

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧОРНОМОРСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ПЕТРА МОГИЛИ

Шлапак Євгеній Олегович

УДК 004.4'233

**СТВОРЕННЯ МОДУЛЮ КАСТОМНИХ СКРИПТІВ ДЛЯ ВИКОНАННЯ У
ВЕБСЕРИДОВИЩ**

Галузь знань 12 «Інформаційні технології» за спеціальністю

122 «Комп'ютерні науки»

122 - БКР.А - 402.21710231

Автореферат

бакалаврської кваліфікаційної роботи на здобуття освітньої кваліфікації

«бакалавр з комп'ютерних наук»

Миколаїв – 2021

Бакалаврська кваліфікаційна робота є рукопис.

Робота виконана в Чорноморському національному університеті імені Петра Могили Міністерства освіти і науки України на кафедрі інтелектуальних інформаційних систем

Науковий керівник: старший викладач, Кошевий В.В.

Рецензент: доцент кафедри інженерії програмного забезпечення, канд. пед. Наук,
Кірей Катерина Олександрівна

Захист відбудеться «___» червня 2021 р. о 9⁰⁰ год. на засіданні екзаменаційної комісії (ауд. 2-403) у Чорноморському національному університеті імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

З бакалаврською кваліфікаційною роботою можна ознайомитися в бібліотеці Чорноморського національного університету імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

Автореферат представлений «___» червня 2021 р.

Секретар
екзаменаційної комісії,
викладач, аспірант

Скакодуб О.С.

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. Досить часто на сайті, для покращення користувацького досвід, не вистачає мінімальної за масштабом реалізації функції. Навіть при наявності прямої комунікації між розробниками та користувачами, потрібно, щоб більшість людей, що користуються веб-ресурсом, запросили додавання нового функціоналу, перед початком його розробки. Розширення для браузеру, що надає змогу створювати та запускати сторонні сценарії виконання, могло б забезпечити поліпшення користувацького досвіду, спростити користування веб-сторінками та зменшити навантаження веб-розробників. Адже можливість додавати власний код до сайту відкриває користування веб-сторінками по-новому. Адже можливість додавати власні функції до улюблених сайтів надає абсолютну свободу дій.

Метою бакалаврської кваліфікаційної роботи є надати користувачам можливість запускати власний код на сторонніх сторінках, змінювати дизайн своїх улюблених сайтів та додавати до них додатковий функціонал.

Практичне значення отриманих результатів. В процесі виконання даної дипломної роботи було зроблено аналіз існуючих методів запуску користувацького коду на веб сторінках. Виявлено їх сильні та слабкі сторони та на їх основі створено власне розширення, що виконує JavaScript код на обраних користувачем сторінках. Було розроблено та налагоджено власний модуль кастомних скриптів для виконання у вебсередовищі.

Структура кваліфікаційної роботи. Пояснювальна записка до бакалаврської кваліфікаційної роботи складається із вступу, ____ розділів, висновків, додатків. Загальний обсяг роботи складає ____ сторінки, ____ рисунків, ____ таблиць та ____ посилань на літературні джерела.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У **вступі** описано актуальність роботи. Запити користувачів можуть бути неймовірно різноманітними від додавання нічного режиму на сайт до автоматизації рутинних задач. Звичайно, задовільнити всі потреби, що можуть з'явитися під час роботи з окремим веб-додатком, практично неможливо. Тому важливо надати користувачам можливість запускати власний код на сторонніх сторінках, змінювати

дизайн своїх улюблених сайтів та додавати до них додатковий функціонал. Розширення для браузеру, що надає змогу створювати та запускати сторонні сценарії виконання, могло б забезпечити поліпшення користувацького досвіду, спростити користування веб-сторінками та зменшити навантаження веб-розробників. Адже можливість додавати власний код до сайту відкриває користування веб-сторінками по-новому. Адже можливість додавати власні функції до улюблених сайтів надає абсолютну свободу дій. На сьогоднішній день розроблено ряд додатків до низки популярних браузерів, які надають можливості написання JavaScript коду та використання його на веб-сторінках. Проте, вони мають певні обмеження або нагромадження інтерфейсу, що затрудняє написання та використання додаткового коду.

