

Upute za korištenje alata u obliku interaktivnog web tutorijala

Novak, Niko

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:227853>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-31**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli

Tehnički fakultet u Puli

NIKO NOVAK

UPUTE ZA KORIŠTENJE ALATA U OBLIKU INTERAKTIVNOG WEB TUTORIJALA

Završni rad

Pula, travanj, 2023.

Sveučilište Jurja Dobrile u Puli

Tehnički fakultet u Puli

NIKO NOVAK

UPUTE ZA KORIŠTENJE ALATA U OBLIKU INTERAKTIVNOG WEB TUTORIJALA

Završni rad

JMBAG: 0303090476, redoviti student.

Studijski smjer: Računarstvo

Predmet: Multimedijalni sustavi

Mentorica: doc.dr.sc. Željka Tomasović

Pula, travanj, 2023.

Sadržaj

1. Uvod.....	4
2. Multimedija, raspodjela i vrste multimedije.....	5
2.1 Osnovni multimedijски elementi	5
2.2. Hipertekst i hipermedija	6
2.2. Interaktivnost i razine interaktivnosti	6
3. Unity.....	8
3.1. Unity UI	8
4. Korišteni alati za izradu „web“ aplikacije	10
4.1. Vue.js	10
4.2. HTML	11
4.3. CSS.....	12
4.4. JavaScript.....	14
5. Izrada „web“ aplikacije.....	15
5.1. Instalacija „Vue.js-a“ i kreiranje projekta.....	15
5.2. Definiranje ruta „index.js“	19
5.3. Komponente.....	21
5.3.1. „Home.vue“	21
5.3.2. „Skripta.vue“	22
5.3.4. „Video.vue“	25
5.3.5. „Navbar.vue“	25
6. Zaključak.....	26
7. Popis slika	27
8. Literatura	28
9. Sažetak.....	30
10. Abstract	31

1.Uvod

U doba interneta i mogućnosti učenja o bilo čemu putem interneta kreirana je ova „web“ aplikacija uz pomoć koje se može lakše naučiti o korištenju alata „Unity“. Unity je izabran jer ima besplatnu verziju „cross-platform engine-a“ te je jedan od popularnijih „game engine-a“ koji se koriste za izradu 2D i 3D igara. Kroz ovaj rad može se naučiti kako kreirati „Vue.js“ aplikaciju koja sadržava tutorijal o „Unity-u“. Tehnologija koja je korištena u ovom radu je „Vue“, čije se datoteke sastoje od „HTML-a“, „CSS-a“ i „Javascript-a“. Cilj rada je izrada uputa za korištenje „Unity-a“ u obliku interaktivnog „web“ tutorijala, što je postignuto tako da korisnici imaju mogućnost pretraživanja tutorijala po svojoj volji uz navigacijske trake na kojima se nalaze gumbi. Gumbi uglavnom vode na onaj naslov koji taj gumb sadržava.

Za ovaj tutorijal nastojao sam kreirati osnovnu platformersku igru, čiji fokus nije na izgledu igre, već samo na funkcionalnosti, koja je potom objašnjena na „web“ aplikaciji. Igra je jednostavna, sastoji se od dva nivoa, s time da se na prvom nivou nalazi sva funkcionalnost igre. Razlog kreiranja drugog nivoa jest taj što se može vidjeti prijelaz iz jednog nivoa na drugi.

2.Multimedija,raspodjela i vrste multimedije

Pojam multimedija(engl. multimedia) dolazi od latinskih riječi mnogi(lat. multus) i medij(lat. medium), a predstavlja integraciju različitih medijskih elemenata koji su u osnovi samostalni sadržaji. Multimedia je informacija predstavljena ili spremljena u kombinaciji teksta, grafike, zvuka, animacije i videa objedinjenih putem računala.^[1] Pojavom interneta i njegovim sve češćim korištenjem, dovelo je do sve manjeg korištenja tradicionalnih masovnih medija kao što su: knjige, filmovi, novine, radio. Informacije se lakše i bolje pamte ako su primljene kroz više osjetila.^[1]

2.1 Osnovni multimedijски elementi

Multimedijalni digitalni materijali zasnivaju se na primjeni osnovnih multimedijских elemenata kao što su:

Grafika

- važna za vizualni dojam multimedijске aplikacije
- bitmape(Paint ili rasterska grafika) ili vektorske(drawn grafika)
- bitmape foto-realistične slike i kompleksne crteže koji zahtijevaju fine detalje
- vektorska grafika – za grafičke oblike koji se mogu matematički izraziti koordinatama, duljinom i kutovima

