якому воно було зроблене, номера замовлення, суми кешбеку, та статусу замовлення (рис. 2.7).

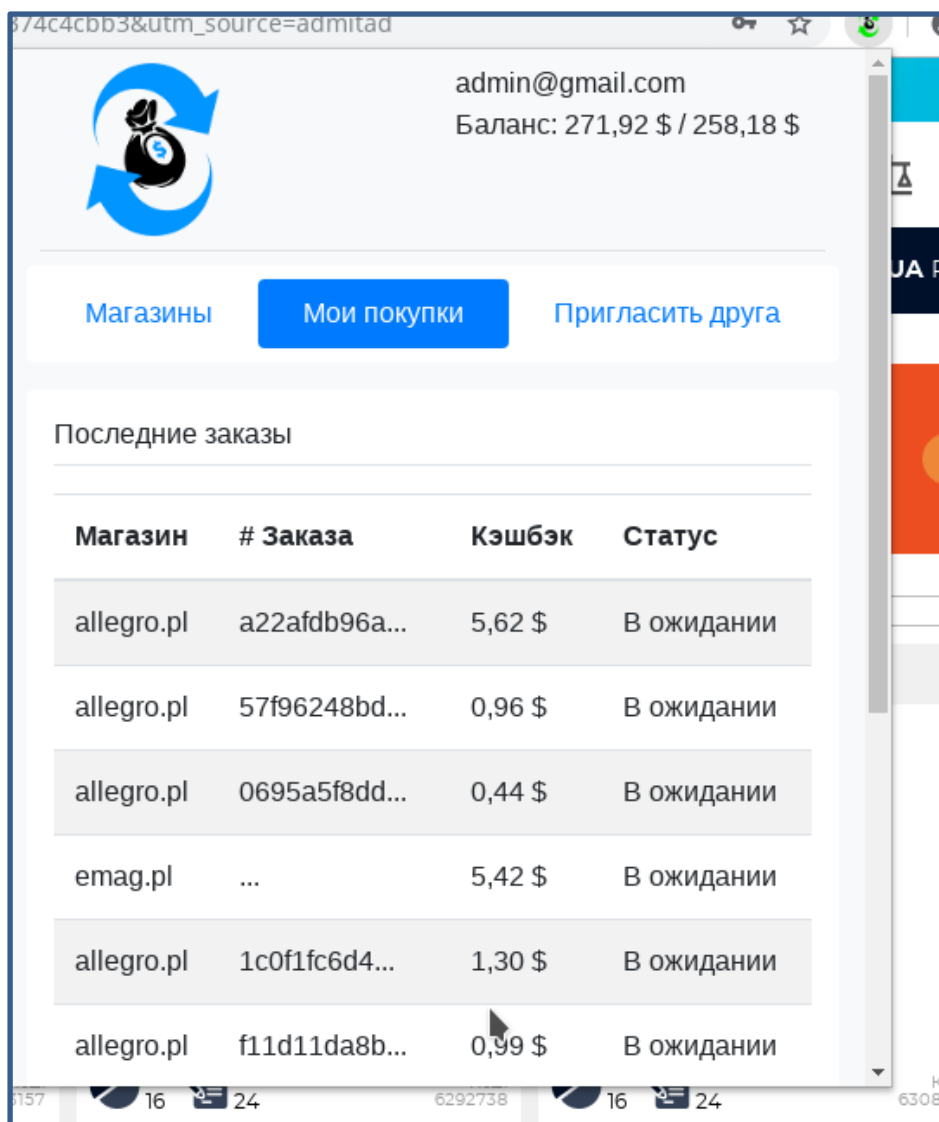


Рис. 2.7. Останні замовлення

На сторінці «Запросити друга» розміщено інформацію про систему заохочення за запрошення нових клієнтів сервісу, та відображено поточний стан реферального рахунку клієнта (рис. 2.8).

Для того, щоб використовувати функції плагіна, користувач повинен залишатися авторизованим у сервісі. За виконанням даної умови, під час відвідання сторінок інтернет-магазинів, плагін буде працювати у фоновому режимі, надаючи можливість активувати кешбек та зробити замовлення на сайті без додаткових дій або переходів на сторінку сервісу.

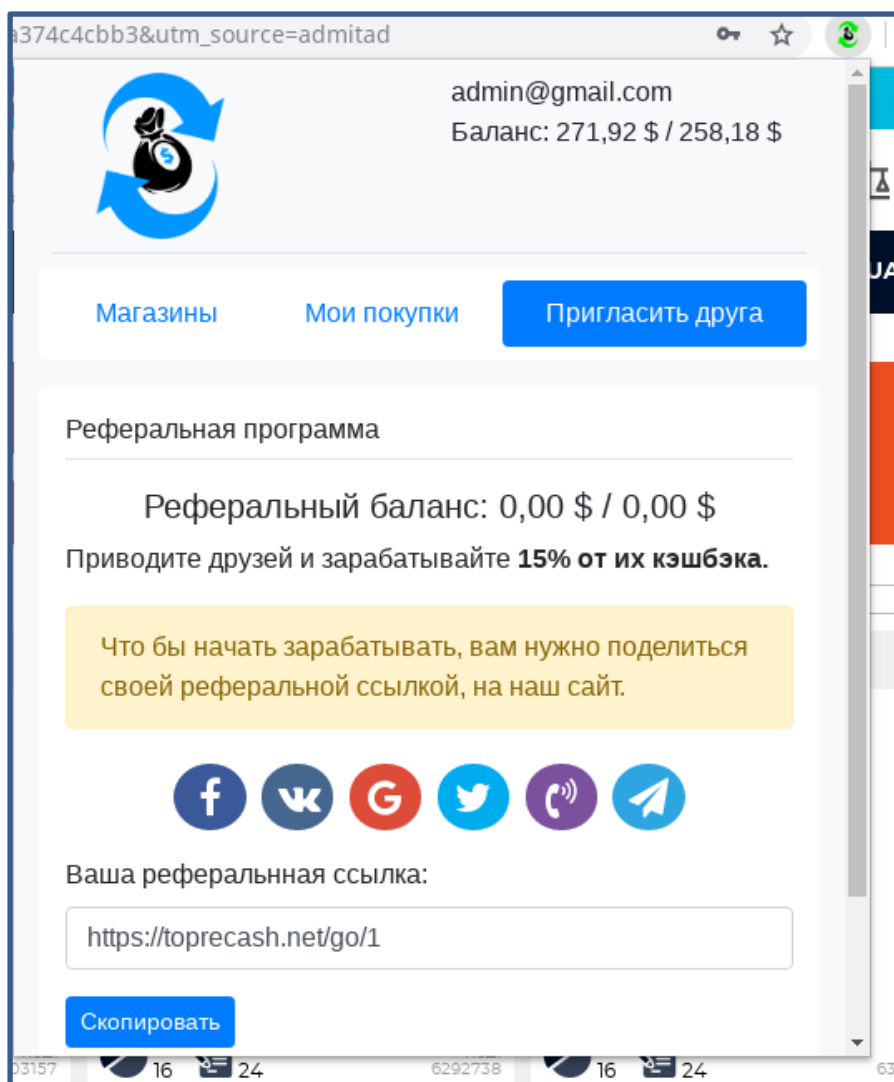


Рис. 2.8. Сторінка «Запросити друга»

Якщо інтернет-магазин, на сторінці якого знаходиться користувач, підтримує контракт на повернення кешбеку із кешбек-сервісом, плагін проінформує про це через спливаючі вікна з відповідним повідомленням та кнопкою для активації кешбеку (рис. 2.9). Після підтвердження активації, клієнт може робити замовлення у магазині, а посилання на отримання кешбеку за них будуть автоматично оброблені плагіном.

