**UNIVERZA V MARIBORU**

**FAKULTETA ZA ELEKTROTEHNIKO,**

**RAČUNALNIŠTVO IN INFORMATIKO**

**SVELTE**

Seminarska naloga

Avtorji:

David Novak

Matej Klemenčič

Mark Hajdinjak

Maribor, maj 2024

Vsebina

[1. UVOD 4](#_Toc165568788)

[2. ZGODOVINA 5](#_Toc165568789)

[3. SPLETNE STRANI Z UPORABO SVELTE 6](#_Toc165568790)

[4.1. Rakuten 6](#_Toc165568791)

[4.2. The New York Times 7](#_Toc165568792)

[4.3. 1Password 7](#_Toc165568793)

[4.4. NameCoach 8](#_Toc165568794)

[4.5. Creative Tim 9](#_Toc165568795)

[4.6. GoDaddy 10](#_Toc165568796)

[4. MINIMALNE SISTEMSKE ZAHTEVE ZA ZAGON APLIKACIJE 11](#_Toc165568797)

[5. NAVODILA ZA NAMESTITEV IN UPORABO 12](#_Toc165568798)

[6.1. GLAVNE ZNAČILNOSTI URADNE DOKUMENTACIJE SVELTE 13](#_Toc165568799)

[6. Prikaz nekaterih izbranih primerov uporabe 14](#_Toc165568800)

[7. PREDNOSTI IN SLABOSTI SVELTE 19](#_Toc165568801)

[8.1. PREDNOSTI 19](#_Toc165568802)

[8.2. SLABOSTI 20](#_Toc165568803)

[8. PRIMERJALNA ANALIZA SVELTE Z DRUGIMI ORODJI 21](#_Toc165568804)

[9.1 Svelte vs React 21](#_Toc165568805)

[**ARHITEKTURA** 21](#_Toc165568806)

[**UČENJE IN UPORABNOST** 21](#_Toc165568807)

[**SKUPNOST IN EKOSISTEM** 21](#_Toc165568808)

[9.2. Svelte vs Angular 22](#_Toc165568809)

[**CELOVITOST REŠITVE** 22](#_Toc165568810)

[**TESTIRANJE IN RAZVOJ** 22](#_Toc165568811)

[9.3. Svelte vs Vue 22](#_Toc165568812)

[**SINTAKSA IN RAZVOJNA IZKUŠNJA** 22](#_Toc165568813)

[**REAKTIVNOST** 23](#_Toc165568814)

[9. Aplikacija 24](#_Toc165568815)

[10.1 Cinematrix aplikacija 24](#_Toc165568816)

[10.2 OMDb API 24](#_Toc165568817)

[10.3 FUNKCIONALNOSTI 25](#_Toc165568818)

[**Registracija** 25](#_Toc165568819)

[**Login** 25](#_Toc165568820)

[**Movies** 26](#_Toc165568821)

[**Logout** 26](#_Toc165568822)

[10. ZAKLJUČEK 27](#_Toc165568823)

[11. VIRI 28](#_Toc165568824)

Kazalo slik:

[Figure 1:Svelte 5](#_Toc165568658)

[Figure 2:Rakuten 7](#_Toc165568659)

[Figure 3:The New York Times 8](#_Toc165568660)

[Figure 4:1Password 8](#_Toc165568661)

[Figure 5:NameCouch 8](#_Toc165568662)

[Figure 6:Creative Tim 10](#_Toc165568663)

[Figure 7:Go Daddy 11](#_Toc165568664)

[Figure 8:Izsek kode za ustvarjanje novega svelte projekta 12](#_Toc165568665)

[Figure 9:Izsek kode 14](#_Toc165568666)

[Figure 10:Izsek kode za integracijo dinamičnih komponent 15](#_Toc165568667)

[Figure 11:Izsek kode za reaktivno posodabljanje UI 15](#_Toc165568668)

[Figure 12:Izsek kode za izdelavo komponent 16](#_Toc165568669)

[Figure 13:Izsek kode za upravljanje stanja 16](#_Toc165568670)

[Figure 14:Izsek kode za navigacijo 17](#_Toc165568671)

[Figure 15:Izsek kode za dinamično nalaganje podatkov 18](#_Toc165568672)

[Figure 16:Izsek kode za animacije 18](#_Toc165568673)

[Figure 17:Cinematrix Aplikacija 24](#_Toc165568674)

[Figure 18:Registracija 25](#_Toc165568675)

[Figure 19:Login 26](#_Toc165568676)

[Figure 20:Movies 26](#_Toc165568677)

# UVOD

Svelte je inovativno JavaScript ogrodje, ki se osredotoča na izdelavo učinkovitih spletnih aplikacij. Značilnost Svelte je, da večji del procesiranja izvede že med samo gradnjo aplikacije, namesto da bi se to izvedlo v brskalniku uporabnika. To omogoča hitrejše delovanje same končne aplikacije. Svelte uporabnikom omogoča razvoj komponent neposredno z uporabo HTML, CSS in JavaScript. S tem poenostavi razvojni proces in zmanjša potrebo po dodatnih orodjih. Poleg svojih tehničnih prednosti, Svelte izstopa zaradi močne in aktivne skupnosti razvijalcev, ki nenehno prispevajo k izboljšanju ogrodja. Prav tako je zaradi svoje preprostosti in učinkovitosti postalo priljubljeno med razvijalci, ki želijo zgraditi spletne aplikacije z manj zapletenosti in boljšo zmogljivostjo. To naredi Svelte odlično izbiro za projekte, ki potrebujejo visoko učinkovitost brez dodatnega bremena velikih ogrodij.