Tekst

- fontovi, stilovi, veličine i boje
- tekst bitan za interakciju i navigaciju kroz aplikaciju: „gumbi“, ključne riječi

Zvuk

- digitalizirane audio datoteke – zvučni valovi unose se u računalo u analognom obliku i putem zvučne kartice se pretvaraju u digitalni oblik
- kod audio zapisa ljudski glas daje smjernice za izvršavanje nekog postupka

Animacija

- udruženi dinamični povezani skup grafičkih elemenata/slika nosi sva obilježja grafičkih prikaza
- promjena položaja objekata u pravilu je vezana za izmjenu slike

- omogućava dinamičko predstavljanje činjenica i događaja kao vremenski slijed različitih slikovnih sadržaja

Video

- video kao film sačinjen od niza kadrova različitih slika koje kao i kod animacije brzo prikazane u nizu stvaraju dojam pokreta

Interaktivna multimedija omogućava korisnicima kontrolu nad aplikacijom, odnosno interaktivnost s aplikacijom. Ako je dodana struktura hiperveza onda to nazivamo hipermedijom.^[2]

2.2. Hipertekst i hipermedija

Hipertekst(engl. hypertext) je nelinearna metoda organiziranja i prikazivanja informacija u obliku teksta koji sadrži veze na druge dokumente.^[2] Pod veze se misli na hiperveze(engl. hyperlinks, hot words, hot links). Korisnik hiperteksta stvara svoj vlastiti put pretraživanja i čitanja, te o načinu na koji želi učiti o alatu. Hipermedija(engl. hypermedia) naglašava netekstualne komponente dodane hipertekstu(animaciju, zvuk i video), odnosno možemo hipermediju prikazati kao spoj hiperteksta i multimedije: hipermedija je jednaka hipertekstu + multimedija. Hipermedijske teme su povezane tako da korisnik u potrazi za informacijom prelazi s predmeta na drugi vezani predmet.^[2]

2.2. Interaktivnost i razine interaktivnosti

Interaktivnost je vrsta dijaloga korisnika i aplikacije. Hipermedija kao svoju bitnu komponentu uključuje interaktivnost i od korisnika zahtijeva aktivnost. Omogućuje korisniku biranje, odlučivanje, ali i povratno utjecanje na program u stvarnom vremenu zahvaljujući postojanju više navigacijskih putanja u hipermedijskom programu. Navigacija(engl. navigation) je način kretanja korisnika kroz hipermedijsku aplikaciju. Postoje više vrsta navigacija, a one su određene pri korištenju modela kod izgradnje aplikacije, a to su: linearna, hijerarhijska, kružna i slobodna navigacija. Olakšavanje navigacije u aplikaciji radimo implementacijom

pomoćnih alata kao što su mape, „gumbovi“ s tekstom ili simbolima itd. Postoje više razina interaktivnosti, a dijelimo ih po:

1. stupnju kontrole koju ima korisnik
2. kreiranju novih iskustava za korisnike od strane programa
3. stupnju povratnih informacija koje nudi program^[1]