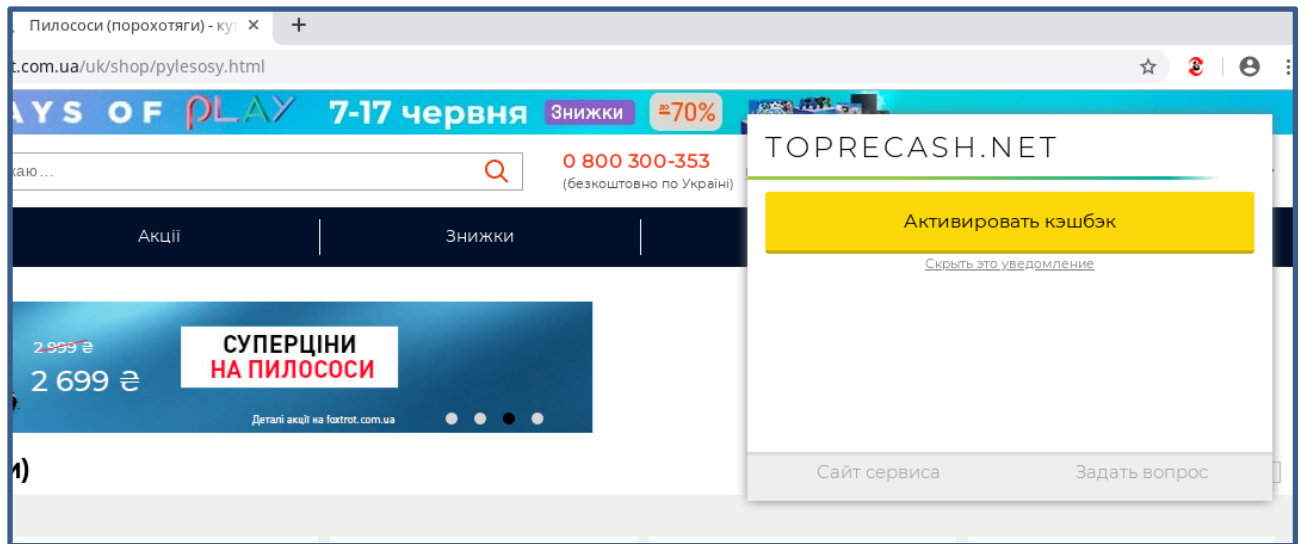


Рис. 2.9. Активація кешбеку на сайті

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 900;
2. коефіцієнт складності програми – 1,6;
3. коефіцієнт корекції програми в ході її розробки – 0,05;
4. годинна заробітна плата програміста – 65 грн/год;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,2;
7. вартість машино-години ЕОМ – 14 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

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

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 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_d - витрати праці на підготовку документації.

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

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p),$$

де q - передбачуване число операторів (900);

C - коефіцієнт складності програми (1,6);

p - коефіцієнт корекції програми в ході її розробки (0,05).

Звідси умовне число операторів в програмі:

$$Q = 1,6 \cdot 900 \cdot (1 + 0,05) = 1512$$

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

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин,}$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 3 до 5 років він складає 1,2.

Прийmemo збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ($B = 1,2$). З урахуванням коефіцієнта кваліфікації $k = 1,2$, отримуємо витрати праці на вивчення опису завдання:

$$t_u = (1512 \cdot 1,2) / (75 \cdot 1,2) = 20,16 \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин, (3.2)}$$

де Q – умовне число операторів програми;

k – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.2), отримаємо:

$$t_a = 1512 / (20 \cdot 1,2) = 63 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин.}$$

$$t_n = 1512 / (25 \cdot 1,2) = 50,4 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин.}$$

$$t_{oml} = 1512 / (5 \cdot 1,2) = 252 \text{ чел.-ч.}$$

- за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ людино-годин.}$$

$$t_{oml}^k = 1,5 \cdot 252 = 378 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ людино-годин,}$$

де $t_{\partial p}$ - трудомісткість підготовки матеріалів і рукопису:

$$t_{\partial p} = \frac{Q}{(15..20) \cdot k}, \text{ людино-годин,}$$

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

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \text{ людино-годин.}$$

Підставляючи відповідні значення, отримаємо:

$$t_{\partial p} = 1512 / (18 \cdot 1,2) = 70 \text{ людино-годин.}$$

$$t_{\partial o} = 0,75 \cdot 70 = 52,5 \text{ людино-годин.}$$

$$t_{\partial} = 70 + 52,5 = 122,5 \text{ людино-годин.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 20,16 + 63 + 50,4 + 252 + 122,5 = 558,06 \text{ людино-годин.}$$

3.2. Рахунок витрат на створення програми

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

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн.}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн,}$$

де: t - загальна трудомісткість, людино-годин;

$C_{ПР}$ - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 65 грн / год, отримуємо:

$$Z_{ЗП} = 558,06 \cdot 65 = 36\,273,9 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн, (3.3)}$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (14 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$З_{mv} = 252 \cdot 14 = 3528 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{ПО} = 36273,9 + 3528 = 39\,801,9 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p} \text{ міс.}$$

де B_k - число виконавців (дорівнює 1);

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

Звідси витрати на створення програмного продукту:

$$T = 558,06 / 1 \cdot 176 \approx 3 \text{ міс.}$$

Висновок

Програмне забезпечення призначене для забезпечення функціонування плагіну браузера для роботи з кешбек-сервісом. Вартість даного програмного забезпечення становить 40 тис. грн. і не вимагає додаткових витрат при впровадженні та експлуатації програми. Очікуваний час розробки становить 3 місяці. Цей термін пов'язаний зі значним числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження програми і підготовку документації.

ВИСНОВКИ

В даній кваліфікаційній роботі було розроблено плагін браузера, призначений для надання інструментарію для роботи з веб-сервісом для повернення кешбеку.

Це програмне забезпечення призначене для спрощення роботи з кешбек-сервісом та заощадження коштів і часу покупців при отриманні кешбеку. Практичне призначення даного застосунку полягає в забезпеченні роботи розширень кешбек-сервісів, і як наслідок – збільшенні ефективності роботи інтернет-магазину через заохочення покупців.

Під час виконання даного проекту були виконані наступні задачі:

- вивчено предметну галузь розв'язуваної задачі;
- створено алгоритм для реалізації поставленого завдання;
- створено програмний код зпроектованого застосунку.

Розроблене програмне забезпечення дозволяє:

- отримувати повідомлення про можливість отримання кешбеку на сайтах, що переглядаються браузером;
- активацію кешбек-сервісу;
- відображення сайтів для потенційної активації кешбек-сервісу у результатах пошукових систем.

Програма реалізована на мові програмування JavaScript у середовищі розробки IntelliJ IDEA.

Також у кваліфікаційній роботі було визначено трудомісткість розробленого плагіна 558 люд-год, проведений підрахунок вартості роботи по створенню програми 39 801 грн. та розраховано час на його створення 3 місяця.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання : (ГОСТ 7.1-2003, IDТ) : ДСТУ ГОСТ 7.1:2006. – Чинний з 2007–07–01. – К. : Держспоживстандарт України, 2007. – 47 с. – (Система стандартів з інформації, бібліотечної та видавничої справи) (Національний стандарт України).
2. Бибо Б. jQuery. Подробное руководство по продвинутому JavaScript / Б. Бибо, И. Кац. — Спб.: Символ-Плюс, 2009. — 384 с.
3. Вертайм К. Цифровой маркетинг. Как увеличить продажи с помощью социальных сетей, блогов, вики-ресурсов, мобильных телефонов и других современных технологий / К. Вертайм, Я. Фенвик. — Альпина Паблишер, 2010. - 384 с.
4. Гарсиа-Молина Г. Системы баз данных. Полный курс / Г. Гарсиа-Молина, Дж. Ульман, Дж. Уидом. — М.: Вильямс, 2003. – 1088 с.
5. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення : ДСТУ 3008-95. – Чинний від 1996–01–01. – К. : Держстандарт України, 1996. – 39 с.
6. ДСТУ 2394-94 Інформація та документація. Комплектування фонду, бібліографічний опис, аналіз документів. Терміни та визначення. – Чинний від 01.01.1995. — Київ : Держстандарт України, 1994. – 88 с.
7. Кантор И. Современный учебник JavaScript. [электронный ресурс]. Режим доступа: <http://learn.javascript.ru/>
8. Мальчук Е.В. HTML и CSS. Самоучитель / Е. В. Мальчук – М.: Вильямс, 2008. – 416 с.
9. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності «Комп'ютерні системи» / О.Г. Вагонова, О.Б. Нікітіна, Н.Н. Романюк; М-во освіти і науки України, ДВНЗ «Нац. гірн. ун-т». – Д.: НГУ, 2013. – 11 с.

10. Методичні рекомендації до виконання кваліфікаційних робіт бакалаврів напряму підготовки 6.050101 «Комп'ютерні науки / І.М. Удовик, Л.М. Коротенко, О.С. Шевцова; Нац. гірн. ун-т. – Д : НТУ «Дніпровська політехніка», 2018. – 65 с.

11. Пьюривал С. Основы разработки веб-приложений / С. Пьюривал. — СПб.: Питер, 2015. — 272 с.

12. Скорочення слів в українській мові у бібліографічному описі. Загальні правила та вимоги : ДСТУ 3582-97. – Чинний від 1998-07-01. – К. : Держстандарт України, 1998. – 24 с. – (Державний стандарт України).