Figure 1:Svelte

# ZGODOVINA

Svelte je bil ustvarjen leta 2016 s strani Richa Harrisa kot poskus, kako izboljšati razvoj spletnih aplikacij z manjšim bremenom tradicionalnih ogrodij. Rich Harris, delovnih novinar in razvijalec, je želel poenostaviti proces razvoja, tako da bi odstranil zapletene abstrakcije in neposredno manipuliral s spletom DOM. S pristopom, ki odpravlja potrebo po virtualnem DOM-u, je Svelte hitro pridobil priljubljenost in priznanje v skupini razvijalcev.

Od svoje uvedbe je Svelte doživel več večjih izdaj, vsaka s pomembnimi izboljšavami v zmogljivosti in uporabniški izkušnji. S svojim edinstvenim pristopom “compile-time” delovanja. Svelte je vzbudil zanimanje številnih razvijalcev, ki iščejo učinkovitejše alternative z JavaScript ogrodjem. Prav tako so se pojavile številne priljubljene knjižnice in vtičniki, ki razširjajo funkcionalnost Svelte, kar priča o njegovi rasti in vse večji uporabi v različnih projektih in aplikacijah.

Svelte je tudi spodbudil razpravo o prihodnjih spletnih ogrodij in razvoja aplikacij. Z naraščajočim zanimanjem za lažje, bolj zmogljive tehnologije, se je Svelte uveljavil kot pomemben produkt v tehnološki skupnosti, ki postavlja nove standarde za razvoj spletnih aplikacij. Njegova sposobnost, da združi hitrost, preprostost in zmogljivost, mu je omogočila, da ostane relevanten in konkurenčen v hitro spreminjajočem se okolju.

# SPLETNE STRANI Z UPORABO SVELTE

## 4.1. Rakuten

Rakuten je uvrščen med podjetja, ki uporabljajo Svelte za razvoj spletnih aplikacij. To bi lahko vključevalo uporabo Svelte za izdelavo bolj učinkovitih in hitreje odzivnih uporabniških vmesnikov v njihovih spletnih platformah. Svelte je znan po tem, da je lahek in hiter, omogoča Rakutenu razvoj robustnih rešitev, ki izboljšujejo uporabniško izkušnjo na njihovih številnih spletnih storitvah.

Natančni načini uporabe Svelte-a v Rakutenu niso jasno zapisani v javno dostopnih virih, zato bi lahko šlo posebne aplikacije, ki so del njihove širše tehnološke infrastrukture. Spletne platforme, kot je Rakuten, pogosto iščejo tehnologije, ki lahko pripomorejo k boljši zmogljivosti in večji skalabilnosti, kar je v skladu s prednostmi, ki jih Svelte nudi.



Figure 2:Rakuten

## 

## 

## 

## 

## 

## 

## 4.2. The New York Times

The New York Times uporablja Svelte predvsem za izboljšanje interaktivnih grafik na svojem spletnem mestu. Svelte omogoča učinkovitejše in hitrejše nalaganje interaktivnih elementov, kar je ključno za medijsko podjetje, ki stremi k zagovarjajo kakovosti in hitro dostopnih novic. Rich Harris (ustvarjalec Svelte) je bil grafični urednik pri The New York Times in je neposredno uporabljal izkušnje, pridobljene pri delu, za izboljšanje in razvoj ogrodja Svelte.

Zaradi Svoje sposobnosti za hitro renderiranje in manjšo obremenitev brskalnika, Svelte omogoča The New York Timesu, da na svoji platformi učinkovito izvaja bogate interaktivne izkušnje, ki so ključnega pomena za sodobno poročanje in prikazovanje kompleknsih podatkovnih vizualizacij.