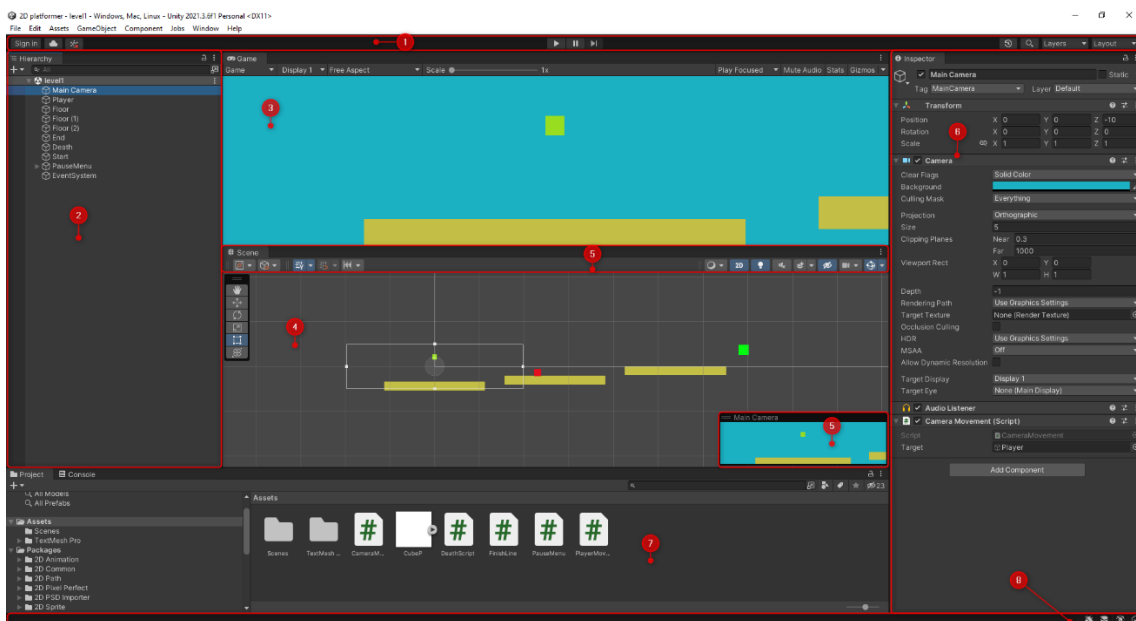
3. Unity

Za stvaranje 2D i 3D igara i aplikacija potrebno je koristiti „Unity Editor“. Za početak je potrebno instalirati „Unity Hub“. „Unity Hub“ je samostalna aplikacija koja pojednostavljuje način na koji se navigira, preuzima i upravlja Unity projektima i instalacijama.[3] „Hub“ se koristiti za:

- Upravljanje, preuzimanje i instaliranje modula i verzije „Unity Editora“
- Kreiranje i upravljanje vlastitih Unity projekata
- Istraživanje predložaka, uzoraka projekta i za učenje različitih vještina
- Upravljanje vlastitim profilom, postavkama i „Unity“ licencama

3.1. Unity UI

Na slici 1. prikazan je glavni zaslone i glavni elementi „Unity Editor-a“.



Slika 1. Prikaz glavnog zaslona "Unity Editor-a"

Izvor: Djelo autora rada

1. **Alatna traka**(engl. toolbar) pruža pristup Unity računu i uslugama „Unity Cloud-a“. Također sadrži kontrole za način rada reprodukcije; „Unity Search“; izbornik vidljivosti sloja(engl. layer visibility menu); „Editor layout menu“.
2. **„The Hierarchy window“** je hijerarhijsko tekstualno prikazivanje svakog „GameObject-a“ u sceni. Svaka stavka u sceni ima unos u hijerarhiji, pa su

dva prozora inherentno povezana. Hijerarhija otkriva strukturu načina na koji „GameObjecti“ prilože jedan drugome.

3. **Prikaz igre**(engl. Game view) simulira kako će konačno rendirana igra izgledati kroz kameru u sceni. Kada kliknemo na „gumb“ „Play“, simulacija igre započinje.
4. **Prikaz scene**(engl. Scene view) omogućuje vizualno navigiranje i uređivanje scene. Prikaz scene može prikazati 3D ili 2D perspektivu, ovisno o vrsti projekta koja se radi.
5. **„Overlays“** sadrži osnovne alate za manipuliranje prikaza scene i „GameObject-ima“ unutar nje. Također može dodati „custom Overlays“ za poboljšavanje radnog procesa.
6. **„The Inspector window“** omogućuje pregledavanje i uređivanje svih svojstava trenutno odabranog „GameObject-a“. Raspored i sadržaj inspektora mijenja se svaki put kad je odabran drugi „GameObject“ zbog toga što različite vrste „GameObject-a“ imaju različite skupove svojstava.
7. **„The Project window“** prikazuje „library“ resursa(engl. library of Assets) koji su dostupni za upotrebu u projektu. Kad se resursi uvežu u projekt, pojavljuju se u „Project window-u“.
8. **Statusna traka**(engl. status bar) pruža obavijesti o različitim procesima „Unity-a“ i brzi pristup povezanim alatima i postavkama.^[3]

4. Korišteni alati za izradu „web“ aplikacije

Za izradu „web“ stranice korištena je tehnologija koja se zove „Vue.js“. Nadovezuje se na standardni „HTML“, „CSS“ i „JavaScript“. U izradi stranice „HTML“ se koristi za prikaz elemenata korisniku, „CSS“ se koristi za uređivanje stila stranice, te na kraju „JavaScript“ koji služi za kreiranje funkcionalnosti, naprimjer: kreiranje funkcije za „gumb“ koji vodi na određeno poglavlje skripte.