13. Скотт Б. Проектирование веб-интерфейсов / Б. Скотт, Т. Нейл. — СПб.: Символ-Плюс, 2010. — 352 с.

14. Столлингс В. Передача данных/ В. Столлингс. - СПб.: Питер, 2004. – 750 с.

15. Флэнаган Д. Java. Справочник, 4-е изд.; пер. с англ. / Д. Флэнаган – СПб.: Символ-Плюс, 2004. - 1040 с.: ил.

16. Хеник Б. HTML и CSS. Путь к совершенству / Б. Хеник – СПб.: Питер, 2011. – 336 с.

17. Шаллоуей А. Шаблоны проектирования. Новый подход к объектно-ориентированному анализу и проектированию / А. Шаллоуей, Дж. Р. Тротт. — М.: Вильямс, 2002. — 288 с.

18. Шкрыль А. А. Програмуємо для web-сайта / А.А. Шкрыль – М.: Диалектика, 2006. – 368с: ил.

19. Эккель Б. Философия Java. Библиотека программиста. 4-е изд.; пер. с англ. / Б. Эккель. – СПб.: Питер, 2009. – 640 с.: ил.

20. Середня заробітна плата програміста у Дніпрі станом на початок 2021 року. [Електронний ресурс] - Режим доступу: <https://dou.ua/lenta/articles/salary-report-devs-june-2020/>

КОД ПРОГРАМИ

Background_mf.js // головний клас програми (для браузера Mozilla Firefox)

```
;(function => {
  class Megatron {
    constructor() {
      const self = this
      self.mainHost = 'http://toprecash.net'
      self.authUrl = self.mainHost + '/cabinet/orders/'
      self.timeActiveOffer = 60 * 60 * 1000 // 1 hour
      self.timeHideOffer = 10 * 60 * 1000 // 10 min
      self.notifyTimeout = 10 * 1000 // 10 s
      self.popupIcon = 'default'
      self.currentOffer = null
      self.isAddTargetUrls = false
      self.db = {}
      self.messenger()
      chrome.storage.local.get(null, db => {
        self.db = db
      })
    }

    getDataUrl() {
      const self = this
      return self.mainHost + '/local/dirix.megatron/api/browser.extension/data/get/?uid=' + (self.db.uid
|| '') + '&token=' + (self.db.token || '')
    }

    getDataOfServer(c) {
      const self = this
      axios
        .get(self.getDataUrl())
        .then(d => {
          try {
            if (d.data.status === 'ok') {
              d.data.hiddenOffers = {}
              d.data.activeOffers = {}
              d.data.notifyOffers = {}
              for (let i in d.data.offers) {
                //d.data.offers[i].offer_logo = d.data.offers[i].offer_logo.replace(/^http/,
'https:').replace(':80', '')
                d.data.offers[i].offer_logo = d.data.offers[i].offer_logo.replace(':80', '')
                d.data.offers[i].host = new URI(d.data.offers[i].offer_site_url).host().replace(/^www\./, '')
                d.data.offers[i].domain = new URI(d.data.offers[i].offer_site_url).domain(true)
                d.data.offers[i].tld = new URI(d.data.offers[i].offer_site_url).tld()
              }
            }
            chrome.storage.local.set(d.data, () => {
              chrome.storage.local.get(null, db => {
                self.db = db
                c(null, db)
                chrome.tabs.query({}, tabs => {
                  for (let i = 0, n = tabs.length; i < n; i++) {
                    if (self.checkIsOffer(new URI(tabs[i].url).host().replace(/^www\./, ''))) {
```

```

        chrome.tabs.executeScript(tabs[i].id, { code:
`;(function(){if(document.querySelector('.fooBarClassName-overlay'))
document.querySelector('.fooBarClassName-overlay').remove()})();` })
    }
    }
    self.checkTab()
})
if (self.isAddTargetUrls === false) {
    self.isAddTargetUrls = true
    const urls = []
    for (let i in db.offers) urls.push(db.offers[i].host)
    chrome.webRequest.onBeforeRedirect.addListener(
        function(detail) {
            if (new RegExp('^' + self.mainHost).test(detail.originUrl)) {
                let testUrl = urls.find(e => new RegExp('^https?:/(www\\.)?([\\w-]{1,}\\.)?' +
e).test(detail.redirectUrl))
                if (testUrl) {
                    //self.db.activeOffers[self.checkIsOffer(new
URI(detail.redirectUrl).host().replace(/^www\\./, ""))] = Date.now()
                    //chrome.storage.local.set({ activeOffers: self.db.activeOffers })
                    let offerId = self.checkIsOffer(new
URI(detail.redirectUrl).host().replace(/^www\\./, ""));
                    self.db.activeOffers[offerId] = Date.now();
                    self.db.notifyOffers[offerId] = 0;
                    chrome.storage.local.set({ activeOffers: self.db.activeOffers });
                    chrome.storage.local.set({ notifyOffers:
self.db.notifyOffers });
                }
            }
        },
        {
            urls: ['<all_urls>'],
            types: ['main_frame']
        },
        ['responseHeaders']
    )
}
})
} else {
    chrome.storage.local.clear() => {
        self.currentOffer = null
        c(null, { msg: chrome.i18n.getMessage('chrome_popup_c_auth_text') })
    }
}
} catch (e) {
    chrome.storage.local.clear() => {
        self.currentOffer = null
        c(chrome.i18n.getMessage('chrome_popup_error'))
        console.log(e)
    }
}
}.catch(e => {
    chrome.storage.local.clear() => {
        self.currentOffer = null
        c(chrome.i18n.getMessage('chrome_popup_error'))
    }
}

```

```

        console.log(e)
    })
})
}

setIcon(type) {
    const self = this
    if (!type) {
        self.popupIcon = 'default'
        return chrome.browserAction.setIcon({ path: 'icons/icon_19x19.png' })
    } else if (type === 'red') {
        self.popupIcon = 'red'
        chrome.browserAction.setIcon({ path: 'icons/icon_19x19_action.png' })
    } else if (type === 'green') {
        self.popupIcon = 'green'
        chrome.browserAction.setIcon({ path: 'icons/icon_19x19_green.png' })
    } else {
        self.popupIcon = 'default'
        chrome.browserAction.setIcon({ path: 'icons/icon_19x19.png' })
    }
}

checkIsOffer(host) {
    const self = this
    self.currentOffer = null
    if (!host || !self.db || !self.db.offers) return false
    for (let i in self.db.offers) {
        if (host === self.db.offers[i].host) {
            self.currentOffer = i
            return i
        }
    }
    let d = host.match(/.\/g)
    let tld = new URI('http://' + host).tld()
    if (d && d.length >= (tld.indexOf('.') === -1 ? 2 : 3)) host = host.replace(/^\w+\./, "")
    if (!host || !self.db || !self.db.offers) return false
    for (let i in self.db.offers) {
        if (host === self.db.offers[i].host) {
            self.currentOffer = i
            return i
        }
    }
    return false
}

checkTab() {
    const self = this
    chrome.tabs.query({ active: true }, function(currentTab) {
        if (!currentTab || !currentTab.length) return
        currentTab = currentTab[0]
        let host = new URI(currentTab.url).host().replace(/^\www\./, "")
        if (/^google\./.test(host) || host === 'yandex.ru') {
            self.setIcon('default')
        }
        // if (!currentTab || currentTab.status !== 'complete') return
        if (!currentTab) return
        if (new RegExp('^' + self.authUrl).test(currentTab.url)) {