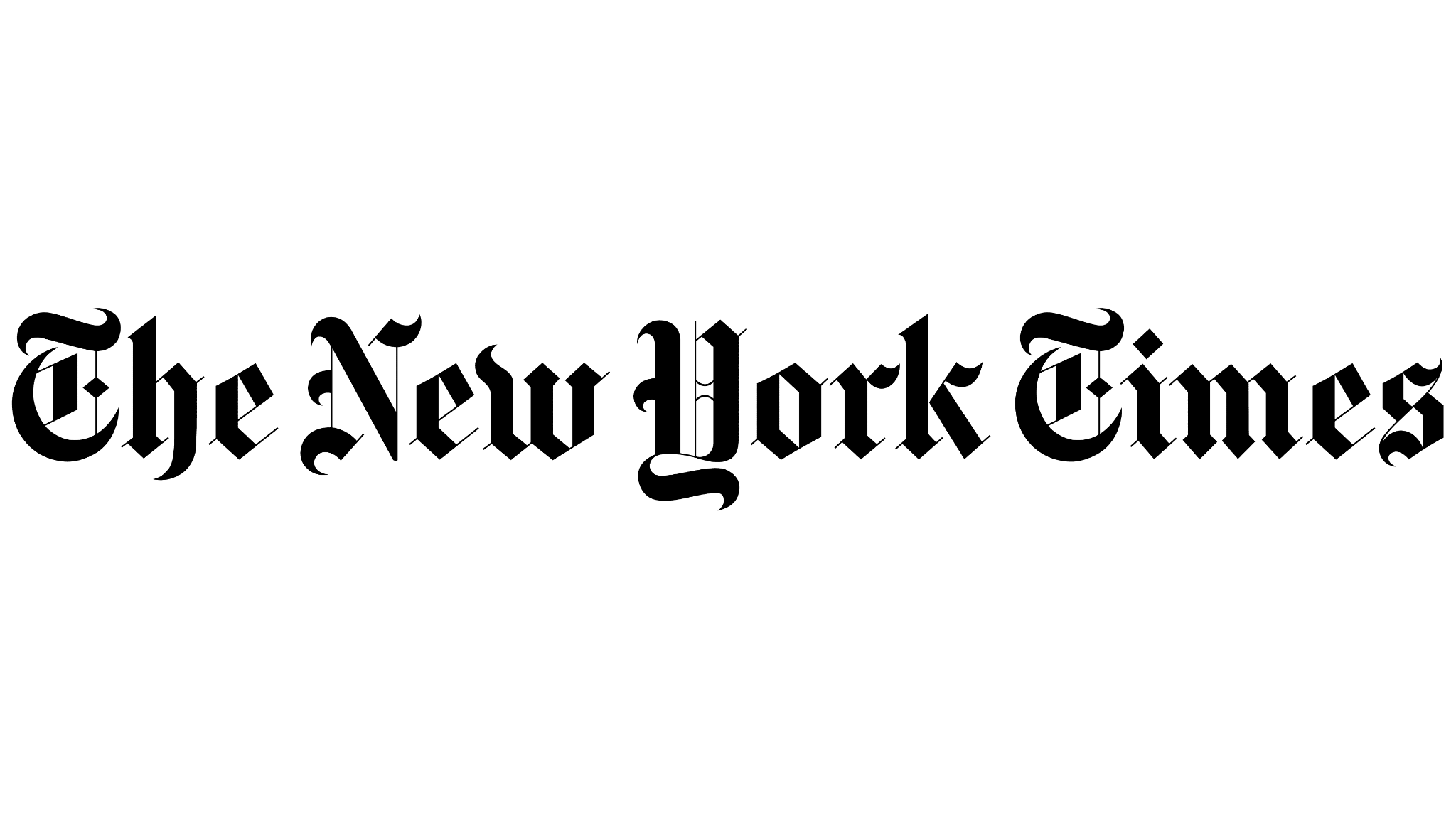


Figure 3:The New York Times

## 4.3. 1Password

Podjetje 1Password, tako kot mnoga druga znana podjetja, uporablja Svelte kot del svoje tehnološke zbirke. Konkretni načini kako 1Password integrira Svelte v svoje aplikacije, niso javno razkriti. Povsej verjetnosti uporabljajo Svelte za izboljšanje zmogljivosti in interaktivnosti svojih spletnih aplikacij, zahvaljujoč Svelte-ovemu učinkovitemu prevajanju v visoko optimiziran JavaScript.

Slika, ki vsebuje besede pisava, grafika, logotip, grafično oblikovanje

Opis je samodejno ustvarjen1Password je upravitelj gesel, ki omogoča varno shranjevanje in upravljanje gesel ter drugih občutljivih informacij, kot so podatki o bančnih karticah in identitetah. Uporabniki lahko shranijo svoja gesla v šifrirani bazi.

Figure 4:1Password

## 4.4. NameCoach

NameCoach je storitev, ki omogoča, da posnamejo pravilno izgovorjavo svojih imen, kar drugim olajša učenje in uporabo teh izgovorjav. To je še posebej uporabno v izobraževalnih ustanovah, podjetjih in drugih organizacijah, kjer je pravilna izgovorjava imen ključna za vzpostavljanje spoštljivih medsebojnih odnosov.

Podrobne informacije kako sta NameCoach in Svelte povezana ni dostopna za javnost. Možno je, da takšne informacije niso javno dostopne ali niso izrecno dokumentirane. NameCoach bi lahko uporabljal Svelte interno za svoje spletne apliakcije, da izboljša uporabniški vmesnike ali poenostavi dostavo svojih storitev, vendar brez konkretnih razkritij ali študij primerov. Natančne podrobnosti o njihovi integraciji ostajajo nejanse.



Figure 5:NameCouch

## 4.5. Creative Tim

Creative Tim je podjetje, ki se specializira za razvoj in ponudbo različnih UI kompletov, predlog in nadzornih plošč za spletne in mobilne aplikacije. Ti izdelki so zasnovani za olajšanje razvoja spletnih projektov z uporabo priljubljenih priljubljenih tehnologij kot so Bootstrap, Vue.js, React, Angular in Svelte. Creative Tim ponuja tako brezplačne kot premium teme, ki so priljubljene med razvojalci.

Med njihovimi produkti so tudi teme in predloge, ki so specifično zasnovane za uporabo z ogrodjem Svelte. To razvijalcem omogoča izdelavo hitrih in odzivnih spletnih aplikacij s pomočjo Svelte-a, izkoristijo pa tudi prednosti preoblikovalnih komponent in dizajnov, ki jih ponuja Creative Tim.

Creative Tim prav tako nudi podporo in dokumentacijo za svoje produkte, kar olajša integracijo in prilagoditev njihovih tem in komponent v obstoječe ali nove projekte. Njihovi izdelki so znani po visoki kakovosti in estetiki, kar je še poseben koristno za razvijalce, ki želijo zagotoviti profesionalen izgled svojih aplikacij brez potrebe po obsežnem dizajniranju od začetka.



Figure 6:Creative Tim

## 4.6. GoDaddy

GoDaddy je eno največji svetovnih podjetij, ki se ukvarja s spletnim gostovanjem in registracijo domen. Ustanovljeno je bilo leta 1997 in je znano po svojih storitvah, ki omogočajo posameznikom in podjetjem, da registrirajo spletne domene, gostujejo spletne strani, uporabljajo e-pošto, varnostne storitve in spletna orodja za izgradnjo spletnih strani.

GoDaddy uporablja Svelte v svojih spletnih projektih, kar je primer uporabe tehnologije v velikih in uveljavljenih podjetjih. Svelte je znan po svojih učinkovitosti in sposobnosti, da omogoča hitrejše nalaganje spletnih aplikacij, zahvaljujoč svoji arhitekturi, ki večino dela opravi že med samo gradnjo aplikacije, namesto med izvajanjem v brskalniku. To naredi Svelte privlačen za podjetja, kot je GoDaddy, ki želijo zagotoviti hitro in učinkovito spletno izkušnjo svojim uporabnikom.