До ряду можливостей стороннього коду можна додати:

- автоматизація;
- стильові зміни сторінок;
- створення додаткових функцій;
- додавання гарячих клавіш на сайт;
- зміна DOM-структури сторінок.

Відповідно до мети було виділено наступні завдання дипломної роботи:

- провести аналіз існуючих модулів кастомних скриптів для виконання у вебсередовищі;
- визначити основні програмні інструменти для розробки власного модулю;
- розробка та налагодження власного модулю кастомних скриптів для виконання у вебсередовищі.

У першому розділі було проведено аналіз предметної області: оглянуто сферу використання розширень для браузеру, проаналізовано існуючі методи запуску користувацького коду, їх переваги та недоліки, та виходячи з них сформовано задачі, необхідні для виконання.

Розширення браузера - це мікропрограма, яка розширює (доповнює) функціонал браузера, вона також здатна через браузер вбудовуватися в хмарний софт, і розширити його функціонал. Їх називають browser extensions (розширення браузера),

plug-in (плагін), або add-on (доповнення), і зараз вони підтримуються практично всіма браузерами. Найпопулярніші розширення для google chrome і розширення для firefox.

Головне призначення розширень зрозуміло вже з самої назви - розширити функціонал браузера. Плагіном можна вбудувати свої елементи в будь-який хмарний сервіс з метою інтеграції, додаткових можливостей, створення мікросервіса, або просто як додатковий інструмент.

Хмарне програмне забезпечення, програми управління складом і багатьох інших систем з ведення бізнесу стає все більш популярним, але ніякої хмарний провайдер не дає можливості вбудуватися в нього, і підігнати під себе. В такому випадку розширення - гарна можливість взяти добре продуманий хмарний сервіс, і вмонтувати в нього одну або кілька своїх кнопок. Їй можна привласнити функцію експорту даних, і налагодити процес обміну даними між різними внутрішніми системами. Таким чином можна отримати додатковий функціонал до чужого хмарного ПЗ або сайту.

Це можуть бути фонові сервіси різного характеру: перевірки, підказки, календарі, перекладачі, помічники та інші.

Розширення може представляти собою цілий мікросервіс. Це мікро застосунки, які викликаються "по кнопці" і взаємодіють зі сторінкою. Потрібні вони для сервісів, які має сенс запускати в середовищі іншого сайту / додатка. Він здатний зчитувати інформацію з сайту, на якому його викликали, і тим самим істотно спростити користувачеві взаємодія. це можуть бути будь-які сервіси заміток, скріншотів або відео, вони можуть не просто зберігати дані, а відразу вивантажувати в хмару, або відправляти, наприклад поштою.

Викликати інструменти не закриваючи браузер - дуже зручна можливість для розробників. І подібний функціонал можуть забезпечити розширення для інтернет браузерів.

Після проведення огляду існуючих розширень для браузеру Google Chrome було поставлено ряд задач та критеріїв відносно продукту, що розробляється. Серед основних задач, котрі повинне вирішувати розширення свої місця зайняли:

- справне виконання зазначеного коду на відповідних йому сторінках;

- механізм збереження, та відтворення скриптів для повторного використання на веб сторінках;
- проведення основних операцій таких як редагування, видалення та вимикання/вимикання виконання коду.

Щодо критеріїв, аналіз існуючих засобів автоматичного запуску коду допоміг конкретизувати та виявити основні та найважливіші аспекти працюючого та зручного програмного засобу. Як показала практика використання деяких з неведених у попередньому підрозділі продуктів, список важливих критеріїв має наступний вигляд:

- розширення повинне працювати безвідмовно, повинне виконувати збережений код при **кожному** завантаженні сторінки, що відповідає скрипту;
- розширення не повинне використовувати сторонніх ресурсів для збереження, відтворення або запуску коду, бути самостійним продуктом, незалежним від інших продуктів;
- розширення повинне працювати повністю без інтернет сполучення, всі збереженні данні повинні зберігатися локально;
- робота з розширенням повинна проводитися в окремому рорир-вікні, без відвідування сторонніх сторінок;
- розширення має мати простий інтерфейс з мінімальною кількістю кнопок, полів та інших елементів відтворення.