```



```

chrome.tabs.executeScript(
  currentTab.id,
  {
    code: `(function(){ if (document.querySelector('[name="Login"]') &&
!document.querySelector('[name="Login"]').getAttribute('plugin-event')) {
      document.querySelector('[name="Login"]').setAttribute('plugin-event', '1')
      document.querySelector('[name="Login"]').addEventListener('click', function(){
        chrome.runtime.sendMessage({type: 'auth'})
      })
    }})`;
  },
  function() {
    if (chrome.runtime.lastError) console.log('executeScript', chrome.runtime.lastError)
  }
)
return
}

if (!self.db.offers) return

if (host === 'yandex.ru') {
  chrome.tabs.executeScript(currentTab.id, { file: 'yandex.js' }, function() {
    if (chrome.runtime.lastError) console.log('executeScript', chrome.runtime.lastError)
  })
  return
}
if (/^google\./.test(host)) {
  chrome.tabs.executeScript(currentTab.id, { file: 'google.js' }, function() {
    if (chrome.runtime.lastError) console.log('executeScript', chrome.runtime.lastError)
  })
  return
}

let keyOffer = self.checkIsOffer(host)

if (keyOffer && keyOffer in self.db.activeOffers && Date.now() <
self.db.activeOffers[keyOffer] + self.timeActiveOffer) {
  self.setIcon('green')
  if (self.db.notifyOffers[keyOffer] === 0) {
    chrome.tabs.executeScript(currentTab.id, { code: `(function(){
      if(!document.querySelector('.fooBarClassName-overlay')) {
        var iframe = document.createElement('iframe');
        iframe.className = 'fooBarClassName-overlay';
        iframe.style.position = 'fixed';
        iframe.style.top = '20px';
        iframe.style.right = '20px';
        iframe.style.zIndex = '10000000';
        iframe.style.borderRadius = '6px';
        iframe.style.boxShadow = '0px 0px 10px gray';
        iframe.style.border = '0px';
        iframe.style.borderRadius = '6px';
        iframe.style.width = '440px';
        iframe.style.minHeight = '324px';
        //iframe.style.minHeight = '260px';
        //iframe.style.minHeight = '178px';
        iframe.src = '${chrome.extension.getURL("")}popup.html?tabid=${currentTab.id}';
        document.body.appendChild(iframe);
      }
    })`;
  }
}

```

```

        //console.log("hear!!");
        setTimeout(function(){
            document.querySelector('.fooBarClassName-overlay').remove();
        }, ${self.notifyTimeout});
    } else {
        document.querySelector('.fooBarClassName-overlay').remove();
    }
    });` })
        self.db.notifyOffers[keyOffer] = 1;
    }
} else if (keyOffer) {
    self.setIcon('red')
    if (!(keyOffer in self.db.hiddenOffers) || Date.now() > self.db.hiddenOffers[keyOffer] +
self.timeHideOffer) {
        chrome.tabs.executeScript(
            currentTab.id,
            {
                code: `(function(){ if (!document.querySelector('.fooBarClassName-overlay')) {
                    var iframe = document.createElement('iframe');
                    iframe.className = 'fooBarClassName-overlay';
                    iframe.style.position = 'fixed';
                    iframe.style.top = '20px';
                    iframe.style.right = '20px';
                    iframe.style.zIndex = '10000000';
                    iframe.style.borderRadius = '6px';
                    iframe.style.boxShadow = '0px 0px 10px gray';
                    iframe.style.border = '0px';
                    iframe.style.borderRadius = '6px';
                    iframe.style.width = '440px';
                    iframe.style.minHeight = '324px';
                    //iframe.style.minHeight = '260px';
                    //iframe.style.minHeight = '178px';
                    iframe.src = '${chrome.extension.getURL("")}popup.html?tabid=${currentTab.id}';
                    document.body.appendChild(iframe);
                }})`;
            },
            function() {
                if (chrome.runtime.lastError) console.log('executeScript', chrome.runtime.lastError)
            }
        )
    }
} else self.setIcon('default')
if (chrome.runtime.lastError) console.log('query', chrome.runtime.lastError)
})
}

messenger() {
    const self = this
    chrome.runtime.onMessage.addListener(function(message, sender, sendResponse) {
        if (message.type === 'start') {
            if (!self.db.uid || parseInt(Date.now() / 1000) > self.db.actual_to) {
                self.getDataOfServer((e, d) => {
                    if (e)
                        return sendResponse({
                            error: e,
                            popupIcon: self.popupIcon,
                            currentOffer: self.currentOffer
                        })
                })
            }
        }
    })
}

```

```

        })
        sendResponse({ db: d, popupIcon: self.popupIcon, currentOffer: self.currentOffer })
    })
    } else sendResponse({ db: self.db, popupIcon: self.popupIcon, currentOffer: self.currentOffer
})

return true
} else if (message.type === 'hide-window') {
    chrome.tabs.executeScript(sender.tab.id, { code:
`; (function() { if (document.querySelector('.fooBarClassName-overlay'))
document.querySelector('.fooBarClassName-overlay').remove() }); ` })
    self.db.hiddenOffers[self.checkIsOffer(new URI(sender.tab.url).host().replace(/www\./, ""))] =
Date.now()
    chrome.storage.local.set({ hiddenOffers: self.db.hiddenOffers })
} else if (message.type === 'auth') {
    setTimeout(function() {
        self.getDataOfServer((e, d) => {
            if (e) return console.log({ error: e })
        })
    }, 5000)
} else if (message.type === 'search') {
    if (self.checkIsOffer(new URI(message.url).host().replace(/^www\./, ""))) {
        sendResponse({
            url: self.db.offers[self.currentOffer].get_cookies_url,
            uid: self.db.uid,
            ratesValue: self.db.offers[self.currentOffer].rates.value
        })
    } else sendResponse({})
    return true
}
})
chrome.tabs.onCreated.addListener(function() {
    self.checkTab()
})
chrome.tabs.onUpdated.addListener(function() {
    self.checkTab()
})
chrome.tabs.onActivated.addListener(function() {
    self.checkTab()
})
chrome.tabs.onHighlighted.addListener(function() {
    self.checkTab()
})
})

run() {
    const self = this
    setTimeout(() => {
        self.getDataOfServer((e, d) => {
            if (e) return console.log(e)
        })
    }, 500)
}
}

const application = new Megatron()
application.run()
})();

```

```

Поруп_mf.js //Клас для реалізації сповіщень (для браузера Mozilla Firefox)
$(document).ready(function() {
    const mainHost = 'http://toprecash.net'

    $('#popup_h1').text(chrome.i18n.getMessage('chrome_popup_h_title'))
    $('#notices').append(`<div id="notice_updating"
class="notice">${ chrome.i18n.getMessage('chrome_popup_notice_update_text')}</div>`)
    $('#footer_link_1')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_1_href'))
        .text(chrome.i18n.getMessage('chrome_popup_footer_link_1_text'))
    $('#footer_link_2')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_2_href'))
        .text(chrome.i18n.getMessage('chrome_popup_footer_link_2_text'))
    $('#footer_link_3')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_3_href'))
        .text(chrome.i18n.getMessage('chrome_popup_footer_link_3_text'))
    $('#button')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_c_btn_link_def'))
        .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_select_store'))

    chrome.runtime.sendMessage({ type: 'start' }, d => {
        if (!d || typeof d !== 'object') {
            // if unknown error
            $('#notices').html(`<div id="notice_updating" class="notice">unknown error</div>`)
        } else if (d && d.error) {
            // if error
            $('#notices').html(`<div id="notice_updating" class="notice">${d.error}</div>`)
        } else if (d.db.user) {
            // if auth
            $('#notices').empty()
            $('.user-info').append(`<div class="user-name">${d.db.user.uname}</div><div class="user-
balance">${d.db.user.ubalance}</div>`)
        } else if (d.db.msg) {
            // if no auth
            $('#notices').empty()
            $('#content')
                .children()
                .eq(0)
                .before(`<p id="auth_text">${d.db.msg}</p>`)
            $('.btn-row').append(`<a id="auth_button" class="button text-danger" href="${mainHost +
'/cabinet/orders/'}" target="_blank">${chrome.i18n.getMessage('chrome_popup_c_auth_btn_text')}</a>`)
        } else {
            console.log('unknown error')
        }
    })
    if (d.popupIcon) {
        switch (d.popupIcon) {
            case 'red':
                $('#button')
                    .attr('href', d.db.offers[d.currentOffer].get_cookies_url + '&uid=' + d.db.uid)
                    .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_activate'))
                if (!d.currentOffer || !(d.currentOffer in d.db.offers)) break
                $('.offers-list')
                    .children()
                    .eq(0).before(`<div class="offer-row">
                        <div class="offer-logo"></div>