Svelte je še posebej primeren za razvoj enostranskih aplikacij (SPA) in komponent, ki se lahko učinkovito ponovno uporabljajo v različnih delih aplikacije. Zaradi teh prednosti je GoDaddy verjetno izbral Svelte za določene dele svoje infrastrukture, da izboljša hitrost in odzivnost svojih spletnih storitev. Uporaba Svelte-a omogoča tudi lažje vzdrževanje in razvoj spletnih aplikacij, saj SVelte zmanjša potrebo po zapletenem upravljaju stanja in posodabljanju DOM-a.



Figure 7:Go Daddy

# MINIMALNE SISTEMSKE ZAHTEVE ZA ZAGON APLIKACIJE

**OPERACIJSKI SISTEM:** Aplikacije, razvite s Svelte so pretežno neodvisne od operacijskega sistema, saj se izvajajo v spletnem brskalniku. Vendar je za razvoj poterbno, da ima razvijalec nameščen katerikoli sodoben operacijski sistem, kot so Windows, macOS ali Linux, ki podpira sodobna orodja za razvoj, kot je Node.js

**NODE.JS:** Za razvoj in zagon Svelte aplikacij je potebna namestitev Node.js. To je okolje, ki omogoča izvajanje JavaScript kode na strani strežnika. Minimalna zahtevana različica Node.js je običajno najnovejša LTS verzij, ki zagotavlja stabilnost in podporo za najnovejše funkcionalnosti JavaScript.

**SPLETNI BRSKALNIK:** Svelte aplikacije zahtevajo sodoben spletni brskalnik za optimalno delovanje. PRiporočeni brskalniki vključujejo Google Chrome, Mozilla Firefox, Safari, in Microsoft Edge v njihovih najnovejših verzija. Ti brskalniki podpirajo sodobne web tehnologije, ki jih Svelte izkorišča za hitro in učinkovito izvajanje aplikacij.

**RAM IN PROCESOR:** Čeprav Svelte optimizira izvajanje aplikacij za zmanjšanje uporabe virov, je za razvoj in testiranje aplikacij priporočljivo imeti vsaj 4 GB RAM-a in sodoben procesor, ki lahko brez težav obvladuje več nalog hkrati. To še posebej velja, če razvijate večje ali bolj kompleksne aplikacije.

**PROSTOR NA DISKU:** Za namestitev razvojnega okolja, odvisnosti in vaših projektov je priporočljivo imeti vsaj nekaj GB prostega diskovnega prostora. Točna količina prostora se lahko razlikuje glede na obseg projekta in število odvisnosti, ki jih projekt uporablja.

**INTERNETNA POVEZAVA:** Za prenos orodij, knjižnic in delo z oddaljenimi repozitoriji (kot je GITHUB) je potrebna stabilna in hitra internetna povezava.

# NAVODILA ZA NAMESTITEV IN UPORABO

**NAMESTITEV POTEBNIH ORODIJ:** Zazačetek potrebujete Node.js, ki ga lahko prenesete in namestite s uradne strani Node.js (<https://nodejs.org/>). To bo omogočilo uporabo “npm” (Node Package Manager), ki je potreben za upravljanje z odvisnostmi in paketi v vašem Svelte projektu.

**USTVARJANJE NOVEGA SVELTE PROJEKTA:** Za ustvarjanje novega Svelte projekta odprite terminal in vnesite naslednji ukaz:



Figure 8:Izsek kode za ustvarjanje novega svelte projekta

Ta ukaz bo ustvaril novo mapo z imenom “moj-svelte-projekt” z vso potrebno začetno kodo.

**NAMESTITEV ODVISNOSTI:** Nato namestite potrebne odvisnosti znotraj projekta z ukazom: ***“npm install”***

**ZAGON RAZVOJNEGA STREŽNIKA:** Za zagon lokalnega razvojnega strežnika, ki omogoča vroče ponovno nalaganje (hot-reloading) uporabite: ***“npm run dev”***. To bo zagnalo lokalni strežnik, običajno dostopen prek “<http://localhost:5000>”, kjer lahko v živo vidite spremembe, ki jih naredite v kodi.

**GRADNJA APLIKACIJE:** Ko končate z razvojem, lahko aplikacijo pripravite za produkcijo z ukazom: “***npm run build”.*** To bo ustvarilo optimizirane datoteke, pripravljene za produkcijo v mapi “***public/build”.***

**NAPREDNEJŠA UPORABA:** Za nadaljnje raziskovanje Svelte funkcionalnosti, vključno z uporabo trgovin za upravljanje stanja (state management) in komponent, je zapisano na **uradni dokumentaciji Svelte**, ki ponuja obsežne informacije in vodiče.

## 6.1. GLAVNE ZNAČILNOSTI URADNE DOKUMENTACIJE SVELTE

**UVOD IN OSNOVE:** Uvod ponuja pregled, kaj je Svelte in kako se razlikuje od drugih JavaScript ogrodij. Zagotavlja vodič za začetek dela, vključno z navodili za namestitev in začetno konfiguracijo.

**VODIČ IN PRIMERI:** Dokumentacija vključuje podrobne vodiče, ki korak za korakom razložijo, kako izdelati aplikacije s Svelte. Primeri kritično pokrivajo izdelavo komponent, upravljanje stanj, integracijo z API-ji in še več.

**API REFERENCE:** Za podrobno tehnično razumevanje Svelte, dokumentacija nudi celovit seznam API-jev z opisi parametri in primeri uporabe. Ta del je ključnega pomena za razvijalce, ki želijo poglobljeno razumeti delovanje Svelte.

**KOMPONENTE IN TRGOVINE:** Posebni izdelki so namenjeni komponentam in trgovinam, ki so ključni elementi za upravljanje stanj znotraj Svelte aplikacij. Razložene so različne vrste trgovin in kako jih učinkovito uporabljati za reaktivno upravljanje podatkov.

**INTEGRACIJE IN DODATKI:** Dokumentacija obravnava tudi, kako integrirati Svelte z drugimi knjižnicami in orodji kot so TypeScript, GraphQL in različni sistem za upravljanje stanj (state management systems).

