Razvoj dodataka za web preglednik

Jusup, Franjo

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:879352

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-12-22**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij / Stručni studij

Razvoj dodataka za web preglednik

Diplomski rad

Franjo Jusup

Osijek, 2019.

SADRŽAJ

1. UVOD

Dodatak (eng. *plugin)* omogućuje dodavanje novih značajki, opcija i mogućnosti internet pregledniku, tj. proširuje mogućnosti preglednika. Drugi učestaliji naziv za dodatak je proširenje (eng. *extensions*). Dodavanjem proširenja korisnik uređuje preglednik prema značajkama koje su mu potrebne. Omogućuju pregledniku dodatne mogućnosti koje se mogu izvršavati uz postojeće. Dodatne mogućnosti mogu biti moćne ili skromne, ovisno o potrebama [1].

Mogućnosti koje proširenje može dodati pregledniku su [2]:

- blokiranje prikazivanja oglasa
- optimizacija memorije za bolji rad preglednika
- dodavanje popisa ili napomena
- upravljanje zaporkama
- lakše kopiranje teksta s web stranice
- privatnost i sigurnije web pretraživanje.

Proširenja nude širok raspon dodatnih funkcija i mogućnosti za lakše obavljanje zadataka ili dobivanje više od web stranice koju se posjećuje. Nažalost, iako je većina proširenja korisna, postoje i proširenja koja negativno utječu na rad preglednika. Na primjer, dodavanjem proširenja za lakše kopiranje teksta može se dogoditi da proširenje doda oglase na web stranicu koju se pregledava, promijeni glavnu tražilicu i slično [1].

Proširenja se pišu u osnovnim, prijateljskim programskim jezicima kao što su: HTML, CSS i JavaScript. To su jezici koji se koriste za izradu najsuvremenijih web aplikacija i stranica. Izrada proširenja u ovim jezicima je sigurnija, brža i lakša, te omogućuje bolje praćenje zahtijevanih web standarda [1].

1.1. Razvoj dodatka za web preglednik

Potrebno je objasniti svrhu i alate za razvoj dodataka (*plugin-ova*) za web preglednik. U dogovoru s mentorom razviti jedan. Objasniti sve korake koje je potrebno napraviti u svrhu komercijalizacije istoga.

2. OPIS PROŠIRENJA I ALATA ZA RAZVOJ

Kada su se proširenja prvi put pojavila, bila su napisana u objektno orijentiranim programskim jezicima kao što je C++. Za izradu proširenja trebalo je puno vremena i stručnosti. Dodavanjem novog koda u preglednik olakšalo je napadačima iskorištavanje slabosti preglednika i ugrožavanje sigurnosti korisnika. Budući da su proširenja bila neobična i velika, uzrokovala su česte prekide u radu (rušenje) preglednika [1].

2.1. Što je proširenje

Proširenje za preglednik je program koji se izvodi u kontekstu (eng. *security sandbox*) web preglednika. Korisnicima pruža nove funkcionalnosti kombiniranjem postojećih značajki web preglednika i omogućuje izvršavanje više naredbi. Ona povećavaju produktivnost u radu, omogućuju lakši pristup podatcima dostupnim na web-u i povećavaju performanse preglednika. Dostupna su samo za računalne verzije web preglednika [3].

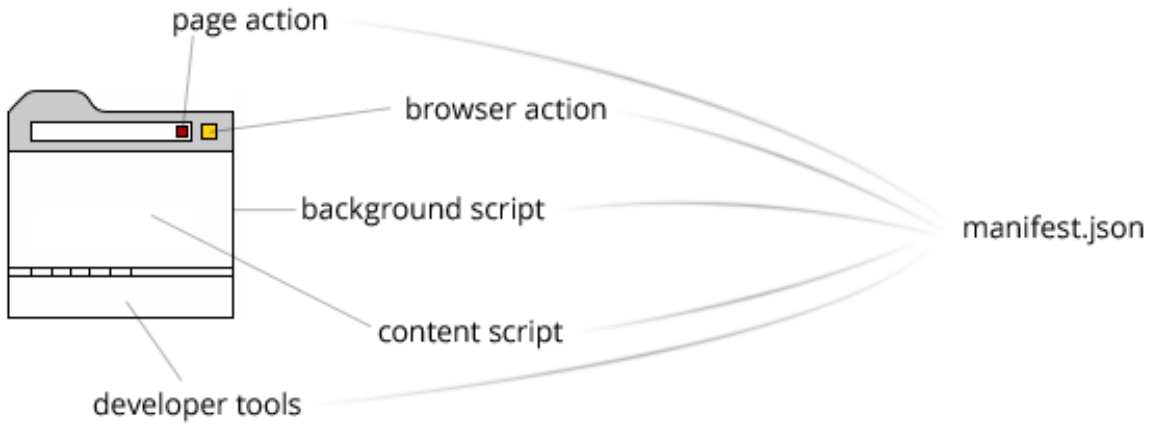
Preglednik Chrome prvi je omogućio API-je temeljene isključivo na HTML-u, CSS-u i JavaScriptu. Beta testiranje započelo je 2009. godine, a sljedeće je godine otvorena Chrome web trgovina. Od 2012. godine instalirano je 750 milijuna proširenja i drugih sadržaja iz web trgovine. Iste te godine Chrome je postao najpopularniji internet preglednik, svrgnuvši s trona Internet Explorer. Njegov je tržišni udio svake godine nastavljao rasti (2018. godine koristilo ga je 60% korisnika [4]).

2.2. Tehnologije za izradu proširenja

Proširenja korisnicima omogućuju veće mogućnosti (funkcionalnost), a programerima jednostavnost izrade. Proširenja su napisana u HTML-u, CSS-u, JavaScript-u i JSON-u. Izrada proširenja moguća je u svakom operacijskom sustavu jer je to skup HTML i JavaScript datoteka. Dizajn proširenja, koji se prikazuje korisniku, napisan je u HTML-u i CSS-u. JavaScript koristi se za logiku proširenja, pristup API-jima i komponentama koje omogućuje proširenje okvira (eng. *extension framework*). Za kreiranje manifest datoteke, koja prosljeđuje informacije pregledniku, koristi se JSON [3].

2.3. Arhitektura proširenja

Arhitektura proširenja ovisi o njenoj funkcionalnosti. Robusna proširenja mogu imati više komponenti. Elementi od kojih se sastoji arhitektura proširenja prikazana je na slici 2.1.[5].



Slika 2.1. Arhitektura proširenja [5]

2.3.1. Manifest

Polazna točka svakog proširenja je manifest.json datoteka koja treba imati valjani JSON objekt. Kako on treba izgledati prikazan je u programskom kodu 2.1.

```
{
    "name": "BrowserActionExtension",
    "version": "0.0.1",
    "manifest_version": 2,
    "browser_action": {
        "default_title": "That's the tool tip",
        "default_popup": "popup.html"
    }
}
```

Programski kod 2.1. Manifest.json [5]

Potrebna svojstva su *ime* (eng. *name*), *verzija* (eng. *version*) i *manifest verzija* (eng. *manifest_ version*). Verzija može biti bilo koji cijeli broj odvojen točkom kao u programskom kodu 2.1. Koristi se takav zapis zbog Google-ovog automatskog ažuriranja da se zna kada proširenje treba

ažurirati. Vrijednost manifest verzije treba biti cijeli broj, npr. broj 2. Manifest može sadržavati i druga svojstva ovisno o proširenju [5].

2.3.2. Pozadinske stranice (eng. *Background Pages*)

Svako proširenje ima nevidljivu pozadinsku stranicu koju pokreće preglednik. Postoje dvije vrste: trajne pozadinske stranice koje su aktivne cijelo vrijeme i stranice događaja koje su aktivne samo kada je to potrebno. Google potiče programere da koriste stranice događaja zbog uštede memorije i poboljšanja performansi preglednika. Pozadinska stranica/skripta ima ulogu mosta između ostalih dijelova proširenja.

U programskom kodu 2.2. prikazano je kako to treba biti unutar manifest.json *skripte*.

```
"background": {
    "scripts": ["background.js"],
    "persistent": false/true
}
```

Programski kod 2.2. Opis pozadinske stranice [5]

Iz programskog se koda može vidjeti da se, ako je trajno svojstvo lažno, tada koriste stranice događaja. Inače se koristi trajna pozadinska stranica [5].

2.3.3. Skripta sadržaja (eng. *Content Script*)

Ako je potreban pristup DOM-u trenutne stranice, tada se mora koristiti skripta sadržaja. Kod se izvodi u kontekstu trenutne web stranice, što znači da će se izvršavati svakim osvježavanjem. Za dodavanje takve skripte treba koristiti sintaksu prikazanu u programskom kodu 2.3. Prilikom dodavanja treba imati na umu da vrijednost podudaranja određuje za koje stranice će se skripta koristiti [5].

```
"content_scripts": [
    {
        "matches": ["http://*/*", "https://*/*"],
        "js": ["content.js"]
    }
]
```

Programski kod 2.3. Skripta sadržaja [5]

2.3.4. Akcija preglednika (eng. *Browser Action*)

Većina programera koristi akciju preglednika (eng. *browser action*) za izgradnju proširenja. Kada se ono postavi, ikona proširenja bit će prikazana u alatnoj traci s desne strane. Korisnici mogu kliknuti na ikonu i otvoriti skočni prozor koji je napisan u HTML-u i koji kontrolira korisnik (slika 2.2.).

Slika 2.2. Prikaz proširenja u alatnoj traci [5]

Manifest.json skripta mora imati podatke koji su prikazani u programskom kodu 2.4.

```
"browser_action": {
    "default_icon": {
        "19": "icons/19x19.png",
        "38": "icons/38x38.png"
    },
    "default_title": "That's the tool tip",
    "default_popup": "popup.html"
}
```

Programski kod 2.4. Podatci unutar manifest.json skripte

Default_title omogućuje da se prikaže mala alatna traka s nazivom proširenja kada korisnik pređe mišem preko ikone. *Default_popup* je HTML datoteka koja je učitana unutar skočnog prozora. Također je moguće postaviti oznaku preko ikone, a to se radi unutar pozadinske skripte [5].

2.3.5. Akcija stranice (eng. *Page Action*)

Svojstvo akcija stranice (eng. *page_action*) slično je akciji preglednika (eng. *browser action*), samo se ikona prikazuje u adresnoj traci (slika 2.3.).