4.1. Vue.js

„Vue.js“ je „JavaScript“ okvir (engl. framework) za izgradnju korisničkih sučelja(engl. user interfaces).[4] Omogućuje jednostavno vezanje podatkovnih modela(engl. data models) na „representation layer“. Jedna od boljih stvari koje omogućuje jest ta da se komponente mogu ponovno upotrebljavati u cijeloj aplikaciji. Može se koristiti na različite načine, ovisno o potrebi:

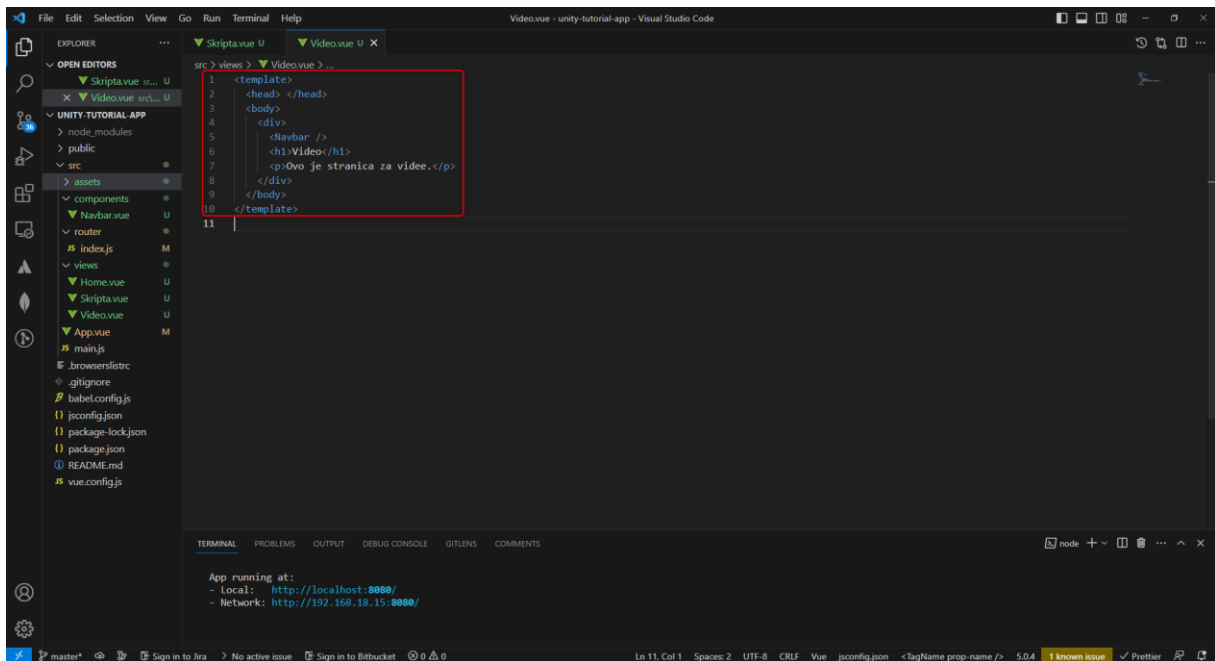
- Poboljšavanje statičkog „HTML-a“ bez koraka izgradnje(engl. build step)
- Ugradnja kao „web“ komponente na bilo koju stranicu
- Aplikacija na jednoj stranici
- „Fullstack“ / iscrtavanje na strani poslužitelja
- „Targeting“ računala, mobilnih uređaja, „WebGL-a“, terminala

„Vue.js“ ima puno prednosti, a neke od njih su reaktivno povezivanje podataka. Koristi sustav reaktivnog povezivanja podataka, odnosno kad se podaci mijenjaju, korisničko sučelje(engl. UI) se automatski ažurira kako bi održavalo te promjene. Na taj način se olakšava izgradnja dinamičkih i interaktivnih korisničkih sučelja. Još jedna od prednosti je virtualni DOM(engl. Document Object Model). Korištenjem virtualnog DOM sustava se promjene u korisničkom sučelju prvo unose u virtualni prikaz DOM-a prije nego se dogode promjene na stvarnom DOM-u. Ovim načinom se optimizira performansa i smanjuje broj potrebnih ažuriranja DOM-a. „DevTools Vue.js“ dolazi sa skupom razvojnih alata koji olakšavaju otklanjanje pogrešaka i optimiziranje koda. Također pruža povratne informacije u stvarnom vremenu o izvedbi aplikacije i omogućuje bolji pregled hijerarhije komponenata. Ima sve što je potrebno za „frontend“ razvoj, te jedna od prednosti za koristiti jest njegova

preglednost, iz razloga što se „HTML“, „JavaScript“ i „CSS“ nalazi u istom dokumentu.