**MIGRACIJA IN RAZLIČICE:** Za tiste, ki prehajajo na Svelte iz drugih ogrodij ali nadgrajujejo iz starejših različic Svelte, dokumentacija nudi navodila za migracijo in opombe o združljivosti med različicami.

Dokumentacija je napisana jasno in strukturirano, z obilico primerov kode, ki pomagajo razvijalcem vizualizirati koncepte v praksi. Spletna skupnost okoli Svelte prav tako prispeva k dokumentaciji, kar pomeni, da se redno posodablja in izboljšuje z najnovejšimi informacijami in tehnikami. Za tiste, ki želijo raziskati vse, kar Svelte nudi, je uradna dokumentacija odlično izhodišče in nepogrešljiv vir.

# Prikaz nekaterih izbranih primerov uporabe

Svelte je priljubljen zaradi svoje učinkovitosti in enostavnosti, zato ga pogosto uporabljajo za izdelavo spletnih aplikacij, ki zahtevajo hitro odzivnost in minimalno breme na strani odjemalca.

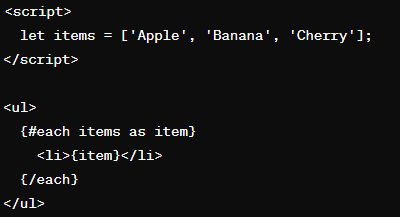
 **ENOSTAVNE APLIKACIJE (SPA):** Svelte je idealen za izdelavo enostavnih aplikacij, kjer je potrebno dinamično posodabljati vsebine brez osveževanja celotne strani. Primer take aplikacije je interaktivni uporabniški vmesnik za spletno trgovini. Primer kode:

Figure 9:Izsek kode

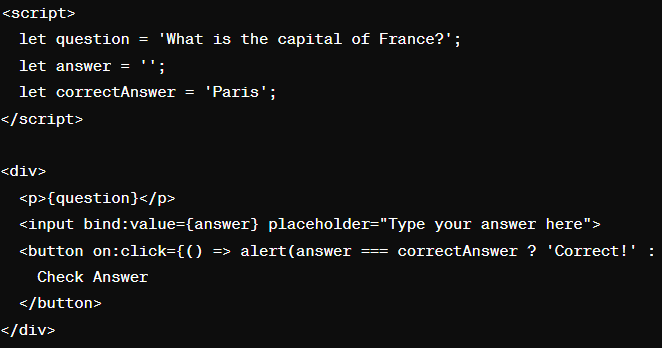
**INTERAKTIVNE IZOBRAŽEVALNE PLATFORME:** Za izobraževalne platforme, ki vključujejo kvize, interaktivne teste in druge oblike učenja, Svelte omogoča gladko integracijo dinamičnih komponent. Primer kode:

Figure 10:Izsek kode za integracijo dinamičnih komponent

**REALNOČASOVNE APLIKACIJE:** Spletne aplikacije, kot so klepetalnice ali aplikacije za sledenje podatkom v realnem času, so še eno področje, kjer se Svelte izkaže. Zmožnost reaktivnega posodabljanja UI brez dodatnega kodiranja je tu ključnega pomena. Primer kode:



Figure 11:Izsek kode za reaktivno posodabljanje UI

**UPORABA SVELTE ZA IZDELAVO KOMPONENT V OBSTOJEČIH APLIKACIJAH:**

Svelte se ne uporablja samo za izdelavo celotnih aplikacij, ampak se lahko uporabi tudi za izdelavo posameznih komponent v večjih, že obstoječih projektih. To omogoča razvijalcem, da postopoma uvajajo Svelte, kjer je to potrebno, ne da bi morali ponovno pisati celotne aplikacije. Primer kode:

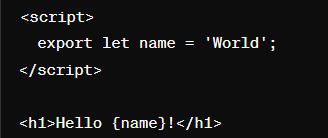


Figure 12:Izsek kode za izdelavo komponent

**UPRAVLJANJE STANJA V KOMPLEKSNIH APLIKACIJAH:** Svelte omogoča učinkovito upravljanje stanja v aplikacijah, kar je še posebej pomembno v večjih, kompleksnih sistemih. Primer kode:

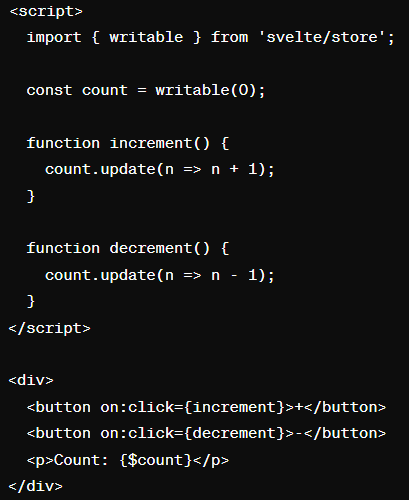


Figure 13:Izsek kode za upravljanje stanja

**SPLETNE APLIKACIJE Z VEČ POGLEDI** ( MULTI-VIEW WEB APPLICATIONS): SvelteKit, nadgradnja Svelte, omogoča enostavno izdelavo spletnih aplikacij z več pogledi in vključuje podporo za SSR (Server-Side Rendering), kar izboljša SEO in čas nalaganja strani. Primer kode za preprosto navigacijo:

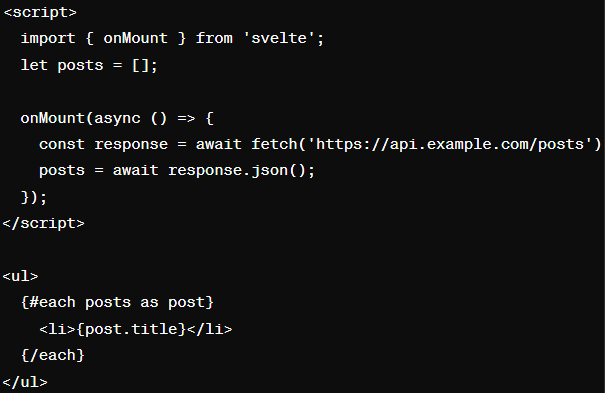


Figure 14:Izsek kode za navigacijo

**DINAMIČNO NALAGANJE PODATKOV:** Svelte omogoča dinamično nalaganje podatkov iz zunanjih virov, kar je ključno za aplikacije, ki se zanašajo na aktualne podatke, kot so novice ali borzni podatki. Primer kode:

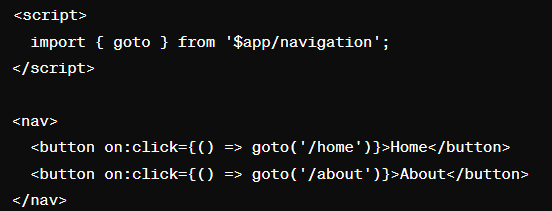


Figure 15:Izsek kode za dinamično nalaganje podatkov

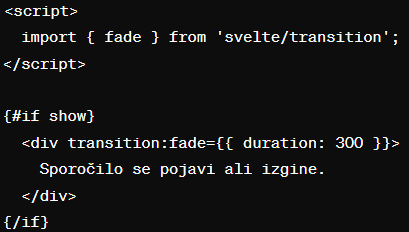
**ANIMACIJE IN PREHODI**: Svelte nudi mehanizme za animacije, ki omogočajo gladke prehode med stanji UI brez dodatne kode. Primer kode za prehod:

Figure 16:Izsek kode za animacije

Ti primeri prikazujejo širok spekter uporabe Svelte, od preprostih števcev do zapletenih aplikacij z dinamičnim nalaganjem podatkov in animacijami.

# PREDNOSTI IN SLABOSTI SVELTE

## 8.1. PREDNOSTI

* **KOMPILACIJSKI PRISTOP:** Svelte je edinstven, ker uporablja kompilacijski pristop, pri katerem se Svelte komponente pretvorijo v učinkovit imperativni JavaScript že med gradnjo aplikacije. To pomeni, da v brskalniku ni potreben dodaten sloh virtualnega DOM-a, kar zmanjšuje obremenitev in izboljša hitrost delovanja.
* **MANJŠA VELIKOST SKRIPT:** Zaradi svojega kompilacijskega pristopa Svelte ustvarja manjše JavaScript skripte, ki se hitreje nalagajo in izvajajo. Ta prednost je še posebej opazna v primerjavi z drugimi JavaScript ogrodji, ki uproabljajo virtualni DOM
* **REAKTIVNOST BREZ DODATNIH KNJIŽNIC:** Svelte omogoča reaktivnost na zelo naraven način, saj lahko razvijalci definirajo reaktivne spremenljivke z uporabo preprostega sintaktičnega sladkorja. To odpravlja potrebo po uporabi dodatnih knjižnic za upravljanje stanj, kot je Redux, ki se pogosto uporabljajo v drugih ogrodjih.
* **ENOSTAVNOST UPORABE:** Svelte je znan po svoji preprosti sintaksi, ki je lahko razumljiva začetnikom in tistim, ki že poznajo HTML, CSS in JavaScript. To naredi ogrodje dostopno širšemu krogu razvijalcev.
* **IZBOLJŠANA IZKUŠNJA RAZVIJALCEV:** Zaradi manjše boilerplate kode in integrirane reaktivnosti lahko razvijalci s Svelte hitreje razvijejo in vzdržujejo svoje aplikacije.

## 8.2. SLABOSTI

* **MANJŠE SKUPNOSTI IN EKOSISTEM:** Čeprav Svelte pridobiva na priljubljenosti, njegova skupnost in ekosistem nista tako velika ali dobro razvita kot pri React ali Angular. To lahko pomeni manjšo razpoložljivost tretjih knjižnic, vtičnikov in orodij.
* **MANJŠA BAZA DELOVNIH MEST:** Za razvijalce, ki iščejo priložnost, je lahko slabost, da je na trgu manj delovnih mest, ki zahtevajo znanje Svelte v primerjavi z bolj uveljavljenimi tehnologijami.
* **OMEJENE VGRAJENE FUNKCIONALNOSTI:** V primerjavi z bolj celovitimi ogrodji, kot je Angular, ki ponuja obsežen nabor vgrajenih funkcionalnosti, Svelte morda ne ponuja toliko izbire. To pomeni, da za določene funkcije morda poterbujete dodatne knjižnice ali lastne implementacije.
* **IZZIVI PRI MIGRACIJI:** Podjetja, ki že uporabljajo druga orodja, lahko naletijo na izzive pri migraciji na SVelte, še posebej, če so njihove trenutne aplikacije kompleksne in tesno integrirane z obstoječimi rešitvami.
* **NESTANDARDNE REŠITVE:** Ker Svelte pristopa k reaktivnosti na nekoliko drugačen način, se lahko pojavijo izzivi pri integraciji z nekaterimi tradicionalnimi JavaScript knjižnicami, ki pričakujejo klasičen model virtualnega DOM-a

Kljub temu so prednosti Svelte, kot so hitrost, enostavnost in manjši JavaScript izpisi, razlog, zakaj mnogi razvijalci in podjetja izbirajo to ogrodje za svoje projekte.

# PRIMERJALNA ANALIZA SVELTE Z DRUGIMI ORODJI

## 9.1 Svelte vs React

### **ARHITEKTURA**

* **SVELTE:** Namesto tradicionalnega virtualnega DOM-a, SVelte pri gradnji aplikacije ustvari visoko optimiziran imperativni JavaScript. Ta pristop zmanjša delo, ki ga mora opraviti brskalnik, kar lahko vodi do hitrejših časov nalaganja in bolj odzivne uporabniške izkušnje.
* **REACT:** Uporablja virtualni DOM, ki omogoča učinkovite posodobitve UI z minimalnim vplivom na dejanski DOM. To je koristno za kompleksne aplikacije, vendar lahko privede do dodatne obremenitve, če ni pravilno opravljeno.