Slika 2.3. Prikaz proširenja unutar adresne trake

Korištenjem ovoga svojstva ikona je u početku skrivena. Potrebno je odlučiti kada će se prikazivati. Npr. na slici iznad je prikazana *RSS* ikona koja će biti prikazana unutar adresne trake samo ako stranica sadrži vezu prema *RSS feed-u*. Ako je potrebno da ikona proširenja bude stalno vidljiva, bolje je koristiti svojstvo *browser_action* [5].

Za dodavanje svojstva akcije stranice (eng. *page action*) potrebno je upisati naredbe koje su prikazane u programskom kodu 2.5. unutar manifest.json skripte. Za razliku od *browser action* ikone, *page action* ikona ne može imati oznaku [5].

```
"page_action": {
    "default_icon": {
        "19": "images/icon19.png",
        "38": "images/icon38.png"
    },
    "default_title": "Google Mail",
    "default_popup": "popup.html"
}
```

Programski kod 2.5. Podatci unutar manifest.json skripte

2.3.6. Alat programera (eng. *Developer Tools*)

Chrome omogućuje dodavanje novih kartica pomoću *DeveloperTools-a*. Potrebno je dodati HTML stranicu koja će se otvoriti u trenutku kada se otvori: `devtools_page: devtools.html`. Unutar HTML datoteke treba staviti samo poveznicu na JavaScript datoteku: `<script src="devtools.js"></script>`;

U datoteku devtools.js dodaje se sljedeći kod (programski kod 2.6.):

```
chrome.devtools.panels.create(
    "TheNameOfYourExtension",
    "img/icon16.png",
    "index.html",
    function() {

    }
);
```

Programski kod 2.6. Kod unutar devtools.js skripte

Prikazani kod dodaje novu karticu s imenom *TheNameOfYourExtension* koja otvara indeks.html unutar *DeveloperTools-a*, ako se klikne na nju [5].

2.3.7. Omnibox

Omnibox je ključna riječ koja se pojavljuje u alatnoj traci ako se doda sljedeće svojstvo unutar manifest.json *skripte*: `omnibox: { keyword : yeah }`. Zatim je potrebno unutar pozadinske skripte dodati kod prikazan u programskom kodu 2.7. [5]

```
chrome.omnibox.onInputChanged.addListener(function(text, suggest)
{
    suggest([
      {content: text + " one", description: "the first one"},
      {content: text + " number two", description: "the second
entry"}
    ]);
});
chrome.omnibox.onInputEntered.addListener(function(text) {
    alert('You just typed "' + text + '"');
});
```

Programski kod 2.7. Kod za Omnibox

Kada se u alatnu traku upiše *yeah*, trebao bi se vidjeti prikaz sličan kao na slici 2.4.

Slika 2.4. Omnibox

2.4. API

API omogućuje izolirano izvršavanje JavaScript koda koji pripada proširenju. U preglednik je moguće instalirati više proširenja (jedno proširenje ne vidi postojanje drugog proširenja). Iz toga se podrazumijeva [3]:

- Različita proširenja neće se slučajno povezati jedno s drugim.
    - Proširenje ne može automatski pristupiti kodu ili memoriji koji pripadaju drugom proširenju.
- Nema sukoba s imenima.
    - Preglednik se neće zbuniti između proširenja *Script_A.js* i drugog proširenja *Script_A.js*
    - Isto vrijedi i za ostale resurse koji pripadaju proširenju kao što su: HTML, JSON datoteke, slike, itd.
- Proširenja se mogu međusobno povezivati na određeni kontrolirani način (za komunikaciju)
    - *Extension framework* pruža *API* za razmjenu poruka koje se koriste u jednokratnim i dugotrajnim vezama.

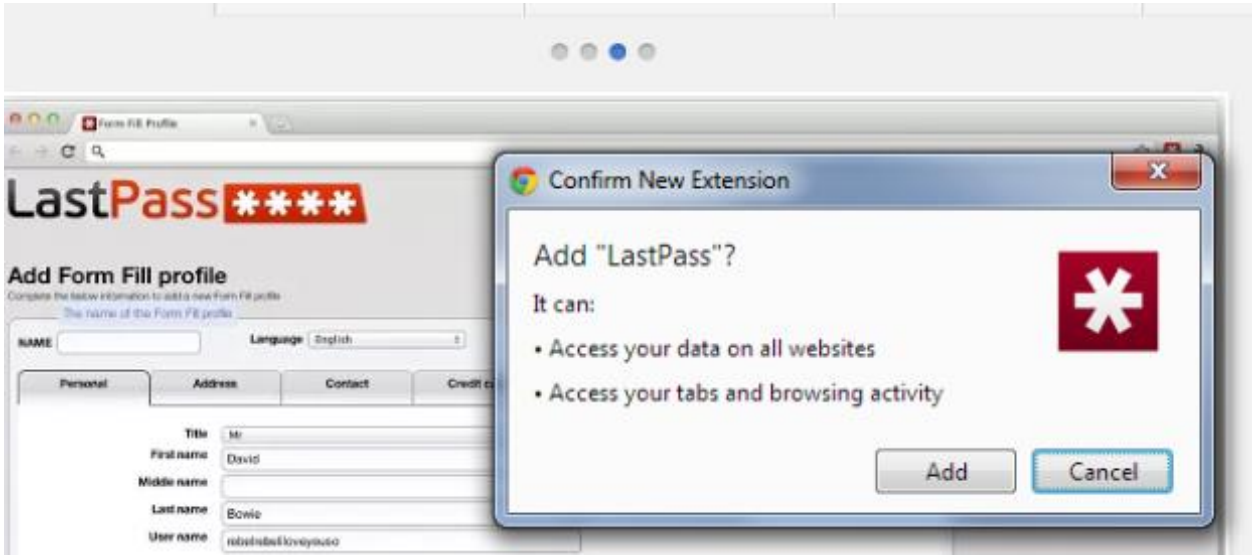
Proširenja uz čitav niz *API-ja* koje pruža *Extension framework* mogu koristiti i standardne *API-je* (poznati kao JavaScript API-ji) koje preglednik omogućuje web stranicama. Neki od podržanih API-ja su: *JavaScript, DOM, XMLHttpRequest, HTML5, WebKit* (za CSS animacije, filtere, itd.) i *V8 (npr. JSON).* HTML5 i drugi API-ji, koje podržava preglednik, omogućuju zvuk, Geo lokaciju, lokalnu pohranu, obavijesti, videozapise i slično.

Korištenjem API-ja mogu se integrirati različite mogućnosti i karakteristike koje se nalaze u pregledniku s proširenjima. Neke od karakteristika su API-ji za alarme, oznake, povijest, kartice, akcije, skladištenje, obavijesti, pretraživanja, itd. [3].

2.5. Sigurnost proširenja

Proširenje je kao bilo koji drugi softver. Instalacija zlonamjernog proširenja može napraviti štetu krajnjem korisniku. Dobra proširenja također mogu imati sigurnosne propuste. Pri izboru proširenja potrebno je odabrati pouzdanog proizvođača [6]. Zlonamjerna proširenja uglavnom se nalaze na web stranicama trećih strana, ali ponekad se mogu naći i na službenim web trgovinama, kao što je bio slučaj s Google Play trgovinom za Android. Nedavno je tim istraživača pronašao četiri proširenja u Chrome web trgovini koja su predstavljala proširenja za bilješke. Oni su zapravo otvaranjem oglasa svojim kreatorima donosili zaradu [7].

Postavlja se pitanje kako takvo proširenje može raditi? Proširenje pri instalaciji traži dozvole kojima korisnik mora dozvoliti rad [7]. Preglednik Chrome daje predodžbu koje dozvole proširenje zahtijeva pri instalaciji. Na taj se način može vidjeti hoće li proširenje raditi samo na jednoj web stranici i hoće li tražiti dodatne dozvole. Kod Firefox-a ne postoji sustav dozvola, već proširenja imaju pristup cijelom pregledniku i više. Internet Explorer ima ograničenu podršku za proširenja [6]. Čak i kod Chrome-a, koji upravlja dozvolama u teoriji, u praksi baš i ne funkcionira. Svako proširenje pri instalaciji traži dopuštenje da pročita i promijeni sve podatke na web stranici koja je posjećena, što ima daje moć da mogu napraviti što god hoće s našim podatcima. Ako im se ne da to dopuštenje, neće se instalirati [7]. Na slici 2.5. je primjer dozvole koju proširenje može tražiti pri instalaciji.



Slika 2.5. Primjer pouzdanog proširenja LastPass [6]

U nastavku se vidi koja dopuštenja proširenja mogu tražiti i njihove potencijalne razine opasnosti [8]:

1. Upozorenje visoke razine
   - Pristup svim podatcima i web-lokacijama koje posjećujete. To znači da to proširenje može pristupiti gotovo svemu (npr. vaša web-kamera, osobne datoteke unutar ili izvan preglednika).
2. Upozorenje srednje razine
   - Pristup podatcima na svim web-lokacijama koje posjećujete. To omogućuje pristup za čitanje, zahtijevanje i izmjenu podataka sa svake stranice koju posjetite (npr. bankovni račun i Facebook).
   - Pristup podatcima na popisu web-lokacija. To omogućuje pristup za čitanje, zahtijevanje ili izmjenu podataka na stranicama s popisa navedenih web-lokacija koje posjećujete.
3. Upozorenje niske razine
   - Pristup popisu instaliranih aplikacija, proširenja i tema. Aplikacija ili proširenje može omogućiti, onemogućiti, deinstalirati ili pokrenuti teme, proširenja i aplikacije koje ste instalirali.
   - Pristup oznakama. Proširenje može čitati, promijeniti, dodati i organizirati vaše oznake.

- Pristup povijesti pregledavanja. Proširenje može čitati i brisati vašu povijest pregledavanja.
- Pristup karticama i aktivnosti pregledavanja. Proširenja mogu vidjeti URL-ove i naslove web-lokacija koje posjećujete, a mogu i otvarati, zatvarati kartice i prozore te prelaziti na nove stranice u otvorenim karticama i prozorima.
- Pristup fizičkoj lokaciji. Proširenja mogu upotrijebiti trenutačnu lokaciju vašeg računala ili uređaja.
- Pristup podatcima koje kopirate i lijepite. Proširenje može pristupiti podatcima koje ste kopirali i zalijepili.

