

НАЦИОНАЛЕН КОНКУРС "МЛАДИ ТАЛАНТИ"

ТЕМА НА ПРОЕКТА

Website Chronometer

Автор:

*Севдалина Георгиева Шидерова
ГПЧЕ "Йоан Екзарх гр. Варна, 11 клас
shiderova.sevdalina@gmail.com, 0879 311 457*

Научен ръководител:

*Емил Еремиев
emil.eremiev, 0888 988 891*

1 Увод

"*Website Chronometer*" представлява добавка към интернет браузъра "*Google Chrome*". Добавките са софтуерни програми, които могат да променят и подобряват функционалността на браузъра.

Нашата добавка предоставя възможност за ограничаване на достъпа до определен сайт след изтичане на зададено от потребителя време. Уеб страницата може да бъде посетена отново след полунощ, когато таймерът бива нулиран и добавката е изцяло презаредена.

Проектът има за цел да повиши концентрацията на хората чрез ограничаване на времето, прекарано в безполезни сайтове. Те несъзнателно губят повече време, отколкото предполагат, сърфирайки в интернет пространството, докато разглеждат снимки или публикации, което намалява тяхната продуктивност многократно. Идеята на тази разработка е чрез задаване на позволеното време, потребителят сам да ограничи достъпа си, като прекара повече време върху важните и стойностни неща, отколкото губещите времето интернет страници.

Добавката работи, като записва фиксирания от потребителя URL адрес и го сравнява с активния в момента. При съвпадане се активира таймер, който се прекратява, когато браузърът се затвори или потребителят отвори друг уеб сайт. Добавката разпознава уеб сайта, който се използва, и не отчита таймер за никой друг от в момента неактивните такива.

Потребителят има възможност да избере определен брой минути и когато таймерът ги достигне, добавката автоматично прекратява достъпа до уеб страницата, като започва да препраща потребителя към търсачката на "*Google Chrome*" всеки път, когато опита да отвори страницата. Достъпът се възстановява след полунощ, когато потребителят има възможност да избере наново сайт и позволени минути.

2 Структура

2.1 manifest.json

Особеностите за създадения "*Chrome extension*" се декларираат в основния файл "*manifest.json*", съдържащ името, описанието и версията му. Ролята на този файл е да изпрати до браузъра информация за функциите на добавката.

За изпълнението на тези функции е нужно деклариране на разрешения. За нашата добавка е нужен "*activeTab permission*", който предоставя достъп до активния уеб сайт, и "*tabs permission*", осъществяващ извеждането на URL адреса му.

Освен това този файл посочва два ресурсни файла чрез "*browser_action*", оформящи дизайна на самата добавка. По този начин се имплементира потребителския интерфейс.

2.2 style.css

CSS е език за описание на стилове. Ние го използваме заедно с HTML за описване на представянето на добавката. В него е включена големината на "*popup*" полето, както и разположението на "*input*" полетата и бутоните, намиращи се в него.

2.3 icon.png

Това е изображение, което се позиционира в дясно на браузъра и при натискане активира прозорец, наречен "*popup*", в който се задават задачите за изпълнение. То се фиксира в основния файл.

2.4 popul.html

Това е стандартен HTML файл, създаващ изскачащото "*popup*" меню в отговор на кликане на потребителя върху "*icon.png*". Той определя съдържанието му, отговаря за потребителския интерфейс. В нашия случай включва две "*input*" полета и два бутона, както и уеб сайт и таймер, които се записват след стартиране на добавката.

2.5 `popup.js`

Този JavaScript файл изпълнява действителната логика на *"popup"* прозореца. Той може да приема стойностите, записани в *"input"* полетата на *"popul.html"*, и да ги предава на файловете, които се нуждаят от тази информация.

2.6 `content.js`

Този файл е нужен, ако добавката трябва да има достъп до уеб страници и да осъществява връзка с тях. Той е част от интернет сайтовете, които потребителят посещава, а не от самата добавка. Това е единственият файл, който има достъп до детайли, свързани с посетените уеб страници, и има позволение да прави промени в тях и да записва данни.

"Content.js" комуникира с добавката и останалите файлове чрез размяна на съобщения и запитвания до *"background.js"*.

2.7 `background.js`

Това представлява невидима страница, която съдържа главната логика и контролираща поведението на добавката. В локалната памет на *"background.js"* под формата на променливи се съхранява и обработва цялата информация, тъй като този файл може да получава информация от *"popup.js"*, тоест данните, въведени от потребителя, и от *"content.js"*, тоест информация за съдържанието на уеб страниците. Това е връзката между двата файла.