### **UČENJE IN UPORABNOST**

* **SVELTE:** Nudi preprostejšo in bolj naravno integracijo z JavaScriptom, CSS-om in HTML-om, kar zmanjšuje potrebo po učenju specifičnih sintaks ali konceptov, kot so JSX pri React.
* **REACT:** Zahteva učenje JSX-a in razumevanje konceptov, kot so komponente, state management in hooks, kar lahko predstavlja višjo začetno učno krivuljo.

### **SKUPNOST IN EKOSISTEM**

* **SVELTE:** Čeprav hitro raste, je skupnost okoli Svelte še vedno manjša v primerjavi z Reactom. To pomeni manjšo razpoložljivost pripravljenih rešitev in dodatkov
* **REACT:** Ima eno največjih skupnosti med front.end ogrodji, obsežen ekosistem in obilo knjižnic, ki omogočajo hitro integracijo z različnimi API-ji in zunanjimi storitvami.

## 9.2. Svelte vs Angular

### **CELOVITOST REŠITVE**

* **Svelte:** Je bolj usmerjen k specifičnim rešitvam in manj obremenjujoč za začetek projekta. Manj je predpisov glede arhitekture aplikacije, kar ponuja večjo fleksibilnost.
* **Angular:** Ponuja ce
* lovit nabor funkcionalnosti, vključno z vgrajenim routingom, formami in naprednimi upravljanjem stanja. To ga naredi za močno rešitev za velike aplikacije, vendar lahko poveča kompleksnost razvoja.

### **TESTIRANJE IN RAZVOJ**

* **Svelte:** Manjša osnovna kompleksnost lahko olajša testiranje in razhroščevanje.
* **Angular:** Zaradi svoje integrirane narave in odvisnosti od TypeScripta, Angluar podpira stroge razvojne vzorce in močno tipizacijo, kar lahko izboljša kakovost kode, a hkrati zahteva bolj strukturiran pristop k testiranju.

## 9.3. Svelte vs Vue

### **SINTAKSA IN RAZVOJNA IZKUŠNJA**

* **Svelte:** Svojo logiko, stilizacijo in označevanje združuje na intuitiven način, ki olajša hitre razvojne cikle.
* **Vue:** Ponuja podobno integracijo HTML, CSS in JavaScript v enotnih datotekah komponent. Vue je priljubljen zaradi svoje enostavne integracije in modularne arhitekture.

### **REAKTIVNOST**

* **Svelte:** Reaktivnost je osnovno vgrajena, kar pomeni, da ni potrebna zunanja knjižnica za upravljanje stanj. Reaktivne izjave so deklarativne in preproste za razumevanje.
* **Vue:** Vue prav tako ponuja močan sistem reaktivnosti, vendar zahteva uporabo Vue-ovega lastnega sistema za sledenje odvisnosti.

Izbira med Svelte, React, Angular ali Vue bo odvisna od številnih dejavnikov, vključno z zahtevami projekta, izkušnjami razvijalskega tima in želenimi značilnostmi končne aplikacije. Medtem ko Svelte ponuja prednost v smislu učinkovitosti izvajanja in enostavnosti razvoja. React in Vue nudita obsežnejše skupnosti in bogatejše ekosisteme, Angular pa je optimalen za razvoj velikih, kompleksnih aplikacij z močno arhitekturo.

# Aplikacija

## 10.1 Cinematrix aplikacija

S pomočjo Svelte smo tudi razvili aplikacijo “Cinematrix”, ki je minimalistična spletna aplikacija za pridobivanje in shranjevanje filmov preko OMDb API-ja. V aplikaciji so razkazni nekateri ključni elementi Svelte kot so dinamične komponente, form-handling, binding za attribute, razrede in HTML kodo ter predstavitev »writeable stores« (globalne spremenljivke npr. kdo je trenutno prijavljen).

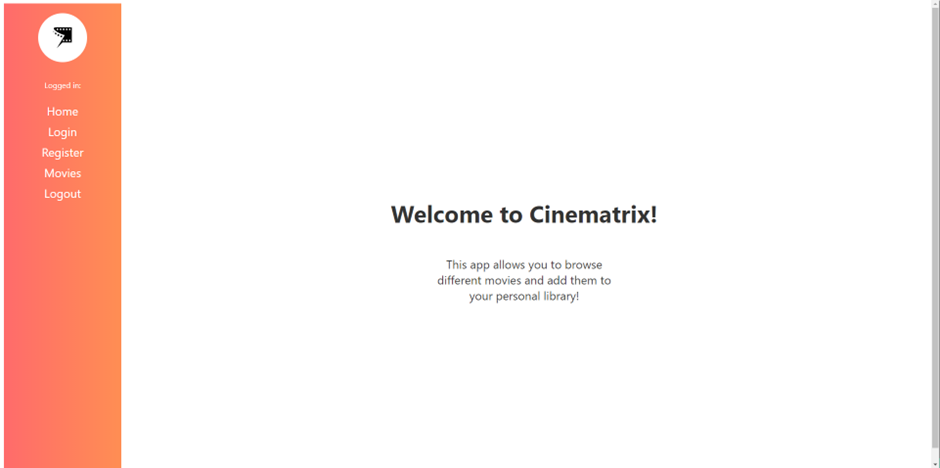


Figure 17:Cinematrix Aplikacija

Uporablja tudi API, ki smo ga zgradili za prenos podatkov o uporabnikih v bazo in iz baze. Ta API deluje na Express middleware.