Proširenja su zanimljiva meta za varalice (hakere) jer mnoga imaju velike korisničke baze podataka. Ona se automatski ažuriraju. To znači, ako korisnik preuzme dobro proširenje, ono se može ažurirati i postati zlonamjerno. Ažuriranje će se odmah prenijeti korisniku, a on neće ništa primijetiti. Dobar programer neće napraviti tako nešto, ali njegov račun može biti otet (hakiran) i zlonamjerno ažuriranje učitano u njegovo ime na službenu web trgovinu. Također i dobra proširenja mogu biti opasna zbog sigurnosnih propusta i prikupljanja velikog broja podataka o korisnicima. Programeri mogu prodati anonimne podatke koje su prikupili od trećih strana. Problem je u tome što ponekad podatci nisu dovoljno anonimni i samim time čini se ozbiljna povreda privatnosti korisnika. Stranke, koje otkupe podatke, mogu identificirati korisnike [7].

S proširenjima treba biti posebno na oprezu jer se pokreću u pregledniku. Loše proširenje može pratiti što se pregledava, brojeve kreditnih kartica, lozinke, itd. Stvarni su rizici minimalni ako se odabere proširenje poznatog proizvođača koje koristi puno korisnika. Ipak ih je dobro imati na umu [6].

2.5.1. Sigurno korištenje proširenja

Usprkos činjenici da proširenja mogu biti opasna, neka od njih su vrlo korisna i zato ih vjerojatno i dalje koristimo. Najsigurnije bi bilo da se uopće ne koriste. To je pomalo nezgodno, stoga ih treba koristiti više-manje sigurno.

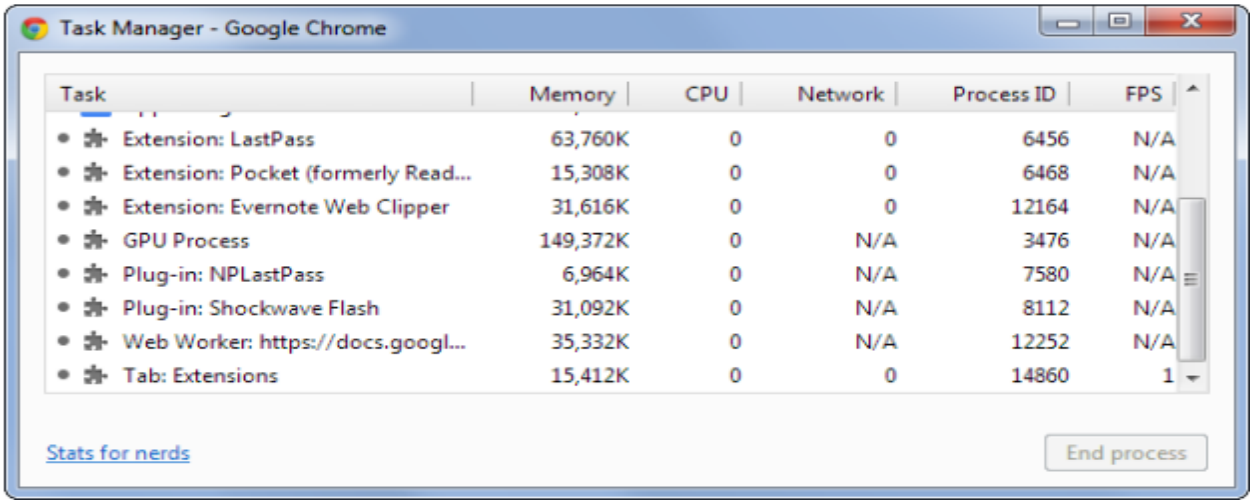
Neki primjeri sigurnog korištenja su [7]:

- Ne treba instalirati previše proširenja. Ne samo da utječu na performanse računala, nego su i potencijalni vektor napada.
- Proširenje se instalira samo sa službene web trgovine. Proširenja, koja se tamo nalaze, prolaze određene kontrole, a stručnjaci za sigurnost filtriraju ona koja su zlonamjerna.
- Treba obratiti pozornost na dozvole koje proširenje zahtijeva. Ako već instalirano proširenje zahtijeva dodatne dozvole, može ujedno biti znakom da je oteto (hakirano) ili prodano. Prije instalacije treba vidjeti koje dozvole proširenje zahtijeva i odgovaraju li ona funkcionalnosti istoga. Ako se ne može pronaći logično objašnjenje dozvole, bolje ga ne instalirati.
- Korištenje antivirusnog programa koji može otkriti i neutralizirati zlonamjerni kod proširenja preglednika.

2.6. Hoće li proširenje usporiti rad preglednika?

Preglednik ne bi trebali preopteretiti proširenjima jer svako je proširenje novi dio koda koji se izvodi na računalu.

Preglednik Chrome proširenja izvodi kao vlastite procese i dodaje novi proces u sustav. Firefox proširenja izvodi u istom procesu, što preglednik može još više usporiti. Korištenje nekoliko proširenja ne bi trebalo utjecati na radne karakteristike, a poboljšat će iskustvo pretraživanja. Proširenja koja se ne koriste treba deinstalirati iz preglednika za ubrzanje rada i smanjivanje resursa sustava. Na slici 2.6. vidi se koliko resursa koristi pojedino proširenje u pregledniku [6].



Slika 2.6. Primjer procesa aplikacije u Task Manageru

2.7. Instaliranje proširenja

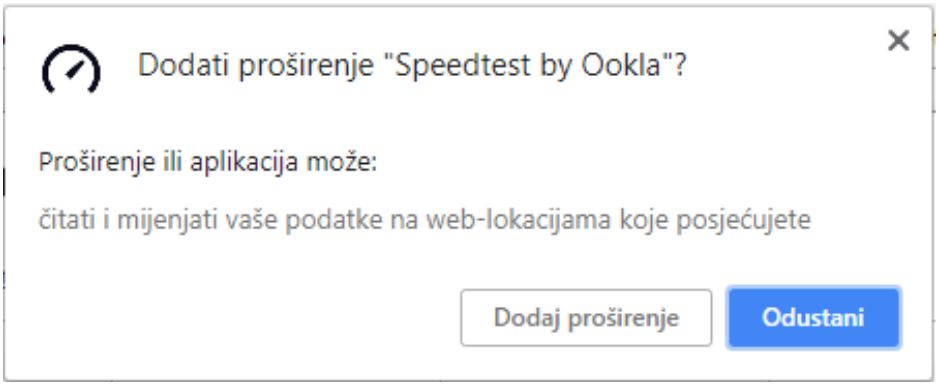
Dodavanje proširenja u preglednik vrlo je jednostavno i potrebno je obaviti samo nekoliko koraka. Na primjer, kod preglednika Chrome proširenja se nalaze u posebnoj Chrome internet trgovini, a pristupa joj se preko URL adrese: https://chrome.google.com/webstore/category/extensions

Koraci pri instalaciji proširenja [8]:

1. Potrebno je pronaći i odabrati *proširenje* koje se želi instalirati,
2. Kliknuti na *Dodaj u Chrome* (slika 2.7),
3. Prikaz obavijesti, ako su za *proširenja* potrebna određena dopuštenja ili podatci. Za dopuštanje dodirnite *Dodaj proširenje* (slika 2.8),
4. Za upotrebu proširenja potrebno je kliknuti na ikonu koja se nalazi s desne strane adresne trake.

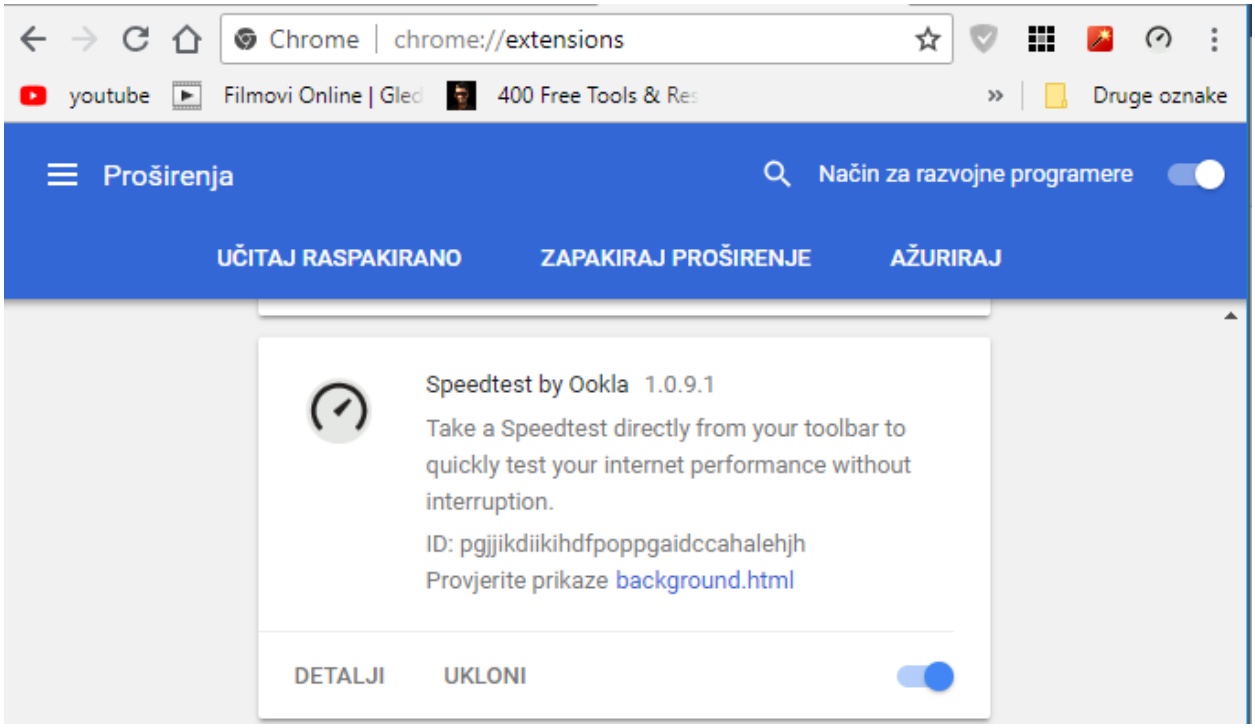
Nakon odabira proširenja instalaciju pokrećemo klikom na *Dodaj u Chrome (Add to Chrome)* kako je prikazano na slici 2.7.

Slika 2.7. Dodavanje proširenja u preglednik



Slika 2.8. Potvrda odabira

Popisu svih proširenja instaliranih u pretraživač pristupa se preko URL adrese: chrome://extensions ili putem $Izbornik > Više\ alata > Proširenja$. Instalirano proširenje nalazi se na popisu kao što je prikazano na slici 2.9. [8].