```

```

        <div class="offer-info"><div class="offer-
name">${d.db.offers[d.currentOffer].offer_name}</div>
        <div class="offer-rates">${d.db.offers[d.currentOffer].rates.value}</div></div><div
class="offer-btns"></div></div>`)
    break
    case 'green':
    $('#button')
    .attr('href', "")
    .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_active'))
    $('#button').addClass('offer-active')
    $(document).on('click', '#button', function(e) {
    e.preventDefault()
    })
    if (!d.currentOffer || !(d.currentOffer in d.db.offers)) break
    $('offers-list')
    .children()
    .eq(0).before(`<div class="offer-row">
        <div class="offer-logo"></div>
        <div class="offer-info"><div class="offer-
name">${d.db.offers[d.currentOffer].offer_name}</div>
        <div class="offer-rates">${d.db.offers[d.currentOffer].rates.value}</div></div><div
class="offer-btns"></div></div>`)
    break
    default:
    $('#button')
    .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_c_btn_link_def'))
    .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_select_store'))
    break
    }
    if (window.location.search.indexOf('tabid') !== -1) {
    $('html').addClass('injected')
    $('.closeWindowLink a').text(chrome.i18n.getMessage('chrome_popup_disable_offer') +
d.db.offers[d.currentOffer].offer_name + chrome.i18n.getMessage('chrome_popup_disable_offer_time'))
    $(document).on('click', '.closeWindowLink a', function(e) {
    e.preventDefault()
    chrome.runtime.sendMessage({
    type: 'hide-window'
    })
    })
    }
    }
    })
    $(document).on('click', 'body a', function(e) {
    setTimeout(() => {
    window.close()
    }, 250)
    })
    })
}

```

Google_mf.js //Клас для роботи з пошуковими системами (для браузера Mozilla Firefox)

```

:(() => {
    try {
    if (!document.getElementsByTagName('head')[0].querySelector('[data-css-megatron]')) {
    let css = 'a.megatron-link:hover{color:#070!important;text-decoration:none!important;}'
    let style = document.createElement('style')
    style.setAttribute('data-css-megatron', '1')
    }
    }
}

```

```

    if (style.styleSheet) style.styleSheet.cssText = css
    else style.appendChild(document.createTextNode(css))
    document.getElementsByTagName('head')[0].appendChild(style)
  }
  document.querySelectorAll('cite').forEach((e, i) => {
    ;(e => {
      let url = e.innerText.trim()
      if (url.indexOf(' ') > -1) url = url.split(' ')[0]
      chrome.runtime.sendMessage({ type: 'search', url: url }, d => {
        try {
          if (d.url) {
            if (e.parentNode.previousElementSibling.previousElementSibling.parentNode &&
!e.parentNode.previousElementSibling.previousElementSibling.parentNode.getAttribute('data-megatron')) {
              let a = document.createElement('a')
              a.href = d.url + '&uid=' + d.uid
              a.setAttribute('target', '_blank')
              a.setAttribute('class', 'megatron-link')
              a.innerText = 'Toprecash.net кэшбэк ' + d.ratesValue
              a.style.color = '#e62727'
              a.style.fontWeight = 'bold'
              a.style.fontSize = '15px'
              a.style.textDecoration = 'none'
              a.style.paddingLeft = '24px'
              a.style.lineHeight = '22px'
              a.style.display = 'block'
              a.addEventListener('mouseover', function(e) {
                this.style.backgroundImage =
`url("${chrome.runtime.getURL("")}icons/icon_19x19_green.png")`
                this.style.backgroundColor = `0 0`
                this.style.backgroundColor = `19px`
                this.style.backgroundColor = `no-repeat`
              })
              a.addEventListener('mouseout', function(e) {
                this.style.backgroundImage =
`url("${chrome.runtime.getURL("")}icons/icon_19x19_action.png")`
                this.style.backgroundColor = `0 0`
                this.style.backgroundColor = `19px`
                this.style.backgroundColor = `no-repeat`
              })
              let icon = document.createElement('div')
              icon.style.backgroundColor =
`url("${chrome.runtime.getURL("")}icons/icon_19x19_action.png")`
              icon.style.backgroundColor = `0 0`
              icon.style.backgroundColor = `19px`
              icon.style.backgroundColor = `no-repeat`
              icon.style.position = `relative`
              icon.style.left = '-22px'
              icon.appendChild(a)

            e.parentNode.previousElementSibling.previousElementSibling.parentNode.setAttribute('data-megatron', '1')

            e.parentNode.previousElementSibling.previousElementSibling.parentNode.insertBefore(icon,
e.parentNode.previousElementSibling.previousElementSibling)
          }
        } catch (e) {
          console.log(e)
        }
      })
    })
  })

```

```

    }
  })
  })(e)
})
} catch (e) {
  console.log(e)
}
})()

```

Background_chrome.js //Головний клас плагіна (для браузера Chrome)

```

;(function() => {
  try {
    if (!document.getElementsByTagName('head')[0].querySelector('[data-css-megatron'])) {
      let css = 'a.megatron-link:hover{color:#070!important;text-decoration:none!important;}'
      let style = document.createElement('style')
      style.setAttribute('data-css-megatron', '1')
      if (style.styleSheet) style.styleSheet.cssText = css
      else style.appendChild(document.createTextNode(css))
      document.getElementsByTagName('head')[0].appendChild(style)
    }
    document.querySelectorAll('cite').forEach((e, i) => {
      ;(e => {
        let url = e.innerText.trim()
        if (url.indexOf(' ') > -1) url = url.split(' ')[0]
        chrome.runtime.sendMessage({ type: 'search', url: url }, d => {
          try {
            if (d.url) {
              if (e.parentNode.previousElementSibling.previousElementSibling.parentNode &&
              !e.parentNode.previousElementSibling.previousElementSibling.parentNode.getAttribute('data-megatron')) {
                let a = document.createElement('a')
                a.href = d.url + '&uid=' + d.uid
                a.setAttribute('target', '_blank')
                a.setAttribute('class', 'megatron-link')
                a.innerText = 'Торprecash.net кэшбэк ' + d.ratesValue
                a.style.color = '#e62727'
                a.style.fontWeight = 'bold'
                a.style.fontSize = '15px'
                a.style.textDecoration = 'none'
                a.style.paddingLeft = '24px'
                a.style.lineHeight = '22px'
                a.style.display = 'block'
                a.addEventListener('mouseover', function(e) {
                  this.style.backgroundImage =
                  `url("${chrome.runtime.getURL(")}icons/icon_19x19_green.png")`
                  this.style.backgroundColor = `0 0`
                  this.style.backgroundColor = `19px`
                  this.style.backgroundColor = `no-repeat`
                })
                a.addEventListener('mouseout', function(e) {
                  this.style.backgroundImage =
                  `url("${chrome.runtime.getURL(")}icons/icon_19x19_action.png")`
                  this.style.backgroundColor = `0 0`
                  this.style.backgroundColor = `19px`
                  this.style.backgroundColor = `no-repeat`
                })
                let icon = document.createElement('div')

```

```

        icon.style.backgroundImage
        `url("${chrome.runtime.getURL("")}icons/icon_19x19_action.png")`
        icon.style.backgroundPosition = `0 0`
        icon.style.backgroundSize = `19px`
        icon.style.backgroundRepeat = `no-repeat`
        icon.style.position = `relative`
        icon.style.left = '-22px'
        icon.appendChild(a)
    }
    e.parentNode.previousElementSibling.previousElementSibling.parentNode.setAttribute('data-megatron', '1')
    e.parentNode.previousElementSibling.previousElementSibling.parentNode.insertBefore(icon,
    e.parentNode.previousElementSibling.previousElementSibling)
    }
    } catch (e) {
    console.log(e)
    }
    })
    })(e)
    } catch (e) {
    console.log(e)
    }
    })()

```

Google_chrome.js //Клас для роботи з пошуковими системами (для браузера Chrome)

```

;(O => {
    try {
        if (!document.getElementsByTagName('head')[0].querySelector('[data-css-megatron]')) {
            let css = 'a.megatron-link:hover{color:#070!important;text-decoration:none!important;}'
            let style = document.createElement('style')
            style.setAttribute('data-css-megatron', '1')
            if (style.styleSheet) style.styleSheet.cssText = css
            else style.appendChild(document.createTextNode(css))
            document.getElementsByTagName('head')[0].appendChild(style)
        }
        document.querySelectorAll('cite').forEach((e, i) => {
            ;(e => {
                let url = e.innerText.trim()
                if (url.indexOf(' ') > -1) url = url.split(' ')[0]
                chrome.runtime.sendMessage({ type: 'search', url: url }, d => {
                    try {
                        if (d.url) {
                            if (e.parentNode.previousElementSibling.previousElementSibling.parentNode &&
                            !e.parentNode.previousElementSibling.previousElementSibling.parentNode.getAttribute('data-megatron')) {
                                let a = document.createElement('a')
                                a.href = d.url + '&uid=' + d.uid
                                a.setAttribute('target', '_blank')
                                a.setAttribute('class', 'megatron-link')
                                a.innerText = 'Топrecash.net кэшбэк ' + d.ratesValue
                                a.style.color = '#e62727'
                                a.style.fontWeight = 'bold'
                                a.style.fontSize = '15px'
                                a.style.textDecoration = 'none'
                                a.style.paddingLeft = '24px'
                                a.style.lineHeight = '22px'