Проаналізувавши більшість існуючих засобів ін'єкції користувацького коду, не було виявлено жодного кандидату, що повністю задовольняв існуючі критерії. Хоча й були досить достойні кандидати, але вони або ж мали рудиментарний функціонал, або ж страждали від його відсутності. Аналіз показав, що розробки власного модулю для запуску користувацького коду не уникнути.

У **другому розділі** здійснено опис існуючих методик розробки програмного забезпечення. Досконало вивчивши розглянуті методики та порівнявши їх сильні та слабкі сторони, проаналізувавши доречність кожного з представлених претендентів, було прийнято зважене рішення про використання методології, представленої

четвертою в наведеному вище списці, Scrum. Цю методику було обрано через порівняну простоту та найбільшу відповідність до поставленої задачі. Scrum – поширена методологія розробки програмного забезпечення. Scrum — це кістяк процесу, який включає набір методів і попередньо визначених ролей. Головні дійові особи — ScrumMaster, той хто опікується процесами, веде їх і працює як керівник проекту, Власник Продукту, людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті сторін, та Команду, яка включає розробників. Обов'язковим артефактом при роботі за методологією Scrum є Product Backlog. Product backlog — це документ, який має список вимог до функціональності, які упорядковані згідно зі ступенем важливості. Product backlog представляє список того, що повинно бути реалізовано. Елементи цього списку називаються «історіями» (user story) або елементами backlog-у (backlog items). Product backlog відкритий для редагування усім учасникам Scrum-процесу.

Через відсутність команди при виконанні дипломної роботи методику Scrum було спрощено та адаптовано для комфортної та ефективної роботи однієї персони. Було вилючено стендапи перед командою, адже команда налічувала лише одну людину. Було збережено ітеративність імплементації змін для розділення процесів розробки. Backlog було організовано завдяки функціоналу інтерактивної дошки планування задач Trello, розміщеної у відкритому доступі в інтернеті.

Виконавши основну задачу – планування проекту, створивши список задач по створенню проекту, провівши аналіз поставлених задач, обравши методику розробки програмного забезпечення, створивши дошку в Trello, було розпочато розробку застосунку. Адже при наявності конкретного плану роботи, процес розробки буде найбільш ефективним, менше часу буде затрачено на вибір та постановку задачі в процесі виконання роботи. Також це допомагає не відволікатися на конкретизацію та формалізацію завдання під час роботи над проектом. Також управління проектами (або проектний менеджмент) як раз і допомагає швидко і ефективно досягати поставлених цілей. Крім того, в процесі цього формується ціла система комплексів, які можуть бути використані для досягнення загальних цілей компанії, а також розробляється схема розумного розподілу ресурсів.

Проведено аналіз існуючих засобів та інструментів, який вказав на всі переваги та недоліки методів розробки продукту та реалізації потрібних функцій. Як вже було зазначено, функціонал розширення має мати такий вигляд:

- запуск коду на сторінках;
- збереження коду для повторного використання.

Серед можливих способів відтворення коду на веб сторінках можна виділити наступні методи: запуск через консоль розробника, вбудована в мову JavaScript функція `eval` та запуск коду методом додавання його в HTML тег `script` та відтворення тегу на сторінці. З наведених варіантів з найкращої сторони себе показала інтеграція тегу `script` у веб сторінку, виходячи з цього було обрано саме такий спосіб запуску коду для подальшої реалізації проекту.

Для збереження скриптів було розглянуто наведений вище список варіантів, кожен з них має свої переваги та недоліки. Деякі з неведених вище варіантів були більш релевантними для поставленої задачі, деякі були менш доречними, а певна частина й того менше, були достойні лише згадки, як існуючі та працюючі. Серед згаданих методів зберігання інформації було наведено:

- реляційні серверні бази даних;
- нереляційні хмарні сервіси збереження даних;
- збереження інформації в файлах;
- збереження даних в Cookies браузеру;
- зберігання коду у вбудованому сховищі Google Chrome `chrome.storage`.

Більшість зі згаданих методів не відповідали вимогам розробки продукту, були залежні від інтернет сполучення, або реалізації серверної частини застосунку, котра не входила до поставлених задач розробки. Після аналізу всіх способів відтворення потрібного функціоналу, відкидавши один за одним існуючі варіанти, було обрано використовувати `chrome.storage`.