Slika 2.9. Prikaz dodanog proširenja

3. RAZVOJ QR CODE PROŠIRENJA

QR kod je dvodimenzionalna verzija barkoda. Kreiran je 1994. godine u japanskoj tvornici „Denso Wave“. Naziv *QR* je skraćenica od *Qiuck Respone* (*brzi odgovor*). Izumitelj koda želio je omogućiti kodiranje podataka na velikim brzinama. *QR* kod također koristi nazive *Denso Barcode*, *ISO/IEC8004* i *JIS X 0510*. Na početku se koristio u automobilskoj industriji za praćenje dijelova, a odnedavno postaje sve popularniji. Koristi se u komercijalne svrhe za reklamiranje te u mobilnim aplikacijama koje se zovu mobilno obilježavanje.

QR kodovi s podatcima mogu se pronaći na raznim objektima (npr. automobili, oglasne površine, novine, kartice, prometni znakovi) pomoću kojih korisnik može doći do željene informacije skeniranjem koda. Za čitanje informacija zapisanih u QR kodu korisnik na mobilnom uređaju s kamerom mora imati instaliranu odgovarajuću i ispravnu aplikaciju za čitanje kodova. Također korisnicu imaju mogućnost kreiranja svojih QR kodova na posebnim stranicama za izradu kodova. Ovaj se kod od ostalih kodova ističe s mogućnošću kodiranja raznih vrsta znakova uz *ASCII* znakove (npr. mobilni brojevi, URL adrese, SMS, binarni znakovi i podatci, japanski jezici Kanji i Kana) [9].

Njegova popularnost raste jer koristi tehnologiju otvorenog koda, tj. može ga svatko koristiti. Prednosti su ovoga koda, u odnosu na konvencionalne barkodove, veći kapacitet podataka i visoka tolerancija na pogreške [10].

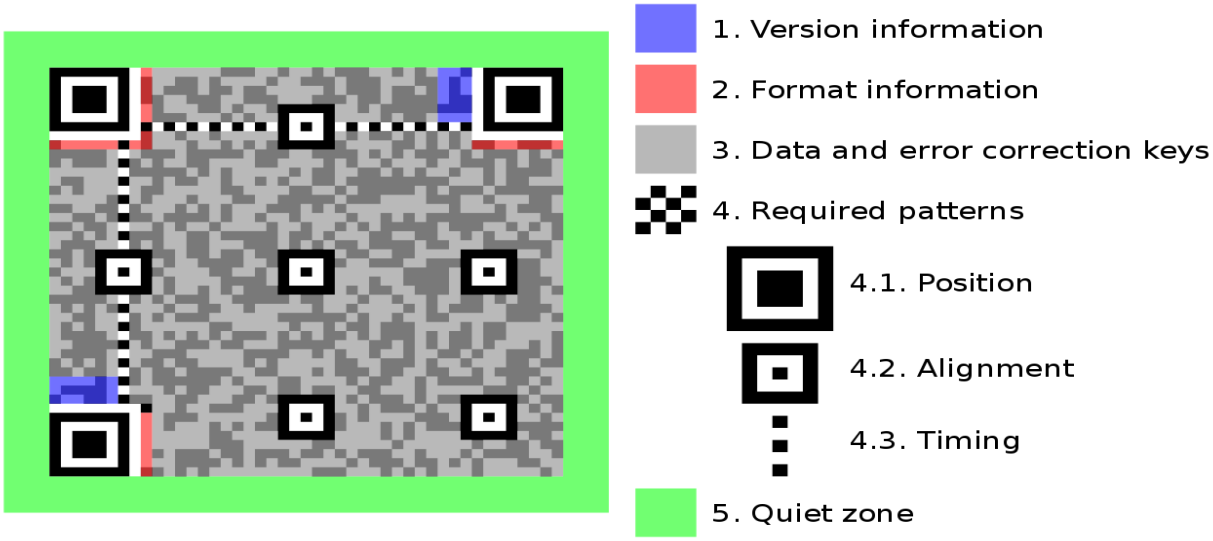
3.1. Struktura QR koda

QR kod izrađen je pomoću uzoraka crno-bijelih piksela koji na prvi pogled izgledaju kao mala križaljka i njihov raspored izgleda kao slučajan. Kada se bolje pogleda, određene strukture mogu se identificirati. QR kod uvijek mora biti pravokutan da ga skener može prepoznati. Elementi prikazani na slici 3.1. osiguravaju ispravno čitanje zapisanih informacija.

Elementi koji to omogućuju su [10]:

1. Informacije o verziji – određuje verziju QR koda koji se koristi. Trenutno postoji 40 različitih verzija dok se u marketinške svrhe koriste verzije od 1 do 7.
2. Informacije o formatu – olakšava skeniranje koda jer sadrži informacije o toleranciji na pogreške i uzorku maske podataka.

3. Polja s podatcima i pogreškama – sadrže stvarne podatke.
4. Potrebni uzorci:
    4.1. Pozicije – označava smjer u kojem se kod ispisuje.
    4.2. Poravnanja – ako je QR kod velik; on pomaže u orijentaciji.
    4.3. Vremena – pomoću ovih linija skener određuje koliko je velika matrica podataka.
5. Mirna zona – omogućuje razlikovanje QR koda od njegove okoline.



Slika 3.1. Struktura QR koda (verzija 7) [12]

3.2. Vrste QR kodova

Postoji više modela QR kodova koji se razlikuju prema izgledu i funkcionalnostima. U nastavku je opisano pet modela i njihove karakteristike.

Modeli:

- **QR kod Model 1 i Model 2**

Model 1 je originalni QR kod koji može pohraniti 1167 znakova sa svojom najvećom verzijom 14 koja ima dimenzije 73x73 modula. Izgled Modela 1 prikazan je na slici 3.2 [11].

Slika 3.2. QR kod Model 1 [12]

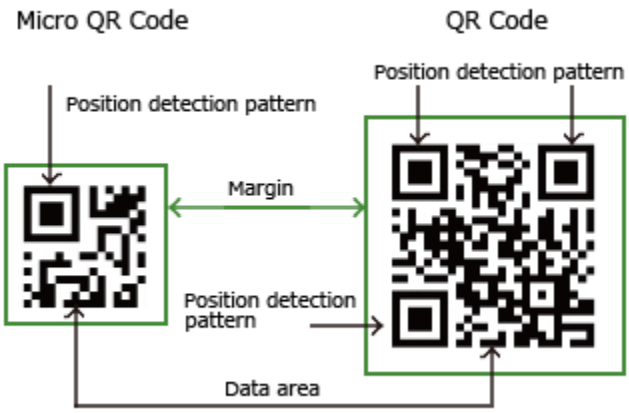
Model 2 je poboljšana verzija Modela 1. Njegova prednost je u lakšem čitanju, čak i ako je kod deformiran. QR kodovi zapisani na zakrivljenoj površini ili čije su slike deformirane za čitanje zbog kuta gledanja mogu se učinkovito očitati pozivajući se na uzorak poravnanja koji je u njih ugrađen. U svojoj najvećoj verziji 40, gdje ima dimenzije 177x177 modula, može pohraniti do 7089 uzoraka. Izgled Modela 2 je prikazan na slici 3.3 [11].

Slika 3.3. QR kod Model 2 [12]

- **Mikro QR kod**

Glavna značajka Mikro QR koda je u tome što ima samo jedan obrazac za otkrivanje položaja u usporedbi sa standardnim QR kodom koji zahtijeva određenu površinu jer su uzorci

otkrivanja smješteni u tri ugla simbola. QR kod zahtijeva barem četiri modula široku marginu oko simbola, dok su kodu Mikro QR koda dovoljna dva. Takva konfiguracija mu omogućuje zapisivanje u područjima manjima od samoga QR koda. Razlika između QR koda i Mikro QR koda prikazana je na slici 3.4.



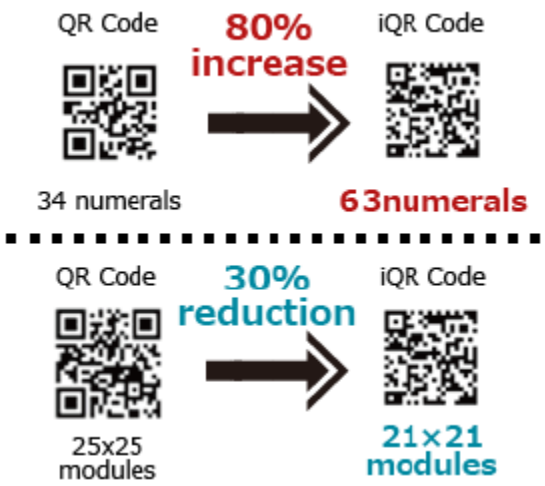
Slika 3.4. Usporedba QR i Mikro QR koda [11]

Količina podataka koja se može pohraniti je manja u odnosu na standardni QR kod i iznosi oko 35 znakova. U usporedbi sa standardnim ova vrsta koda može učinkovitije kodirati podatke. Njegova veličina ne treba biti veća jer se količina pohranjenih podataka povećava. Postoje četiri vrste ovoga koda, od M1 do M4. Maksimalna količina podataka, koja se može kodirati za verziju M4, manja je od one kodirane verzijom 1 QR koda [11].

- **iQR kod**

iQR kod je 2D kod matričnog tipa koji omogućuje jednostavno čitanje položaja i veličine. Omogućuje široki raspon kodova, od malih do standardnih QR i Mikro QR kodova, do velikih koji mogu pohraniti više od tih kodova. Kod se može tiskati u: pravokutnom obliku, kao obrnuti kod, crno-bijeli inverzni kod ili kao točkasti uzorak, što omogućuje široki raspon primjene u raznim područjima.

iQR kod iste veličine kao standardni QR kod može pohraniti 80% više informacija; do 30% manju veličinu ako je pohranjena ista količina podataka. Na slici 3.5. može se vidjeti usporedba u vidu količine pohranjenih podataka i veličine koda [11].



Slika 3.5. Usporedba iQR i standardnog QR koda [11]

S ovom vrstom koda mogu se generirati kodovi koji sadrže puno više informacija, nego što je moguće sa standardnim QR kodom. Sa standardnom verzijom QR koda moguće je kodirati oko 7000 znakova dok je s ovim kodom taj broj puno veći. Najveća verzija koda ima dimenzije 422x422 modula i može pohraniti oko 40 000 znakova. Ovaj kod ima veću sposobnost obnavljanja od standardnog QR koda, što se može vidjeti na slici 3.6. [11].