```



```

        a.style.display = 'block'
        a.addEventListener('mouseover', function(e) {
            this.style.backgroundImage =
`url("${chrome.runtime.getURL()}icons/icon_19x19_green.png")`
            this.style.backgroundPosition = `0 0`
            this.style.backgroundSize = `19px`
            this.style.backgroundRepeat = `no-repeat`
        })
        a.addEventListener('mouseout', function(e) {
            this.style.backgroundImage =
`url("${chrome.runtime.getURL()}icons/icon_19x19_action.png")`
            this.style.backgroundPosition = `0 0`
            this.style.backgroundSize = `19px`
            this.style.backgroundRepeat = `no-repeat`
        })
        let icon = document.createElement('div')
        icon.style.backgroundImage =
`url("${chrome.runtime.getURL()}icons/icon_19x19_action.png")`
        icon.style.backgroundPosition = `0 0`
        icon.style.backgroundSize = `19px`
        icon.style.backgroundRepeat = `no-repeat`
        icon.style.position = `relative`
        icon.style.left = '-22px'
        icon.appendChild(a)

```

```
e.parentNode.previousElementSibling.previousElementSibling.parentNode.setAttribute('data-megatron', '1')
```

```

e.parentNode.previousElementSibling.previousElementSibling.parentNode.insertBefore(icon,
e.parentNode.previousElementSibling.previousElementSibling)
    }
    } catch (e) {
        console.log(e)
    }
    })
    })(e)
    })
    } catch (e) {
        console.log(e)
    }
    })()

```

Попуп_chrome.js //Клас для реалізації сповіщень (для браузера chrome)

```

$(document).ready(function() {
    const mainHost = 'http://toprecash.net'

    $('#popup_h1').text(chrome.i18n.getMessage('chrome_popup_h_title'))
    $('#notices').append(`<div id="notice_updating"
class="notice">${chrome.i18n.getMessage('chrome_popup_notice_update_text')}</div>`)
    $('#footer_link_1')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_1_href'))
        .text(chrome.i18n.getMessage('chrome_popup_footer_link_1_text'))
    $('#footer_link_2')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_2_href'))
        .text(chrome.i18n.getMessage('chrome_popup_footer_link_2_text'))
    $('#footer_link_3')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_3_href'))

```

```

    .text(chrome.i18n.getMessage('chrome_popup_footer_link_3_text'))
    $('#button')
    .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_c_btn_link_def'))
    .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_select_store'))

chrome.runtime.sendMessage({ type: 'start' }, d => {
  if (!d || typeof d !== 'object') {
    // if unknown error
    $('#notices').html(`<div id="notice_updating" class="notice">unknown error</div>`)
  } else if (d && d.error) {
    // if error
    $('#notices').html(`<div id="notice_updating" class="notice">${d.error}</div>`)
  } else if (d.db.user) {
    // if auth
    $('#notices').empty()
    $('.user-info').append(`<div class="user-name">${d.db.user.uname}</div><div class="user-
balance">${d.db.user.ubalance}</div>`)
  } else if (d.db.msg) {
    // if no auth
    $('#notices').empty()
    $('#content')
    .children()
    .eq(0)
    .before(`<p id="auth_text">${d.db.msg}</p>`)
    $('.btn-row').append(`<a id="auth_button" class="button text-danger" href="${mainHost +
/cabinet/orders/" target="_blank">${chrome.i18n.getMessage('chrome_popup_c_auth_btn_text')}</a>`)
  } else {
    console.log('unknown error')
  }
  if (d.popupIcon) {
    switch (d.popupIcon) {
      case 'red':
        $('#button')
        .attr('href', d.db.offers[d.currentOffer].get_cookies_url + '&uid=' + d.db.uid)
        .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_activate'))
        if (!d.currentOffer || !(d.currentOffer in d.db.offers)) break
        $('.offers-list')
        .children()
        .eq(0).before(`<div class="offer-row">
          <div class="offer-logo"></div>
          <div class="offer-info"><div class="offer-
name">${d.db.offers[d.currentOffer].offer_name}</div>
          <div class="offer-rates">${d.db.offers[d.currentOffer].rates.value}</div></div><div
class="offer-btns"></div></div>`)
        break
      case 'green':
        $('#button')
        .attr('href', "")
        .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_active'))
        $('#button').addClass('offer-active')
        $(document).on('click', '#button', function(e) {
          e.preventDefault()
        })
        if (!d.currentOffer || !(d.currentOffer in d.db.offers)) break
        $('.offers-list')
        .children()

```

```

        .eq(0).before(`<div class="offer-row">
            <div class="offer-logo"></div>
            <div class="offer-info"><div class="offer-
name">${d.db.offers[d.currentOffer].offer_name}</div>
            <div class="offer-rates">${d.db.offers[d.currentOffer].rates.value}</div></div><div
class="offer-btns"></div></div>`)
        break
    default:
        $('#button')
            .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_c_btn_link_def'))
            .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_select_store'))
        break
    }
    if (window.location.search.indexOf('tabid') !== -1) {
        $('html').addClass('injected')
        $('.closeWindowLink a').text(chrome.i18n.getMessage('chrome_popup_disable_offer') +
d.db.offers[d.currentOffer].offer_name + chrome.i18n.getMessage('chrome_popup_disable_offer_time'))
        $(document).on('click', '.closeWindowLink a', function(e) {
            e.preventDefault()
            chrome.runtime.sendMessage({
                type: 'hide-window'
            })
        })
    }
})
})

```

Background_opera.js // Головний клас програми (для браузера Опера)

```

;(function() => {
    class Megatron {
        constructor() {
            const self = this
            self.mainHost = 'http://toprecash.net'
            self.authUrl = self.mainHost + '/cabinet/orders/'
            self.timeActiveOffer = 60 * 60 * 1000 // 1 hour
            self.timeHideOffer = 10 * 60 * 1000 // 10 min
            self.notifyTimeout = 10 * 1000 // 10 s
            self.popupIcon = 'default'
            self.currentOffer = null
            self.isAddTargetUrls = false
            self.db = {}
            self.messenger()
            chrome.storage.local.get(null, db => {
                self.db = db
            })
        }

        getDataUrl() {
            const self = this
            return self.mainHost + '/local/dirix.megatron/api/browser.extension/data/get/?uid=' + (self.db.uid
|| '') + '&token=' + (self.db.token || '')
        }

        getDataOfServer(c) {
            const self = this

```

```

axios
.get(self.getDataUrl())
.then(d => {
  try {
    if (d.data.status === 'ok') {
      d.data.hiddenOffers = {};
      d.data.activeOffers = {};
      d.data.notifyOffers = {};
    }
    for (let i in d.data.offers) {
      d.data.offers[i].offer_logo = d.data.offers[i].offer_logo.replace(":80", "");
      d.data.offers[i].host = new URI(d.data.offers[i].offer_site_url).host().replace(/^www\./, "");
      d.data.offers[i].domain = new URI(d.data.offers[i].offer_site_url).domain(true);
      d.data.offers[i].tld = new URI(d.data.offers[i].offer_site_url).tld()
    }
    chrome.storage.local.set(d.data, () => {
      chrome.storage.local.get(null, db => {
        self.db = db
        c(null, db)
        chrome.tabs.query({}, tabs => {
          for (let i = 0, n = tabs.length; i < n; i++) {
            if (self.checkIsOffer(new URI(tabs[i].url).host().replace(/^www\./, ""))) {
              chrome.tabs.executeScript(tabs[i].id, {
                code:
                `(function(){if(document.querySelector('.fooBarClassName-overlay'))
document.querySelector('.fooBarClassName-overlay').remove() });`
              })
            }
          }
          self.checkTab()
        })
        if (self.isAddTargetUrls === false) {
          self.isAddTargetUrls = true
          const urls = []
          for (let i in db.offers) urls.push(db.offers[i].host)
          chrome.webRequest.onBeforeRedirect.addListener(
            function(detail) {
              if (new RegExp('^' + self.mainHost).test(detail.initiator)) {
                let testUrl = urls.find(e => new RegExp('^https?:/(www\\.)?(\\w-){1,}\\.' +
e).test(detail.redirectUrl))
                if (testUrl) {
                  //self.db.activeOffers[self.checkIsOffer(new
URI(detail.redirectUrl).host().replace(/^www\./, ""))] = Date.now()
                  //chrome.storage.local.set({ activeOffers: self.db.activeOffers })
                  let offerId = self.checkIsOffer(new
URI(detail.redirectUrl).host().replace(/^www\./, ""));
                  self.db.activeOffers[offerId] = Date.now();
                  self.db.notifyOffers[offerId] = 0;
                  chrome.storage.local.set({ activeOffers: self.db.activeOffers });
                  chrome.storage.local.set({
                    notifyOffers:
                    self.db.notifyOffers });
                }
              }
            },
            {
              urls: ['<all_urls>'],
              types: ['main_frame']
            },
            ['responseHeaders']
          )
        }
      })
    })
  }
}