У третьому розділі було розпочато роботу над наступним етапом розробки – створення інтерфейсу. Насправді через відсутність досвіду розробки користувацьких інтерфейсів, було прийнято рішення зробити інтерфейс максимально простим.

Було описано принципи роботи розширення, описано програмну реалізацію та приведено керівництво користувача.

Розширення справляється з поставленими задачами на відмінно. Код невідмовно виконується при кожному запуску певної веб сторінки. Збережені скрипти не зникають зі списку до тих пір, поки користувач самостійно їх не видалить. Редагування коду відбувається буквально в два натиска лівої кнопки миші. Код оновлюється мментально та вже з новим завантаженням сторінки виконується в оновленому вигляді.

Також користувач може обирати на свій розсуд рівень вкладеності сайту, на якій хоче додати додатковий код. Це може бути як конкретно головна сторінка сайту, так і сторінка, наприклад, перегляду відео на відеохостингу youtube, незважаючи на те, яке відео переглядається.

Варто відмітити, що додаток дозволяє створювати декілька скриптів для одного сайту. Це досить зручна функція, що дозволяє відокремити код, що відповідає різним цілям та виконує різні дії.

Інтерфейс розширення досить простий, це не є випадковістю, адже для спрощення частини програми, з якою зустрічається користувач, було прикладено немало зусиль. Хоча розібратися зі застосунком достатньо легко, але для уникнення будь-яких труднощів, було створено попередній підрозділ «Керівництво користувача», де детально описано процес експлуатації розширення. Проведена робота над додатком та керівництвом дає впевненість, що кожен зможе користуватися створеним продуктом.

Звичайно, варто згадати, що створення власного скрипту потребує навичок написання коду на мові JavaScript, але це надає змогу створити будь-який функціонал для будь-якого сайту. Можливості розширення обмежуються лише фантазією користувача та навичками використання мови програмування JavaScript.

У розділі з охорони праці було:

- розглянуто робоче місце інженера-програміста, який є об'єктом дослідження з охорони праці.

- був проведений аналіз виробничих факторів, таких як температура повітря, освітлення, шум та інше на робочому місці.
- у ході роботи над розділом були розглянуті детальні питання щодо рівня шуму. Розрахунок у свою чергу показав, що загальний рівень шуму є помірним.

Метою охорони праці є створення системи безпеки, збереження здоров'я та працездатності людини в процесі праці. Ця мета досягається виконанням відповідних правил, тобто комплексом взаємопов'язаних правил техніки безпеки, що здійснюються суб'єктом управління і спрямовано на об'єкт управління.

Аналіз і оцінка стану умов та безпеки праці – це обов'язкова складова роботи керівництва щодо планування відповідних заходів з охорони праці. На підприємстві аналіз і оцінка стану умов та безпеки праці здійснюється на підставі наступних загальних показників:

- рівень виробничого травматизму;
- рівень професійних захворювань пов'язаних з умовами праці;
- кількість працівників, що працюють в умовах, які не відповідають санітарно-гігієнічним нормам;
- кількість обладнання, що не відповідає вимогам нормативних актів про охорону праці;
- кількість технологічних процесів, що не відповідають вимогам нормативно-правових актів з охорони праці;
- кількість будівель та споруд, технічний стан яких не відповідає будівельним нормам і правилам;
- забезпечення працівників засобами індивідуального захисту;
- забезпеченість працівників санітарно-побутовими приміщеннями;
- витрати на покращення стану безпеки, гігієни праці та виробничого середовища;
- витрати на відшкодування збитків потерпілим від нещасних випадків та професійних захворювань, що пов'язані з умовами праці;

- витрати на розслідування та ліквідацію наслідків аварій, нещасних випадків та професійних захворювань.

Мета впровадження системи охорони праці — це всебічне сприяння виконанню вимог, які повністю ліквідують, нейтралізують або знижують до допустимих норм вплив на працівників небезпечних та шкідливих факторів виробничого середовища, забезпечують усунення джерел небезпеки, ізолювання від них персоналу, використання засобів, що усувають небезпечні ситуації та підвищують технічну безпеку, створюють надійні санітарно-гігієнічні та ергономічні умови. Це передбачає встановлення конкретних кількісних показників діяльності виробничих підрозділів, підтримування котрих в заданих межах забезпечує досягнення основної мети щодо організації безпечних та нешкідливих умов праці.