Slika 3.6. Usporedba razine korekcije [11]

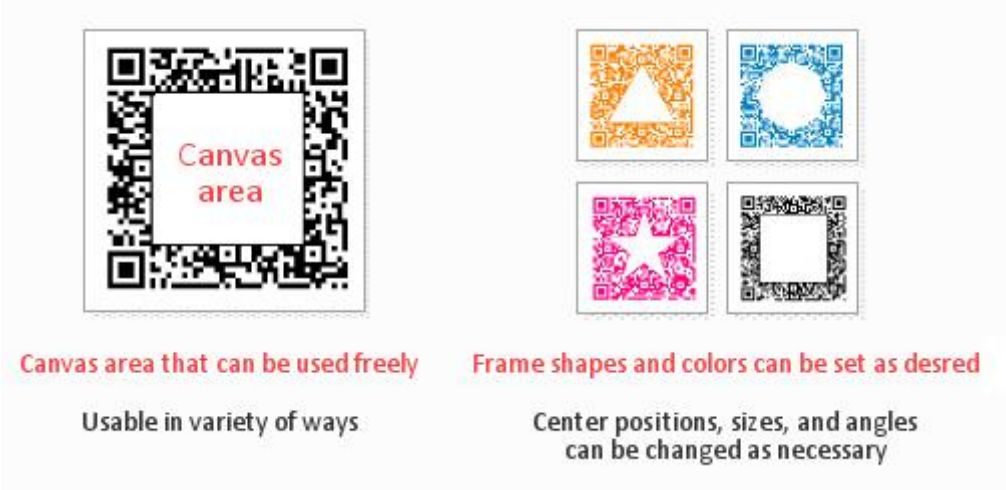
Kod standardnog QR koda razina ispravka pogreške iznosi oko 30% cijelog koda, što znači da je moguće vratiti kod s ovim postotkom oštećenja. Kod iQR koda taj postotak povećava na 50% [11].

- **SQRC**

SQRC je vrsta QR koda koja ima funkciju ograničenja čitanja. Koristi se za pohranu privatnih podataka i za upravljanje internim podatcima neke tvrtke i slično. Može se očitati samo sa specijalnim tipom skenera, ali ta funkcionalnost ne garantira da će kodirani podatci biti sigurni. Podatci pohranjeni ovim kodom sastoje se od javnog i privatnog dijela. Pomoću ovoga koda moguće je pohraniti dvije razine podataka u jednom kodu. SQRC kod ima izgled kao standardni QR kod i zadržava sve funkcionalnosti kao što je ispravljanje pogrešaka [11].

- **Frame QR**

Frame QR kod je QR kod koji u središtu ima područje ili okvir za držanje slike (slikarsko platno). U tome području koda mogu se urediti slike, slova i sl. Time se omogućuje postavljanje koda bez gubitka dizajna ilustracije, slike i sl. kao što je prikazano na slici 3.7. Područje platna ne ometa čitanje kodova dok se veličina i oblik mogu odrediti iz predložaka ili prema želji korisnika [11].



Slika 3.7. Predlošci Frame QR koda

3.3. Pohrana podataka i ispravak pogreške

U jedan QR kod može se unijeti 7089 znamenki ili 4296 znakova, uključujući interpunkcijske i posebne znakove. Osim znakova i brojeva, mogu se kodirati riječi i fraze (npr. web adrese). Kako se u QR kodove dodaje sve više podataka, veličina koda se povećava, a struktura postaje sve složenija [10]. Maksimalni kapacitet znakova, koji se mogu zapisati u QR kod, može se vidjeti u tablici 3.1.

Tablica 3.1. Maksimalni kapacitet pohrane znakova [12]

Način unosa	Maksimalni broj znakova	Bita / znakova	Mogući znakovi, zadano kodiranje
Numerički	7,089	3⅓	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Alfanumerički	4,296	5½	0–9, A–Z (samo velika slova), razmak, $, %, *, +, -, ., /, :
Binarni / bajt	2,953	8	ISO 8859-1
Kanji / kana	1,817	13	Shift JIS X 0208

Kako bi se informacije sadržane unutar QR koda mogle pročitati i u slučaju oštećenja koda, polja se podataka dupliciraju. Prilikom toga se može uništiti i do 30% posto strukture koda bez utjecaja na čitljivost [10]. Kodne riječi su duge 8 bitova i koriste Reed Solomon-ov algoritam za ispravljanje pogrešaka s četiri razine ispravljanja. Što je veća razina ispravljanja pogreške, manji je kapacitet pohrane [12]. U tablici 3.2. nalaze se približne vrijednosti korekcije pogrešaka u svakoj od četiri razine.

Tablica 3.2. Korekcija pogrešaka prema razinama [13]

Razina	Kapacitet korekcije grešaka
L (niska)	7% kodnih riječi / znakova može biti obnovljeno
M (srednja)	15% kodnih riječi / znakova može biti obnovljeno
Q (kvartalna)	25% kodnih riječi / znakova može biti obnovljeno
H (visoka)	30% kodnih riječi / znakova može biti obnovljeno

Reed Solomon-ov algoritam razvili su Irving S. Reed i Gustav Solomon 1960. godine kao djelatnici laboratorija MIT Lincoln. Objasnili su kako prepoznati i popraviti pogreške u slučaju pojave više slučajnih simbola pomoću sustavne nadogradnje kodova. Ovaj kod se koristi u raznim uređajima kao što su: mediji za digitalnu pohranu podataka, modemi (ADSL, XDSL), digitalna i satelitska komunikacija, itd.

Reed Solomon-ov koder prima podatke kao digitalne blokove kojima dodatno dodaje nepotrebne bitove. Pogreške se javljaju u prijenosu ili pohrani iz raznih razloga (npr. šumovi, smetnje, oštećenja na CD-u i sl.). Dekoder pokušava vratiti izvorne podatke tako što obrađuje svaki digitalni blok. Ispravak pogrešaka ovisi o karakteristikama Reed Solomon-ovog koda. Reed Solomon-ov kod je linearni blok kod jer pripada u podskup BHC kodova. Naveden je kao RS (n, k) što znači da koder uzima k bajtova podataka i dodaje paritetne bajte kako bi napravio n bajtova kodne riječi. Dekoder u kodnoj riječi može ispraviti pogreške do t bajta, gdje je $2\mathrm{t} = \mathrm{n} - \mathrm{k}$ [14].

Algebarski postupak Reed Solomon-ovog dekodiranja može ispraviti i obrisati pogrešku. Dekoder koristi brisanje do $2\mathrm{t}$ bajta ako je poznat položaj pogrešnog, a ispravlja pogreške do t bajtova. Moguća su tri rezultata dekodiranja kodne riječi:

1. Ako je $2s + r < t$ (s pogreška, r brisanje); prenesena kodna riječ može se obnoviti,
2. U suprotnom, dekoder ne može vratiti riječ kodu,
3. Ili će vratiti netočnu kodnu riječ.

Vjerojatnost pojave bilo kojega od tri moguća ishoda ovisi o vrsti Reed Solomon-ovog koda, broju i vrsti pogrešaka. Reed Solomon-ov kod temelji se na Galois području matematike ili konačnom polju. Kodna je riječ generirana posebnim polinomom. Sve kodne riječi mogu se dijeliti s polinomom generatora.

Opći je oblik polinoma generatora: $g(x) = (x - \alpha^i)(x - \alpha^{i+1}) \dots (x - \alpha^{i+2t})$

Kodna riječ konstruirana je pomoću $c(x) = g(x).i(x)$, gdje je:

- $g(x)$ polinom generatora
- $i(x)$ blok podataka
- $c(x)$ kodna riječ; zove se primitivni element polja [14].

Kod većih QR kodova poruka je razbijena na nekoliko Reed Solomon-ovih kodnih blokova. Veličina je bloka definirana tako da se unutar svakog bloka može ispraviti najviše 15 pogrešaka zbog ograničavanja složenosti algoritma dekodiranja. Kodni blokovi se međusobno isprepliću, zbog čega je malo vjerojatno da će lokalno oštećenje QR koda nadjačati kapacitet bilo kojeg bloka.

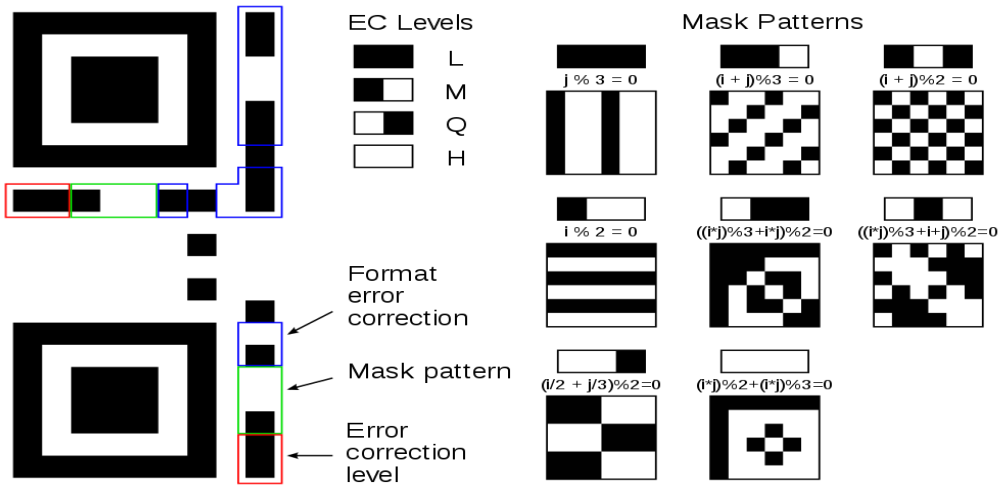
Zbog mogućnosti ispravljanja pogrešaka moguće je dizajnirati umjetničke QR kodove. Oni se mogu ispravno skenirati, iako u sebi sadrže namjerne pogreške kako bi bili čitljiviji i ugodniji ljudskom oku. Mogu se ugraditi različite boje, logotipi i druge značajke unutar bloka QR koda. Slika 3.8. prikazuje ljudskom oko ugodan QR kod koji ispravan, iako sadrži drugu boju i logotip [12].