```

```

    }
  })
})
} else {
  chrome.storage.local.clear() => {
    self.currentOffer = null
    c(null, { msg: chrome.i18n.getMessage('chrome_popup_c_auth_text') })
  })
}
} catch (e) {
  chrome.storage.local.clear() => {
    self.currentOffer = null
    c(chrome.i18n.getMessage('chrome_popup_error'))
    console.log(e)
  })
}
})
).catch(e => {
  chrome.storage.local.clear() => {
    self.currentOffer = null
    c(chrome.i18n.getMessage('chrome_popup_error'))
    console.log(e)
  })
})
}

setIcon(type) {
  const self = this
  if (!type) {
    self.popupIcon = 'default'
    return chrome.browserAction.setIcon({ path: 'icons/icon_19x19.png' })
  } else if (type === 'red') {
    self.popupIcon = 'red'
    chrome.browserAction.setIcon({ path: 'icons/icon_19x19_action.png' })
  } else if (type === 'green') {
    self.popupIcon = 'green'
    chrome.browserAction.setIcon({ path: 'icons/icon_19x19_green.png' })
  } else {
    self.popupIcon = 'default'
    chrome.browserAction.setIcon({ path: 'icons/icon_19x19.png' })
  }
}

checkIsOffer(host) {
  const self = this
  self.currentOffer = null
  if (!host || !self.db || !self.db.offers) return false
  for (let i in self.db.offers) {
    if (host === self.db.offers[i].host) {
      self.currentOffer = i
      return i
    }
  }
  let d = host.match(/.\/g)
  let tld = new URI('http://' + host).tld()
  if (d && d.length >= (tld.indexOf('.') === -1 ? 2 : 3)) host = host.replace(/^\w+\/./, "")
  if (!host || !self.db || !self.db.offers) return false
}

```

```

for (let i in self.db.offers) {
  if (host === self.db.offers[i].host) {
    self.currentOffer = i
    return i
  }
}
return false
}

checkTab() {
  const self = this
  chrome.tabs.query({ active: true }, function(currentTab) {
    if (!currentTab || !currentTab.length) return
    currentTab = currentTab[0]
    let host = new URI(currentTab.url).host().replace(/^www\/./, "")
    if (/^google\./.test(host) || host === 'yandex.ru') {
      self.setIcon('default')
    }
    // if (!currentTab || currentTab.status !== 'complete') return
    if (!currentTab) return
    if (new RegExp('^ + self.authUrl').test(currentTab.url)) {
      chrome.tabs.executeScript(
        currentTab.id,
        {
          code: `(function(){ if (document.querySelector('[name="Login"]') &&
!document.querySelector('[name="Login"]').getAttribute('plugin-event')) {
              document.querySelector('[name="Login"]').setAttribute('plugin-event', '1')
              document.querySelector('[name="Login"]').addEventListener('click', function(){
                chrome.runtime.sendMessage({ type: 'auth' })
              })
            }})`;
        },
        function() {
          if (chrome.runtime.lastError) console.log('executeScript', chrome.runtime.lastError)
        }
      )
      return
    }

    if (!self.db.offers) return

    if (host === 'yandex.ru') {
      chrome.tabs.executeScript(currentTab.id, { file: 'yandex.js' }, function() {
        if (chrome.runtime.lastError) console.log('executeScript', chrome.runtime.lastError)
      })
      return
    }
    if (/^google\./.test(host)) {
      chrome.tabs.executeScript(currentTab.id, { file: 'google.js' }, function() {
        if (chrome.runtime.lastError) console.log('executeScript', chrome.runtime.lastError)
      })
      return
    }
  })

  let keyOffer = self.checkIsOffer(host)

```

```

        if (keyOffer && keyOffer in self.db.activeOffers && Date.now() <
self.db.activeOffers[keyOffer] + self.timeActiveOffer) {
            self.setIcon('green')
            //chrome.tabs.executeScript(currentTab.id, { code:
`;function(){if(document.querySelector('.fooBarClassName-overlay'))
document.querySelector('.fooBarClassName-overlay').remove() });` })
            if( self.db.notifyOffers[keyOffer] == 0 ) {
                chrome.tabs.executeScript(currentTab.id, { code: `(function(){
                    if(!document.querySelector('.fooBarClassName-overlay')) {
                        var iframe = document.createElement('iframe');
                        iframe.className = 'fooBarClassName-overlay';
                        iframe.style.position = 'fixed';
                        iframe.style.top = '20px';
                        iframe.style.right = '20px';
                        iframe.style.zIndex = '10000000';
                        iframe.style.borderRadius = '6px';
                        iframe.style.boxShadow = '0px 0px 10px gray';
                        iframe.style.border = '0px';
                        iframe.style.borderRadius = '6px';
                        iframe.style.width = '440px';
                        iframe.style.minHeight = '324px';
                        //iframe.style.minHeight = '260px';
                        //iframe.style.minHeight = '178px';
                        iframe.src = '${chrome.extension.getURL("")}popup.html?tabid=${currentTab.id}';
                        document.body.appendChild(iframe);

                                                                    //console.log("hear!!");

                        setTimeout(function(){
                            document.querySelector('.fooBarClassName-overlay').remove();
                        },${self.notifyTimeout});
                    } else {
                        document.querySelector('.fooBarClassName-overlay').remove();
                    }
                })();` })
                    self.db.notifyOffers[keyOffer] = 1;
            }
        }
    } else if (keyOffer) {
        self.setIcon('red')
        if (!(keyOffer in self.db.hiddenOffers) || Date.now() > self.db.hiddenOffers[keyOffer] +
self.timeHideOffer) {
            chrome.tabs.executeScript(
                currentTab.id,
                {
                    code: `(function(){ if (!document.querySelector('.fooBarClassName-overlay')) {
                        var iframe = document.createElement('iframe');
                        iframe.className = 'fooBarClassName-overlay';
                        iframe.style.position = 'fixed';
                        iframe.style.top = '20px';
                        iframe.style.right = '20px';
                        iframe.style.zIndex = '10000000';
                        iframe.style.borderRadius = '6px';
                        iframe.style.boxShadow = '0px 0px 10px gray';
                        iframe.style.border = '0px';
                        iframe.style.borderRadius = '6px';
                        iframe.style.width = '440px';
                        iframe.style.minHeight = '324px';
                        //iframe.style.minHeight = '260px';
                        //iframe.style.minHeight = '178px';

```

```

        iframe.src = '${chrome.extension.getURL("")}popup.html?tabid=${currentTab.id}';
        document.body.appendChild(iframe);
    });`
    },
    function() {
        if (chrome.runtime.lastError) console.log('executeScript', chrome.runtime.lastError)
    }
)
}
} else self.setIcon('default')
if (chrome.runtime.lastError) console.log('query', chrome.runtime.lastError)
})
}

messenger() {
    const self = this
    chrome.runtime.onMessage.addListener(function(message, sender, sendResponse) {
        if (message.type === 'start') {
            if (!self.db.uid || parseInt(Date.now() / 1000) > self.db.actual_to) {
                self.getDataOfServer((e, d) => {
                    if (e)
                        return sendResponse({
                            error: e,
                            popupIcon: self.popupIcon,
                            currentOffer: self.currentOffer
                        })
                    sendResponse({ db: d, popupIcon: self.popupIcon, currentOffer: self.currentOffer })
                })
            } else sendResponse({ db: self.db, popupIcon: self.popupIcon, currentOffer: self.currentOffer
        })

        return true
    } else if (message.type === 'hide-window') {
        chrome.tabs.executeScript(sender.tab.id, {
            code:
            `;(function(){if(document.querySelector('.fooBarClassName-overlay'))
            document.querySelector('.fooBarClassName-overlay').remove() })();`
        })
        self.db.hiddenOffers[self.checkIsOffer(new URI(sender.tab.url).host().replace(/www\./, ""))] =
        Date.now()
        chrome.storage.local.set({ hiddenOffers: self.db.hiddenOffers })
    } else if (message.type === 'auth') {
        setTimeout(function() {
            self.getDataOfServer((e, d) => {
                if (e) return console.log(e)
            })
        }, 5000)
    } else if (message.type === 'search') {
        if (self.checkIsOffer(new URI(message.url).host().replace(/^www\./, ""))) {
            sendResponse({
                url: self.db.offers[self.currentOffer].get_cookies_url,
                uid: self.db.uid,
                ratesValue: self.db.offers[self.currentOffer].rates.value
            })
        } else sendResponse({})
        return true
    }
})
})
chrome.tabs.onCreated.addListener(function() {
    self.checkTab()