Роботодавець зобов'язаний створити в кожному структурному підрозділі і на робочому місці умови праці відповідно до вимог нормативних актів, а також забезпечити дотримання прав працівників, гарантованих законодавством про охорону праці.

ЗАГАЛЬНІ ВИСНОВКИ

Метою дипломної роботи було надати користувачам можливість запускати власний код на сторонніх сторінках, змінювати дизайн своїх улюблених сайтів та додавати до них додатковий функціонал.

В процесі виконання даної дипломної роботи було зроблено аналіз існуючих методів запуску користувацького коду на веб сторінках. Виявлено їх сильні та слабкі сторони та на їх основі створено власне розширення, що виконує JavaScript код на обраних користувачем сторінках.

Так як створення додатку для браузеру відбувається на веб технологіях, було використано HTML, CSS та JavaScript. Це дозволило створити найбільш доступний і простий інтерфейс.

Було виконано завдання:

- проведено аналіз існуючих модулів кастомних скриптів для виконання у вебсередовищі;
- визначено основні програмні інструменти для розробки власного модулю;
- розроблено та налагоджено власний модуль кастомних скриптів для виконання у вебсередовищі.

Дипломну роботу виконано повністю.

АНОТАЦІЯ

Шлапак Євгеній Олегович. Створення модулю кастомних скриптів для виконання у вебсередовищі. – На правах рукопису.

Бакалаврська кваліфікаційна робота на здобуття освітньої кваліфікації «бакалавр з комп'ютерних наук» в галузі знань 12 «Інформаційні технології» за спеціальністю 122 «Комп'ютерні науки».

Чорноморський національний університет імені Петра Могили, Миколаїв.

Дипломну роботу виконано на 70 аркушах, вона містить 3 додатки та перелік посилань на використані джерела з 19 найменувань. У роботі наведено 24 рисунки.

Метою даної дипломної роботи є створення механізму для запуску користувацьких скриптів на сайтах для поліпшення користувацького досвіду та розширення функціоналу сайтів.

У роботі проведено аналіз існуючих рішень указаної задачі – браузерних додатків що виконують користувацькі скрипти та додатки, що автоматизують користувацькі дії. Виконано їх порівняння з погляду виконання поставленої задачі та зручності використання користувачем. Для розв'язання задачі було з поміж створення нового браузера с можливостями додавання додаткових скриптів та прив'язки їх до окремих сторінок та створення додатку, який розширює функціонал, до вже існуючого браузера, було обрано останнє через простіші як процес розробки, так і умови використання продукту користувачем.

З поміж існуючих браузерів було обрано Google Chrome через найбільшу популярність серед користувачів мережі Інтернет, через зручні інструменти розробника та через розвинений магазин розширень браузеру.

Ключові слова: додаток, браузер, Google Chrome, скрипт, JavaScript, сайт, користувацький досвід, jQuery, WEB-розробка.

ABSTRACT

Shlapak Yevhenii Olegovich. Creating a custom script module for execution in the web environment. – On the rights of the manuscript.

Bachelor's qualification work for the educational qualification "Bachelor of Computer Science" in the field of knowledge 12 "Information Technology" in the specialty 122 "Computer Science".

Petro Mohyla Black Sea National University, Mykolaiv.

Thesis is done on 70 sheets, it contains 3 appendices and a list of references to the sources used with 19 titles. The paper presents 24 figures.

The purpose of this thesis is to create a mechanism for running custom scripts on sites to improve the user experience and expand the functionality of sites.

The paper analyzes the existing solutions of this problem - browser applications that execute user scripts and applications that automate user actions. The user performs their comparison in terms of the task and ease of use. To solve the problem, between creating a new browser with the ability to add additional scripts and linking them to individual pages and creating an application that extends the functionality to an existing browser, the latter was chosen because of simpler both the development process and terms of use product by the user.

Among the existing browsers, Google Chrome was chosen because of its popularity among Internet users, user-friendly developer tools, and a well-developed browser extension store.

Key words: application, browser, Google Chrome, script, JavaScript, site, user experience, jQuery, WEB-development.