Slika 3.8. Umjetnički QR kod [12]

3.4. Kodiranje

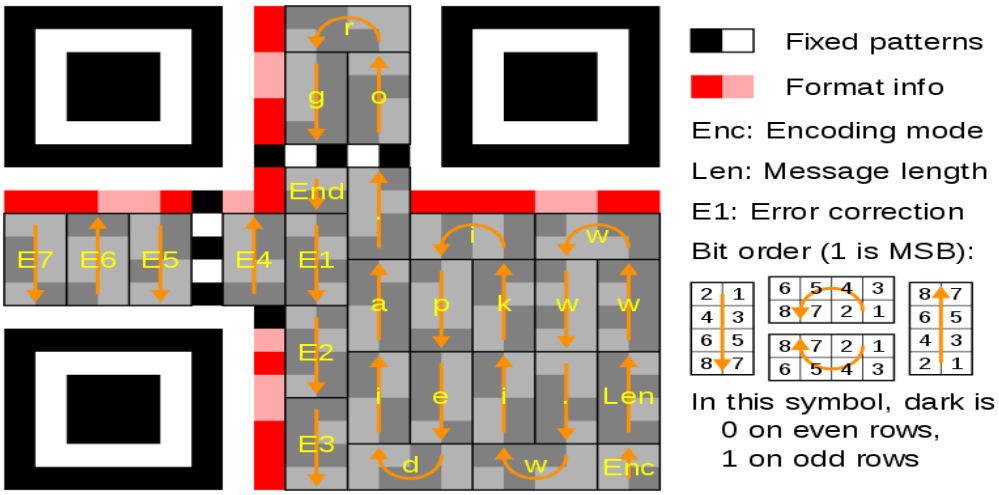
Svaka informacija sadrži dva elementa: razinu ispravljanja pogreške i masku simbola. Maskiranje se koristi za razbijanje uzoraka u području podataka koji mogu zbuniti skener, npr. velika prazna područja ili obmanjujuće značajke koje izgledaju kao oznake za lociranje. Obrazac maske je definiran u rešetki koja se po potrebi ponavlja, kako bi se pokrio cijeli simbol. Moduli, koji odgovaraju tamnim područjima maske, obrnuti su. Informacije o formatu zaštićene su BCH kodom od pogreške, te svaki QR kod ima dvije potpune kopije.

Podatci poruke postavljaju se u cik-cak uzorku s desna na lijevo, kao što je prikazano na slici 3.9. Kodiranje je većih QR kodova komplicirano zbog prisutnosti uzoraka poravnavanja i korištenja višestruko umetnutih blokova za ispravljanje pogreške [12].



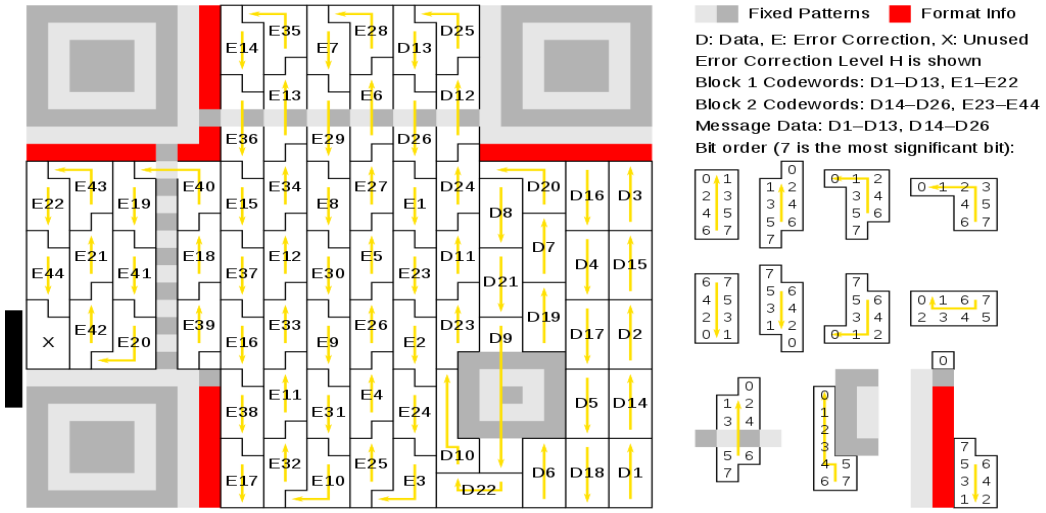
Slika 3.9. Informacije o formatu kodiranja [12]

Na slici 3.9. informacije o formatu su zaštićene BHC kodom (15, 5), koji može ispraviti do 3 bita pogreške. Duljina koda je 15 bita, a broj bitova podataka 5 (EC razina plus obrazac maske). Za ispravljanje pogrešaka koristi se dodatnih 10 bitova. Maska za tih 10 dodatnih bitova je [1001010101] i isključuje fiksne kvadrate uzoraka. Prvih se 5 bitova otkriva.



Slika 3.10. Poruka pretvorena u QR kod [12]

Na slici 3.10. poruka je kodirana pomoću Reed Solomon-ovog koda (255, 249), koji je skraćen umetanjem na kod (24, 18) i može ispraviti do 3 bajta pogrešaka.



Slika 3.11. Uklapanje blokova u veći QR kod [12]

Poruka na slici 3.11. ima 26 bajtova podataka i kodirana je pomoću Reed Solomon-ova koda (255, 211) koji je skraćen na kod (70, 26) i može ispraviti do 22 bajta u jednom nizu. Za ispravljanje pogreške bajtovima podataka dodaju se dodatna 44 paritetna bajta (što rezultira ukupno 70 bajtova koda). Time se postiže ispravljanje pogreške razine H.

Opća struktura QR kodiranja je kao niz 4-bitna indikatora i dužina niza ovisi o modelu indikatora (npr. duljina kodiranja bajta ovisi o prvom bajtu). Četvero bitni indikatori koriste se za odabir kodiranja i prijenos informacija. Opća struktura i modeli kodiranja nalaze se u tablici 3.3. [12].

Tablica 3.3. Broj bitova i indikatora u karakteru QR koda [12]

Vrsta indikatora	**Opis**	**Tipična struktura '[vrsta : veličina u bitovima]'**
0001	Numeričko kodiranje (10 bita po 3 znamenke)	[0001 : 4] [Pokazatelj broja znakova: varijabla] [Stream podataka : 10 × charcount]
0010	Alfanumeričko kodiranje (11 bita po 2 znaka)	[0010 : 4] [Pokazatelj broja znakova: varijabla] [Stream podataka : 11 × charcount]
0100	Kodiranje bajtova (8 bita po znaku)	[0100 : 4] [Pokazatelj broja znakova: varijabla] [Stream podataka : 8 × charcount]
1000	Kanji kodiranje (13 bita po znaku)	[1000 : 4] [Pokazatelj broja znakova: varijabla] [Stream podataka : 13 × charcount]
0011	Strukturirani dodatak (koristi se dijeljenje poruke na više QR simbola)	[0011 : 4] [Pozicija simbola : 4] [Ukupno simbola: 4] [Paritet : 8]
0111	Interpretacija proširenog kanala (odaberite alternativni skup znakova ili kodiranja)	[0111 : 4] [ECI broj zadatka : varijabla]
0101	FNC1 na prvoj poziciji (pogledaj kod 128 za dodatne informacije)	[0101 : 4] [Numerička / Alfanumerička / Bajt / Kanji nosivost : varijabla]
1001	FNC1 na drugoj poziciji	[1001 : 4] [Pokazatelj primjene: 8] Numerička / Alfanumerička / Bajt / Kanji nosivost : varijabla]
0000	Kraj poruke (Terminator)	[0000 : 4]

Vrste kodiranja unutar QR koda mogu se miješati (npr. URL s dugim nizom alfanumeričkih znakova). Sintaksa:

```
[Indikator načina rada][Vrsta bitstreama]-->[Indikator načina
    rada][ Vrsta bitstreama]--> etc.   > [0000 Kraj poruke
                        (Terminator)]
```

Svaki indikator može imati drugi oblik kodiranja. U svakom se polju nalazi onoliko znakova koliko ih je kodirano tim načinom. Broj bitova u polju ovisi o vrsti kodiranja i simbolu (tablica 3.4.) [12].

Tablica 3.4. Broj bitova u polju [12]

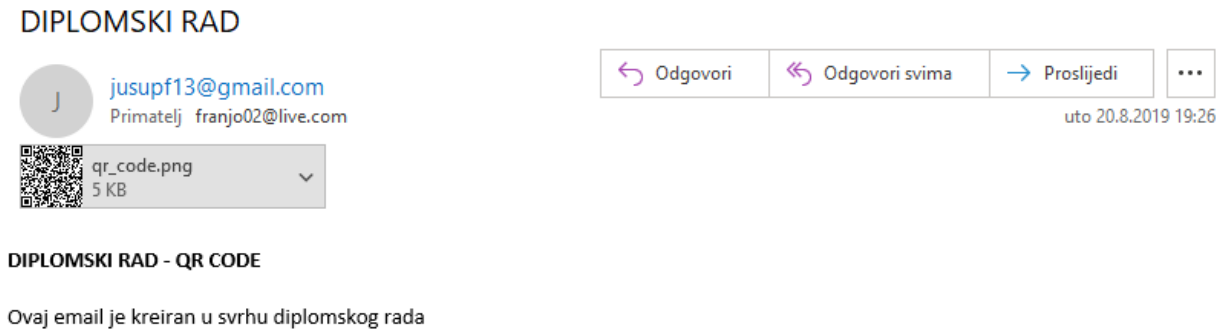
Kodiranje	Verzija 1–9	Verzija 10–26	Verzija 27–40
Numerički	10	12	14
Alfanumerički	9	11	13
Bajt	8	16	16
Kanji	8	10	12

3.5. Razvijeno proširenje

Razvijeno proširenje omogućuje korisnicima kreiranje QR koda iz URL adrese web stranice koja je trenutno otvorena u kartici preglednika. Za kreiranje QR koda potrebno je kliknuti na ikonu proširenja koja se nalazi u alatnoj traci. Pokretanjem proširenja odmah se generira slika s QR kodom web stranice koja je otvorena u trenutnoj kartici preglednika, što je prikazano na slici 3.12. Prelaskom strelice miša preko slike QR koda može se vidjeti koja je URL adresa stranice zapisana u sliku. Korisnik ima mogućnost iz aplikacije poslati sliku QR koda na adresu e-pošte.

Slika 3.12. Izgled QR kod proširenja

Na slici 3.13. prikazan je primjer e-poruke koju će dobiti primatelj kada mu se pošalje slika QR koda iz proširenja. Prije skeniranja QR koda sliku je potrebno otvoriti unutar e-pošte ili skinuti na računalo. Potrebno imati instaliranu aplikaciju za skeniranje QR koda na mobitelu ili računalu da bismo otvorili web stranicu koja je zapisana unutar slike.