## 10.2 OMDb API

OMDb API je javni brezplačen API za pridbivanje podatkov o filmih. Vsebuje veliko podatkov, ki so predstavljeni v JSON obliki, vključuje pa tudi postere filmov

<http://www.omdbapi.com/>

Za uporabo API-ja je potreben API-key, katerega dobiš ob registraciji v njihov sistem.

## 10.3 FUNKCIONALNOSTI

### **Registracija**

Spletna aplikacija vsebuje funkcijo za registracijo uporabnika, ki deluje tako, da uporabnik vnese svoje podatke v Register form in jih pošlje preko API-ja v podatkovno bazo, kjer se zabeleži. Aplikacija preverja če uporabnik že obstaja in ali je vnesel pravilen email naslov.

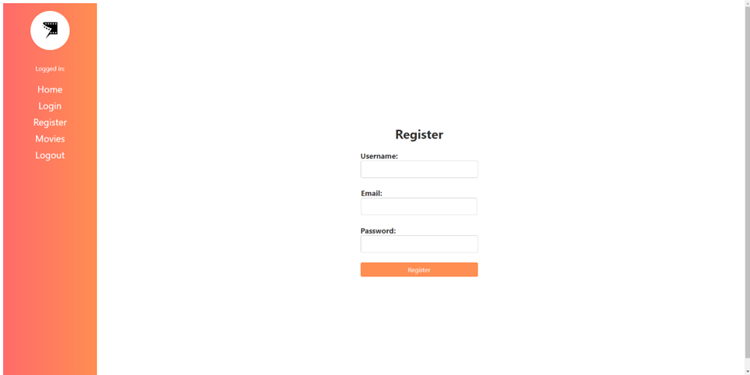


Figure 18:Registracija

### **Login**

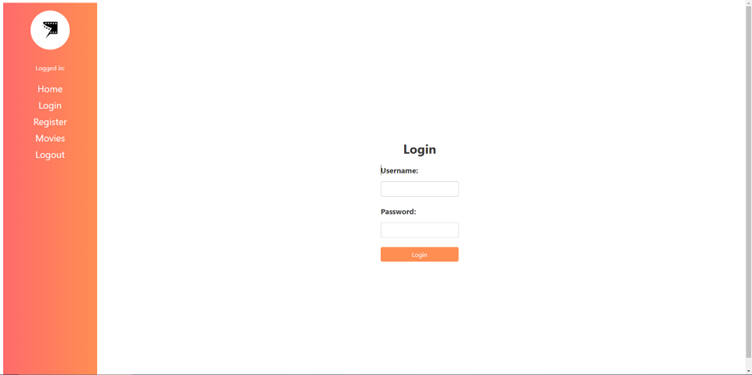
Uporabnik se po registracjii nato lahko prijavi preko login okna, vpiše svoje podatke in sistem preveri ali uporabnik obstaja in ali je pravilno vnesel geslo.

Figure 19:Login

### **Movies**

Glavna funkcionalnost spletne aplikacije pa je dodajanje filmov na osebni profil preko OMDb API-ja.

Če se prijavimo in se pomaknemo na okno Movies dobimo search bar in list z filmami dodanimi na profil. V search bar lahko napišemo katerikoli naziv filma in z pritiskom na gumb search pošljemo GET zahtevo na API. Rezultat zahteve nato vrne podatke povezane z filmom, katere parseamo in jih na spletni strani pokažemo (v našem primeru poster in opis filma).

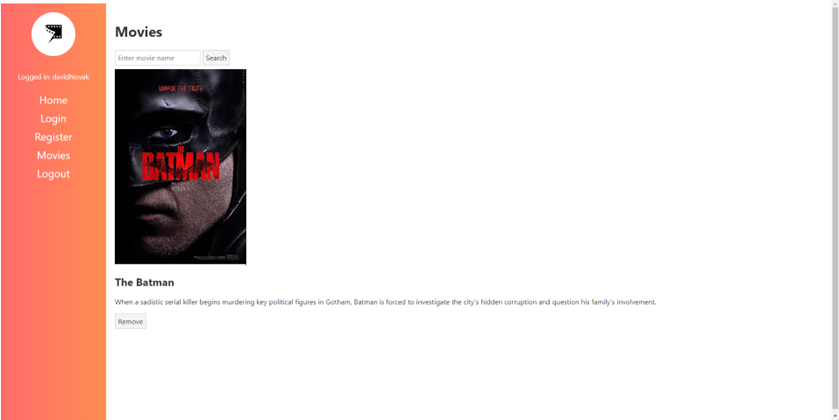


Figure 20:Movies

### **Logout**

Iz aplikacije se preprosto odjaviš s pritiskom na logout gumb, ki poskrbi, da se vsi podatki uporabnika shranijo in izbrišejo iz lokalne shrambe brskalnika.

# ZAKLJUČEK

V naši spletni aplikaciji se pokaže učinkovitost Svelte-a, saj vidimo po gradnji aplikacije, če bi uporabili samo čisti Javascript za programiranje, bi rabili 5000+ vrstic JS. Z uporabo Svelte smo število vrstic zmanjšali na približno 500 vrstic JS.

Razvoj aplikacije je osebno bil zelo preprost saj deluje na istem routing principu kot npr. .NET MVC, za usmerjanje strani in prenos podatkov preko strani. Svelte je definitivno eden izmed vodilnih ogrodij kot so React, Vue.js, Angular, … za izdelavno spletnih strani/aplikacij, edina slabost bi lahko bila nekatere zakomplicirane in slabo vpeljane komponente. Priporočljivo je tudi uporabiti kakšne UI knjižnice kot npr. Daisyui in Tailwind CSS za CSS kodo.

# VIRI

* <https://en.wikipedia.org/wiki/Svelte>
* <https://github.com/sveltejs/learn.svelte.dev>
* <https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Svelte_getting_started>
* <https://kit.svelte.dev/docs/introduction>
* <https://learn.svelte.dev/tutorial/welcome-to-svelte>

# 