3 Техническа реализация

Проектът е реализиран чрез предаване на информация с помощта на съобщения между *"content.js"*, *"background.js"* и *"popup.js"* и използване на локалната браузър памет като база данни.

3.1 Локална памет

Локалната памет е памет в интернет браузъра, която позволява записването на информация без времево ограничение. Освен това записаните

данни се запамятват дори след затваряне на браузъра и изключване на компютъра, което е от голямо значение за нашата добавка. Също така локалната памет улеснява многократно достъпа до данните, тъй като те не се губят.

3.2 URL адрес

При въвеждане на URL адрес от потребителя *"porup.js"* го записва в локалната си памет, а *"background.js"* взима стойността от там и я записва като стойност на променливата *"websites"* съответно в своята локална памет.

Едновременно с това *"content.js"* отчита активния уеб сайт и записва неговия URL адрес. *"Background.js"* изпраща запитване към *"content.js"* за тази стойност и след получаването ѝ, я записва като локална променлива *"sentURL"*. Така в *"background.js"* се сравняват въведения URL адрес, който се взима от локалната памет, и активния уеб сайт.

Чрез поредица от няколко проверки в кода е осъществено разпознаване на сайта независимо от вида на записването му в *"input"* полето. Например *"www.abv.bg"*, *"abv.bg"* и *"http://www.abv.bg/"* биват възприети от *"porup.js"* и предадени чрез съобщения на *"background.js"*

3.3 Таймер

Потребителят има възможност да зададе ограничение на времето, прекарано в избран от него уеб сайт, под формата на минути. Аналогично на ситуацията със зададения URL адрес, *"porup.js"* записва избраното време в своята локална памет, а *"background.js"* взима стойността от там и я записва като числена стойност на променливата *"time"* съответно в своята локална памет.

Едновременно с това при съвпадане на активния и зададения уеб сайт се навлиза във функция. Тя активира таймер, който се записва като стойност на числовата променлива *"countDown"* в локалната памет на *"background.js"*. Тази числова променлива променя стойността си всяка една секунда, като става с единица по-голяма. При излизане от функцията променливата запазва стойността си и при повторно навлизане тази

стойност не се нулира.

3.4 Прекратяване на достъп до уеб сайт

Когато времето на потребителя, прекарано в определен уеб сайт, съвпадне с зададеното в началото, то достъпът до уеб страницата бива отнет, като потребителят автоматично бива пренасочван към търсачката на *"GoogleChrome"*.

Тази функция се осъществява по следния начин. В *"background.js"* се сравняват зададения и активирания таймер. Те съвпадат, когато е изпълнено следното равенство: $60 * \text{стойността на променливата "countDown"} / = \text{стойността на променливата "time"} /$, тъй като това изравнява мерните единици. Ако равенството е изпълнено, *"background.js"* изпраща съобщение до *"content.js"*, при чието получаване потребителят бива пренасочен към онлайн търсачката.

4 Описание на добавката

При натискане на иконка, намираща се в десния ъгъл на интернет браузъра *"Google Chrome"*, изскача *"popup"* прозорец с едно *"input"* поле *"Choose your website"*, в което следва да бъде введен URL адреса на желаните за следене сайт. При натискане на бутона *"Add"*, URL адреса бива записан в паметта и може да се вижда от потребителя в *"popup"* прозореца, докато не бъде сменен.

Когато уеб страницата бива въведена в съответното поле, се появява ново *"input"* поле, в което потребителят въвежда времето за позволен достъп в минути. Стойността се приема след натискане на бутона *"Set"*.

5 Използвани технологии

За осъществяване на проекта са използвани следните езици за програмиране: JavaScript, HTML, CSS. Използвана е и JavaScript библиотеката jQuery.

Добавките обединяват вичките си файлове в един архивиран файл с

разширение .srx, който лесно може да бъде изтеглен и инсталиран от интернет потребителите.

Осъщестен е бърз и лесен достъп до добавката. След имплементация на потребителски интерфейс, ние създаваме малката икона в дясно от интернет търсачката благодарение на *"browser action"* във файла *"manifest.json"*.

6 Постигнати резултати

Създадена е напълно безплатна софтуерна програма, работеща напълно и изпълняваща безпроблемно всички функции за един уеб сайт.