Slika 3.13. E-poruka s QR kodom

3.6. Opis korištenih tehnologija i alata

HTML

HTML je prezentacijski jezik osmišljen za izradu web stranica koje poslije može svatko pregledavati tko ima pristup internetu. Relativno je jednostavan za naučiti i poprilično moćan u onome što može sve napraviti. Neprestano se vrši revizija i evolucija kako bi se udovoljilo zahtjevima i potrebama internet korisnika, a obavlja ju organizacija W3C.

Definicija HTML-a je *HyperText Markup Language*.

- *HyperText* je metoda kojom se krećemo po webu; klikom na poseban tekst (*hiperlink*) omogućuje nam se odlazak na sljedeću stranicu .
- *Markup* (označavanje) je ono što HTML oznake čine tekstu, odnosno označavaju ga kao određenu vrstu teksta (npr. podebljavanje teksta).
- *HTML* je jezik koji kao i svaki drugi jezik ima svoje kodne riječi i sintaksu.

HTML se sastoji od niza kratkih kodova koje je autor web stranice napisao u tekstualnu datoteku, odnosno to su oznake. Tekst se sprema u obliku HTML datoteke i pregledava putem preglednika. Preglednik čita datoteku i prevodi tekst u vidljivi oblik, stoga je pri pisanju potrebno pravilno koristiti oznake kako bi tekst mogao biti preveden u vidljiv oblik [15].

```
<body>
    <div class="modal-header">
        <h1 class="logo">
            <img src="images/QR.png" alt="QR simbol"
class="logo-icon"> QR generator
            <span class="version">(1.0.0)</span>
        </h1>
    </div>

    <div class="modal-content">
        <div id="qrimg"></div>
    </div>
```

Programski kod 3.1. Dio koda za strukturu *proširenja*

CSS

CSS je jednostavan stilski jezik kojemu je cilj pojednostaviti postupak prezentacije web stranice. Pomoću njega se uređuje izgled web stranice. Mogu se kontrolirati boje tekstova, stil fontova, razmak između odlomaka, veličina i raspored stupaca, pozadinske slike ili boje, varijacija prikaza za različite uređaje i veličine zaslona. CSS je jednostavan za učenje i razumijevanje. Ujedno omogućuje snažnu kontrolu prezentacije HTML dokumenta. Najčešće se kombinira s jezicima za označavanje kao što su: HTML ili XHTML [16].

Prednosti CSS-a [16]:

- štedi vrijeme
- stranice se brže učitavaju
- jednostavnije održavanje
- superiorniji stilovi za HTML
- kompatibilnost na više vrsta uređaja
- globalni web standardi.

```
html,
body {
    font-family: "Open Sans", sans-serif;
    font-size: 14px;
    margin: 0;
    min-height: 180px;
    padding: 0;
    width: 330px;
}

h1 {
    font-family: "Menlo", monospace;
    font-size: 22px;
    font-weight: 400;
    margin: 1;
    color: #2f5876;
}
```

Programski kod 3.2. Dio koda za dizajn *proširenja*

JavaScript

JavaScript je jednostavan, interpretiran programski jezik namijenjen razvoju interaktivnih HTML stranica. Jezgra JavaScripta uključena je u većinu današnjih preglednika. Omogućuje izvršavanje određenih radnji u statičkim HTML dokumentima kao što su interakcija s korisnikom, promjena svojstva prozora preglednika ili dinamičko stvaranje HTML sadržaja. Za standardizaciju skriptnih jezika zadužena je organizacija ECMA. Program koji obrađuje i izvršava skripte zove se interpreter. Interpreter čita kod i prevodi ga u strojni jezik svaki put kada se pokrene skripta. Svaki jezik koji se interpretira, tj. koji izvršava interpreter, zove se skriptni jezik. Interpreter za JavaScript ugrađen je u većinu današnjih preglednika. Skriptni programi koriste se za razvoj programa zato što su jednostavniji; kod njih se skripte ne trebaju prevoditi u strojni jezik [17].

```
// dohvaćanje trenutnog URL-a
function getUrl(callback) {
    // info o trenutnoj kartici
    var queryInfo = {
        currentWindow: true,
        active: true,
    };

    chrome.tabs.query(queryInfo, function(cards) {
        var card = cards[0];
        var url = card.url;

        callback(url);
    });
}
```

Programski kod 3.3. Dio koda za dohvat URL kartice

Za generiranje slike QR koda iz URL adrese korišten je open source kod koji se može naći na *GitHub linku*.

Node.JS

Node.JS je otvorenog koda i više platformsko *runtime* okruženje koje omogućuje izvršavanje JavaScript koda izvan web preglednika. Popularan je za izradu svih vrsta projekata. Za implementiranje i izvršavanje koda koristi Google-ov *V8 JavaScript engine*. Pokreće se u jednom procesu bez stvaranja novih niti za svaki zahtjev. Temelji se na događajima i izvodi se asinkrono. Kada izvršava I/O operaciju, poput čitanja s mreže, pristupa bazi podataka ili sustavu datoteka. Umjesto da blokira nit i troši CPU, on će nastaviti s radom kada dođe odgovor. To mu omogućuje obradu tisuće istodobnih veza s jednim poslužiteljem bez upravitelja za niti [18].

Značajke Node.JS-a:

- ekstremna brzina
- I/O operacije su asinkrone i vođen je događajima
- skalabilni web servisi
- biblioteke otvorenog koda

```
const express = require("express");
const app = express();

const PORT = process.env.PORT || 3000;

let server = app.listen(PORT, function() {
    console.log("Server started at http://localhost:%s",
PORT);
});
```

Programski kod 3.4. Dio koda za pokretanje servera

Nodemailer

Nodemailer je modul za Node.JS aplikacije koji omogućuje jednostavnije slanje e-mail poruka. Projekt je započeo 2010. godine kada nije postojala nijedna mogućnost slanja e-mail poruka, a danas je to standardno rješenje kojem se okreću korisnici Node.JS-a. Licenciran je pod licencom MIT-a [19].

Značajke Nodemailer-a [19]:

- fokusiranje na sigurnost
- podrška za Windows (može se instalirati s NPM-om na Windows, kao i bilo koji drugi modul)
- dodavanje privitaka porukama
- rad s HTML sadržajem
- sigurna isporuka e-poruka pomoću TLS/STARTTLS-a
- ugrađena SMTP podrška

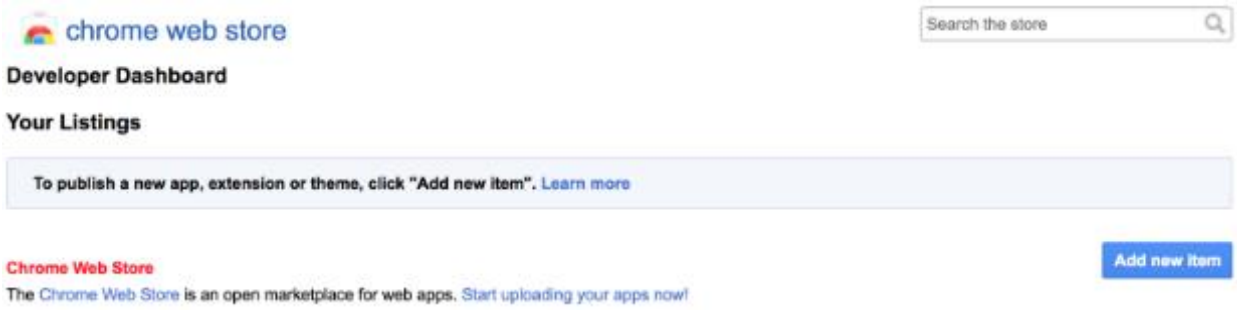
```
const nodeMailer = require('nodemailer');
app.post('/send-email', function(req, res) {

    let transporter = nodeMailer.createTransport({
        host: 'smtp.gmail.com',
        port: 465,
        secure: true,
        auth: {
            user: 'email',
            pass: 'password'
        }
    });
```

Programski kod 3.5. Dio koda za slanje e-poruke

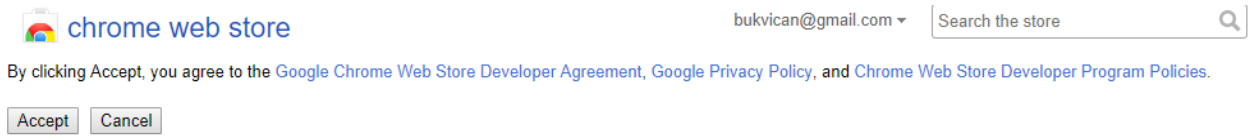
4. KOMERCIJALIZACIJA

4.1. Objava proširenja

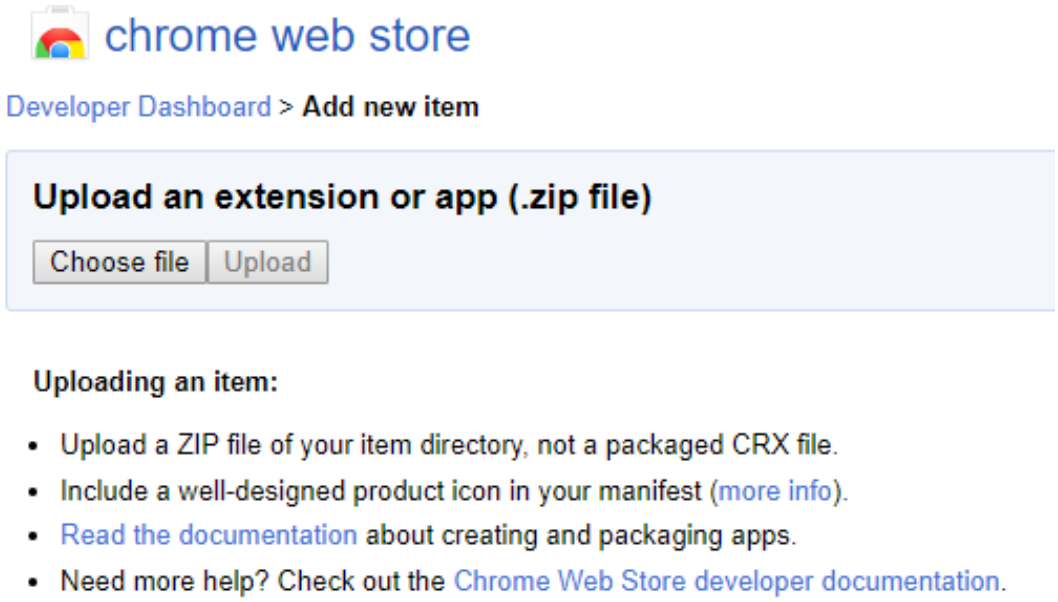
Kada je proširenje za preglednik gotovo i radi kako je zamišljeno (napravljene sve izmjene dizajna i dodane sve funkcionalnosti), može se objaviti u Chrome web trgovini. Nadzorna ploča Chrome web trgovine nalazi se na *linku* (potrebna prijava Google računom). Za objavu prvog proširenja potrebno je jednokratno platiti 5 dolara za registraciju programera i provjeru računa. Novo se proširenje dodaje klikom na gumb *Add new item* (slika 4.1.). Potrebno je prihvatiti uvjete za objavljivanje (slika 4.2.) nakon čega se prikazuje sučelje u koje se učitava proširenje (slika 4.3.). Mapa u kojoj se nalazi projekt mora biti komprimirana u ZIP datoteku.