```



```

    })
    chrome.tabs.onUpdated.addListener(function() {
      self.checkTab()
    })
    chrome.tabs.onActivated.addListener(function() {
      self.checkTab()
    })
    chrome.tabs.onHighlighted.addListener(function() {
      self.checkTab()
    })
  }

  run() {
    const self = this
    setTimeout(() => {
      self.getDataOfServer((e, d) => {
        if (e) return console.log(e)
      })
    }, 500)
  }
}

const application = new Megatron()
application.run()
})()

```

Google_opera.js //Клас для роботи з пошуковими системи (для браузера Опера)

```

;(0 => {
  try {
    if (!document.getElementsByTagName('head')[0].querySelector('[data-css-megatron]')) {
      let css = 'a.megatron-link:hover{color:#070!important;text-decoration:none!important;}'
      let style = document.createElement('style')
      style.setAttribute('data-css-megatron', '1')
      if (style.styleSheet) style.styleSheet.cssText = css
      else style.appendChild(document.createTextNode(css))
      document.getElementsByTagName('head')[0].appendChild(style)
    }
    document.querySelectorAll('cite').forEach((e, i) => {
      ;(e => {
        let url = e.innerText.trim()
        if (url.indexOf(' ') > -1) url = url.split(' ')[0]
        chrome.runtime.sendMessage({ type: 'search', url: url }, d => {
          try {
            if (d.url) {
              if (e.parentNode.previousElementSibling.previousElementSibling.parentNode &&
              !e.parentNode.previousElementSibling.previousElementSibling.parentNode.getAttribute('data-megatron')) {
                let a = document.createElement('a')
                a.href = d.url + '&uid=' + d.uid
                a.setAttribute('target', '_blank')
                a.setAttribute('class', 'megatron-link')
                a.innerText = 'Торрeсh.net кэшбэк ' + d.ratesValue
                a.style.color = '#e62727'
                a.style.fontWeight = 'bold'
                a.style.fontSize = '15px'
                a.style.textDecoration = 'none'
                a.style.paddingLeft = '24px'
                a.style.lineHeight = '22px'

```

```

        a.style.display = 'block'
        a.addEventListener('mouseover', function(e) {
            this.style.backgroundImage =
`url("${chrome.runtime.getURL()}icons/icon_19x19_green.png")`
            this.style.backgroundPosition = `0 0`
            this.style.backgroundSize = `19px`
            this.style.backgroundRepeat = `no-repeat`
        })
        a.addEventListener('mouseout', function(e) {
            this.style.backgroundImage =
`url("${chrome.runtime.getURL()}icons/icon_19x19_action.png")`
            this.style.backgroundPosition = `0 0`
            this.style.backgroundSize = `19px`
            this.style.backgroundRepeat = `no-repeat`
        })
        let icon = document.createElement('div')
        icon.style.backgroundImage =
`url("${chrome.runtime.getURL()}icons/icon_19x19_action.png")`
        icon.style.backgroundPosition = `0 0`
        icon.style.backgroundSize = `19px`
        icon.style.backgroundRepeat = `no-repeat`
        icon.style.position = `relative`
        icon.style.left = '-22px'
        icon.appendChild(a)

```

```
e.parentNode.previousElementSibling.previousElementSibling.parentNode.setAttribute('data-megatron', '1')
```

```
e.parentNode.previousElementSibling.previousElementSibling.parentNode.insertBefore(icon,
e.parentNode.previousElementSibling.previousElementSibling)
```

```

    }
    } catch (e) {
        console.log(e)
    }
    })
    })(e)
    })
    } catch (e) {
        console.log(e)
    }
    })()

```

Попуп_opera.js //Клас для реалізації сповіщень (для браузера опера)

```

$(document).ready(function() {
    const mainHost = 'http://toprecash.net'

    $('#popup_h1').text(chrome.i18n.getMessage('chrome_popup_h_title'))
    $('#notices').append(`<div id="notice_updating"
class="notice">${chrome.i18n.getMessage('chrome_popup_notice_update_text')}</div>`)
    $('#footer_link_1')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_1_href'))
        .text(chrome.i18n.getMessage('chrome_popup_footer_link_1_text'))
    $('#footer_link_2')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_2_href'))
        .text(chrome.i18n.getMessage('chrome_popup_footer_link_2_text'))
    $('#footer_link_3')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_footer_link_3_href'))

```

```

    .text(chrome.i18n.getMessage('chrome_popup_footer_link_3_text'))
    $('#button')
    .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_c_btn_link_def'))
    .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_select_store'))

chrome.runtime.sendMessage({ type: 'start' }, d => {
  if (!d || typeof d !== 'object') {
    // if unknown error
    $('#notices').html(`<div id="notice_updating" class="notice">unknown error</div>`)
  } else if (d && d.error) {
    // if error
    $('#notices').html(`<div id="notice_updating" class="notice">${d.error}</div>`)
  } else if (d.db.user) {
    // if auth
    $('#notices').empty()
    $('.user-info').append(`<div class="user-name">${d.db.user.uname}</div><div class="user-
balance">${d.db.user.ubalance}</div>`)
  } else if (d.db.msg) {
    // if no auth
    $('#notices').empty()
    $('#content')
    .children()
    .eq(0)
    .before(`<p id="auth_text">${d.db.msg}</p>`)
    $('.btn-row').append(`<a id="auth_button" class="button text-danger" href="${mainHost +
/cabinet/orders/" target="_blank">${chrome.i18n.getMessage('chrome_popup_c_auth_btn_text')}</a>`)
  } else {
    console.log('unknown error')
  }
  if (d.popupIcon) {
    switch (d.popupIcon) {
      case 'red':
        $('#button')
        .attr('href', d.db.offers[d.currentOffer].get_cookies_url + '&uid=' + d.db.uid)
        .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_activate'))
        if (!d.currentOffer || !(d.currentOffer in d.db.offers)) break
        $('.offers-list')
        .children()
        .eq(0).before(`<div class="offer-row">
          <div class="offer-logo"></div>
          <div class="offer-info"><div class="offer-
name">${d.db.offers[d.currentOffer].offer_name}</div>
          <div class="offer-rates">${d.db.offers[d.currentOffer].rates.value}</div></div><div
class="offer-btns"></div></div>`)
        break
      case 'green':
        $('#button')
        .attr('href', "")
        .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_active'))
        $('#button').addClass('offer-active')
        $(document).on('click', '#button', function(e) {
          e.preventDefault()
        })
        if (!d.currentOffer || !(d.currentOffer in d.db.offers)) break
        $('.offers-list')
        .children()

```

```

    .eq(0).before(`<div class="offer-row">
      <div class="offer-logo"></div>
      <div class="offer-info"><div class="offer-
name">\${d.db.offers[d.currentOffer].offer_name}</div>
      <div class="offer-rates">\${d.db.offers[d.currentOffer].rates.value}</div></div><div
class="offer-btns"></div></div>`)
      break
    default:
      $('#button')
        .attr('href', mainHost + chrome.i18n.getMessage('chrome_popup_c_btn_link_def'))
        .text(chrome.i18n.getMessage('chrome_popup_c_btn_text_select_store'))
      break
    }
    if (window.location.search.indexOf('tabid') !== -1) {
      $('html').addClass('injected')
      $('.closeWindowLink a').text(chrome.i18n.getMessage('chrome_popup_disable_offer') +
d.db.offers[d.currentOffer].offer_name + chrome.i18n.getMessage('chrome_popup_disable_offer_time'))
      $(document).on('click', '.closeWindowLink a', function(e) {
        e.preventDefault()
        chrome.runtime.sendMessage({
          type: 'hide-window'
        })
      })
    }
  })
})

```

Відгук керівника економічного розділу

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файла	Опис
Пояснювальні документи	
Самойленко.doc	Пояснювальна записка до проекту. Документ Word.
Самойленко.pdf	Пояснювальна записка до проекту в форматі PDF.
Програма	
diplom.zip	Архів. Містить коди програми і откомпільовану програму
Презентація	
Самойленко.ppt	Презентація проекту.