Подробно са разгледани браузър добавки с подобна функционалност и сходни цели. Обърнато е особено внимание на мненията и критиките на техните потребители. Установени са недостъците на тези добавки и нашето приложение се старее да премахне част от тях и да ги превърне в свои предимства.

За целта нашата програма отчита единствено активния уеб сайт и не отчита таймер за останалите отворени страници. Това се оказва проблем за много от хората, използващи останалите браузър добавки, тъй като търсейки информация в интернет за своя проект, таймерът отчита време за сайт, минимизиран е неактивното пространство. Нашето приложение се справя успешно с този проблем.

7 Заключение

Разработката представлява добавка към интернет браузъра *"Google Chrome"*, достъпна за всички потребители. Подходяща е за хора, които желаят да повишат концентрацията и продуктивността си. Особено препоръчителна е за тези с натоварено ежедневиe без време за губене. Добавката е направена и в помощ на ученици и студенти, които стоят с часове над компютъра без да осъзнават колко ценно време губят в безмислени неща.

Стилът на добавката е изключително улеснен и опростен, което дава

възможност да бъде използвана от хора без особени познания в областта програмирането и информационните технологии.

Добавката е изцяло на английски език с цел по-широко разпространение извън пределите на България.

8 Бъдещо развитие

В бъдеще имаме за цел да направим добавката функционираща за безброй много сайтове.

Сайтовете ще могат да бъдат разделяни в категории според времето, след което да бъдат блокирани. Тоест за всеки сайт потребителят ще може да избере различно позволено време. Това означава, че ако има п въведени сайта, то ще има п отделно активирани таймера.

Освен това всеки потребител ще има възможност да избира сайт, към който да бъде пренасочван след изтичане на позволеното време. Този сайт може да бъде различен за всеки един от записаните URL адреси.

За по-голямо удобство на потребителя оставащите секунди ще бъдат изобразявани в ъгъла на съответния сайт. Така няма да е нужно непрестанното отваряне на *"popup"* полето и претрупването му с п на брой таймери, което може да създаде объркване за това кой таймер за кой от URL адресите отговаря.

9 Използвана литература

[1] "<https://developer.chrome.com/extensions>", посетен последно на 9 април 2016

[2] "<http://stackoverflow.com/>" посетен последно на 7 април 2016

[3] "<https://robots.thoughtbot.com/how-to-make-a-chrome-extension>" посетен последно на 16 март 2016

[4] “<http://code.tutsplus.com/tutorials/developing-google-chrome-extensions-net-33076> посетен последно на 28 март 2016

НАЦИОНАЛЕН КОНКУРС "МЛАДИ ТАЛАНТИ"

ПРИЛОЖЕНИЕ

/Website Chronometer/

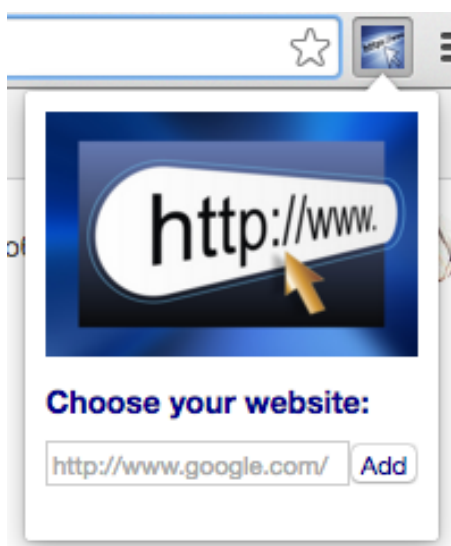
Автор:

*Севдалина Георгиева Шидерова
ГПЧЕ "Йоан Екзарх гр. Варна, 11 клас
shiderova.sevdalina@gmail.com, 0879 311 457*

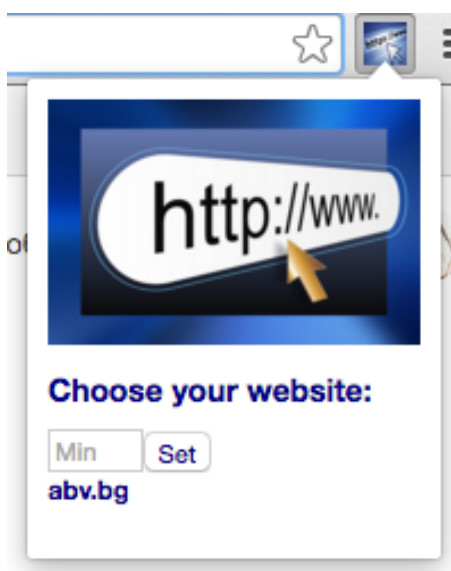
Научен ръководител:

*Емил Еремиев
ganchev@math.bas.bg, 0886 940 450*

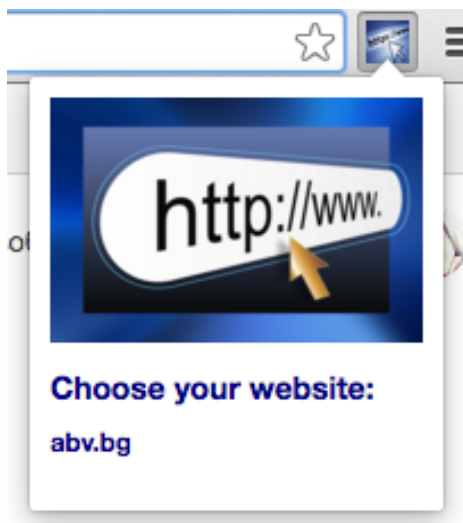
1. "Рорир" прозорец



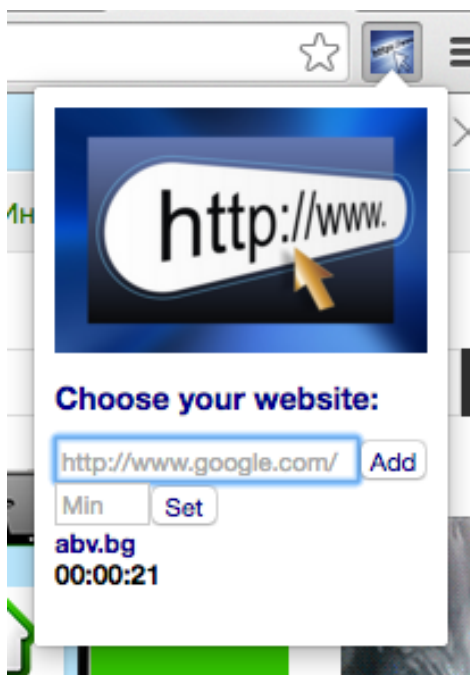
Фиг. 1 Начален екран



Фиг. 2 Екран след въвеждане на URL адрес

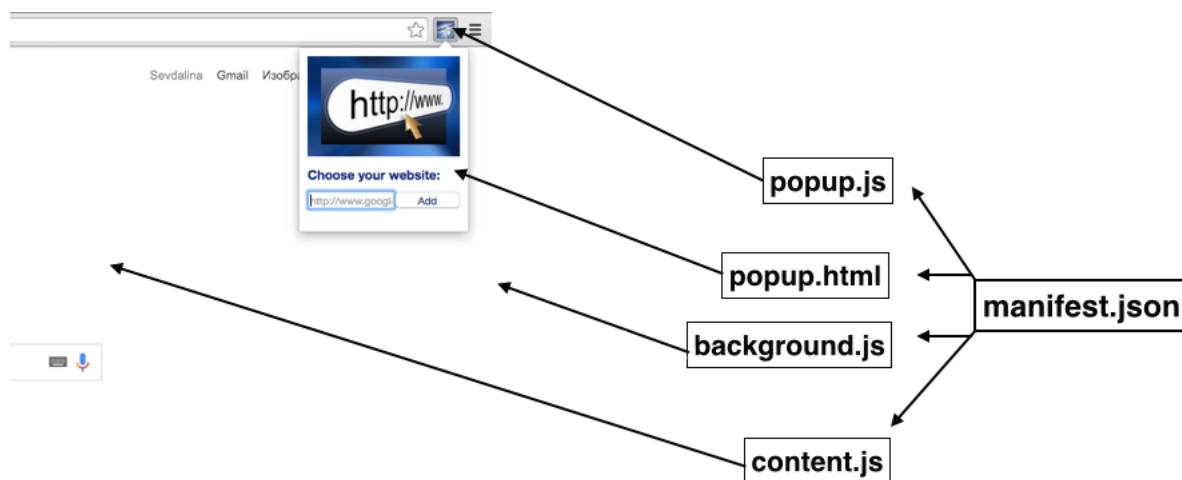


Фиг. 3 Екран след въвеждане на желаното време



Фиг. 4 Екран по време на престой във въведения уеб сайт

2. Структура на добавката



Фиг. 5 Структура на добавката