Slika 4.1. Dodavanje novog proširenja



Slika 4.2. Prihvaćanje uvjeta



Slika 4.3. Sučelje za učitavanje *proširenja*

Nakon uspješnog učitavanja datoteke treba dodati informacije o proširenju u ponuđeni obrazac. Može se dodati ikona, detaljan opis, snimke proširenja, itd. U opis treba dodati slike proširenja koje trgovina može upotrijebiti za promociju. Za isticanje proširenja potrebno je dodati što više slika koje ga prikazuju. Prije objave moguće je pogledati kako proširenje izgleda unutar web trgovine klikom na gumb *Pregled promjena*. *Proširenje* se objavljuje klikom na *Objavite izmjene*.

Nakon objave proširenje se može naći u web trgovini preko imena ili u kategoriji u kojoj je objavljeno. Nekada mora proći i do 24 sata od objave da bi proširenje bilo vidljivo. Razlog je tome što web trgovina provjerava određene sigurnosne aspekte proširenja. Proširenje je moguće podijeliti na društvenim mrežama i među prijateljima kako bi porastao broj njegovih korisnika. Također, proširenje je dobro dodati na stanicu Product Hunt [20]. Product Hunt je stranica koja omogućuje korisnicima dijeljenje i otkrivanje novih proizvoda.

5. ZAKLJUČAK

Cilj je ovoga rada bio objasniti što su to proširenja za web preglednik, što je sve potrebno za njegovu izradu i kako ga komercijalizirati. Pri razvoju potrebno je poznavati arhitekturu proširenja i programske jezike u kojima se može napraviti. Arhitektura je proširenja vrlo bitna zbog testiranja i komercijalizacije jer ako nije napravljeno prema pravilima, proširenje je neupotrebljivo. Izrada je proširenja za web preglednik vrlo jednostavna, ali da bi ono imalo dobar dizajn, željene funkcionalnosti i bilo vrijedno, potrebno je uložiti dosta vremena. Za izradu potrebno je znati raditi s HTML-om, CSS-om, JavaScript-om i kako dodati funkcionalnosti putem JavaScript API-a koje Chrome omogućuje. U slučaju pojave problema prilikom izrade proširenja, rješenje je moguće naći u Chrome dokumentaciji za proširenja, koja su vrlo dobro napisana. Za objavu proširenja u web trgovini potrebno je imati Google račun i slijediti pravila po kojima se proširenje može učitati.

Proširenja je za web preglednike svakim danom sve više jer povećavaju mogućnosti preglednika, što korisnicima olakšava svakodnevno korištenje. Skidanje i instaliranje proširenja s interneta je brzo i jednostavno, a njihova svrha može biti od dugoročne ili kratkoročne pomoći korisniku. Prilikom odabira proširenja treba odabrati pouzdanog razvojnog programera jer, osim što olakšavaju korisniku pretraživanje, mogu narušiti njegovu privatnost i sigurnost. Svako proširenje traži određene dozvole prilikom instaliranja bez kojih ono ne može raditi. Davanjem dozvole omogućujemo praćenje što se pregledava, spremanje brojeva kreditnih kartica, lozinki i slično.

U zadnjih par godina raste popularnost QR koda zbog brze čitljivosti i velikih mogućnosti spremanja podataka, pa je u radu napravljeno proširenje koje generira QR kod iz URL adrese web stranice. Kada se generira QR kod, moguće ga je poslati e-poštom kao sliku. Primatelj, da bi vidio što je zapisano, mora skenirati kod pomoću mobitela ili imati instaliranu aplikaciju za čitanje QR kodova. QR kod može biti i do 50% oštećen, a da se iz njega mogu pročitati zapisane informacije.

LITERATURA

[1] The Google Chrome Team, *Foreword to 20 things*, listopad 2010.

[2] Lawrence Abrams, *What are Google Chrome Extensions?*, https://www.bleepingcomputer.com/tutorials/understanding-google-chrome-extensions/, pristupljeno 28. srpnja 2019.

[3] Prateek Mehta, *Creating Google Chrome Extensions*, Apress, 2016.

[4] *Browser extension*, https://en.m.wikipedia.org/wiki/Browser_extension?fbclid=IwAR1MepiNIJ4_xrOlNxWJYK4y dbDIiE_2 JvPEH0m5dBhbpAUzH0FASiNpQdY, pristupljeno 22. srpnja 2019.

[5] Krasimir Tsonev, *Developing Google Chrome Extensions*, https://code.tutsplus.com/tutorials/developing-google-chrome-extensions--net-33076, pristupljeno 30. srpnja 2019.

[6] Chris Hoffman, *Beginner Geek: Everything You Need To Know About Browser Extensions*, https://www.howtogeek.com/169080/beginner-geek-everything-you-need-to-know-about-browser-extensions/, pristupljeno 28. srpnja 2019

[7] Perekalin Alex, Why you should be careful with browser extensions, https://www.kaspersky.com/blog/browser-extensions-security/20886/, pristupljeno 28. kolovoza 2019.

[8] *Dopuštenja koja zahtijevaju aplikacije i proširenja*, https://support.google.com/chrome webstore/answer/186213?hl=hr&ref_topic=6238977, pristupljeno 15. kolovoza 2019.

[9] *2D kodovi*, http://materijali.grf.unizg.hr/media/2D%20kodovi.pdf, pristupljeno 18. kolovoza 2019.

[10] *QR Code Basics*, https://www.qr-code-generator.com/qr-code-marketing/qr-codes-basics/, pristupljeno 22. srpnja 2019.

[11] *Types of QR Code*, https://www.qrcode.com/en/codes/, pristupljeno 20. kolovoza 2019.

[12] *QR code*, https://en.wikipedia.org/wiki/QR_code, pristupljeno 22. srpnja 2019.

[13] *Data Encoding*, https://www.thonky.com/qr-code-tutorial/data-encoding, pristupljeno 25. srpnja 2019.

[14] Kristina Janko, *Projektiranje individualnog QR koda*, https://eprints.grf.unizg.hr/1605/1/DB271_Janko_Kristina.pdf, pristupljeno 18. kolovoza 2019.

[15] Ross Shannon, *What is HTML?*, https://www.yourhtmlsource.com/starthere/whatishtml.html, pristupljeno 20. kolovoza 2019.

[16] *What is CSS?*, https://www.tutorialspoint.com/css/what_is_css.htm, pristupljeno 20. kolovoza 2019

[17] Denis Stančer, *Osnove JavaScripta*, https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c501_polaznik.pdf, pristupljeno 20. kolovoza 2019.

[18] *Introduction to Node.JS*, https://nodejs.dev/, pristupljeno 20. kolovoza 2019.

[19] *Nodemailer*, https://nodemailer.com/about/, pristupljeno 20. kolovoza 2019.

[20] Jake Prins, *How to Create and Publish a Chrome Extension in 20 minutes*, https://www.freecodecamp.org/news/how-to-create-and-publish-a-chrome-extension-in-20-minutes-6dc8395d7153/, pristupljeno 28. srpnja 2019.

SAŽETAK

Ovaj rad definira što je proširenje web preglednika, kako ga dodati web pregledniku te koje su prednosti i nedostatci njegove uporabe. Ovaj rad također opisuje cjelokupni proces razvoja, aspekte sigurnosti i elemente arhitekture od kojih se sastoji. Praktični dio opisuje alate i programski jezik koji se koristi za razvoj. Cijeli razvojni proces prikazan je razvojem generatora QR koda. Teoretski je opisana struktura QR koda, njegovi elementi, funkcionalnosti i pravila za njegovo stvaranje. Posljednji dio opisuje komercijalizaciju proširenja web preglednika.

Ključne riječi: arhitektura, Chrome, preglednik, proširenje, QR

ABSTRACT

Web browser plugin development

This paper defines what is web browser extension, how to add it to the web browser and what are advantages and disadvantages of using it. Also this paper describes the whole process of development, security aspects and elements of architecutre of which it consists. Pracital part describes tools and programing language used for development. The entire development process is shown through the develop of a QR code generator. The structure of the QR code, its elements, functionalities and rules for creating it are theoretically described. Last part describes commercialization of web browser extension.

Keywords: arhitecture, Chrome, browser, extensions, QR

ŽIVOTOPIS

Franjo Jusup, rođen 20.9.1993. godine u Našicama, Hrvatska. Osnovnu školu „Antun Gustav Matoš“ završava u Čačincima odličnim uspjehom, a Srednju školu „Marka Marulića“ (smjer elektrotehnike) u Slatini vrlo dobrim uspjehom. Tijekom srednjoškolskog obrazovanja odradio je praksu u HEP ODS d.o.o. u Orahovici. Na sveučilištu Josipa Juraja Strossmayera u Osijeku upisuje Elektrotehnički fakultet, Stručni studij Informatika 2012. godine. Tijekom studiranja odradio je praksu na Elektrotehničkom fakultetu u Osijeku, na Zavodu za elektrostrojarstvo. Po završetku stručnog studija Informatike 2015. godine upisuje razlikovnu godinu smjer Računarstvo za stjecanje uvjeta za upis na diplomski studij Računarstva. Diplomski studij Računarstva, smjer podatkovne i informacijske znanosti, upisuje 2016. godine nakon završetka razlikovne godine.

Potpis

PRILOZI

U prilogu diplomskog rada nalazi se optički medij na kojemu je pohranjen rad u .docx i .pdf formatu, te programski kod razvijenog rješenja.