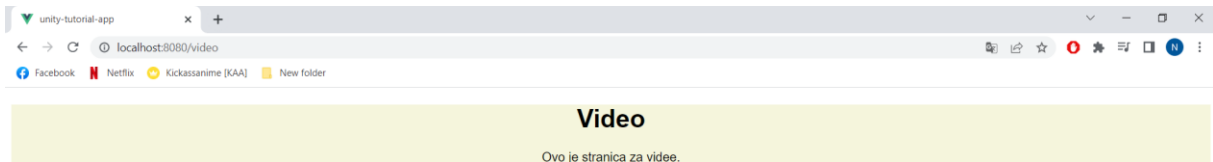
4.2. HTML

„HTML“ (engl. HyperText Markup Language) je standard za kreiranje „web“ stranica. Omogućuje stvaranje i strukturu odjeljaka (engl. sections), odlomaka (engl. paragraphs) i poveznica (engl. links) koristeći „HTML“ elemente. Ti elementi su ustvari građevni blokovi (engl. building blocks) svake stranice. Uz pomoć „<template> </template>“ dajemo „Visual Studio Code-u“ do znanja da pišemo „HTML“. Na slici 2. se vidi primjer strukture „HTML-a“ u „Vue.js-u“, a ispod na slici 3. se vidi kako izgleda primjer strukture na stranici.



Slika 2. Primjer strukture „HTML-a“ u „Vue.js-u“

Izvor: Djelo autora rada



Slika 3. Primjer strukture „HTML-a“ u „Vue.js-u“ prikazan kao stranica

Slika 3. Izvor: Djelo autora rada

Tri glavna dijela elementa su:

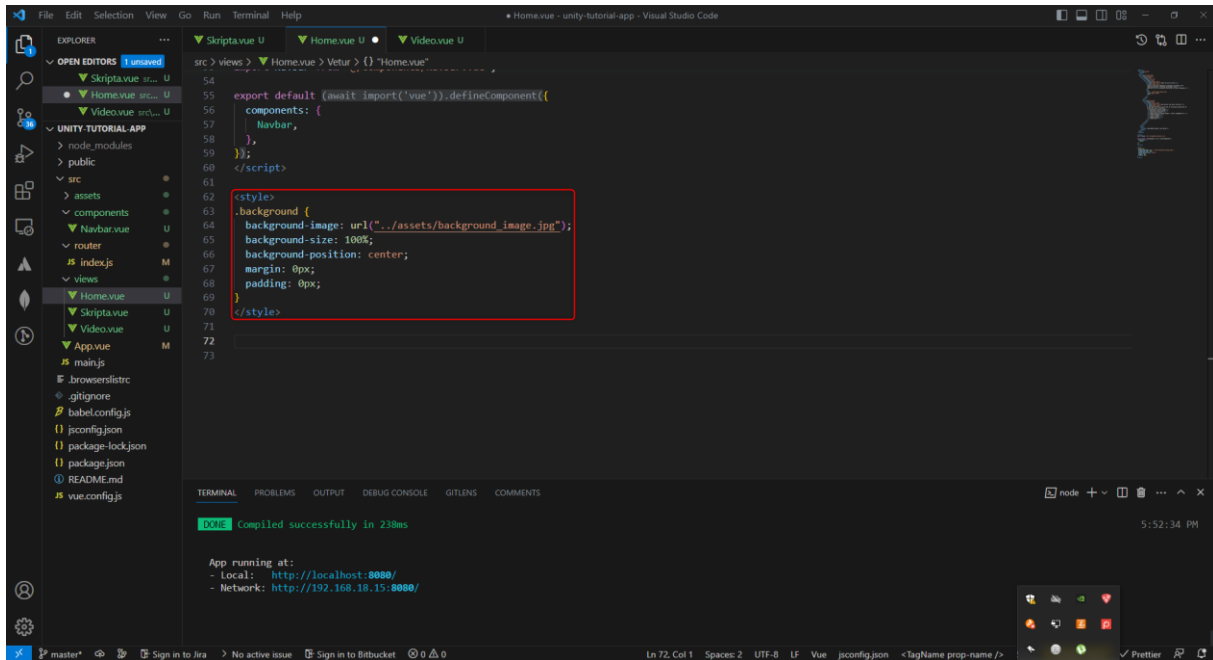
- Oznaka otvaranja (engl. opening tag) – koristi se za navođenje gdje element počinje. Označava se simbolom „<“.
- Sadržaj (engl. content) – ovo je izlaz (engl. output) koji korisnici vide.
- Oznaka zatvaranja (engl. closing tag) – koristi se za navođenje gdje element završava. Označava se simbolom „>“.

Kombinacijom ova tri dijela stvara se „HTML“ element. Npr. to može biti neki paragraf tipa: „<p> Ovo je paragraf. </p>“. Može se vidjeti da su obuhvaćeni svi potrebni dijelovi za kreiranje „HTML“ elementa.

4.3. CSS

„CSS“ (engl. Cascading Style Sheets) je stilski jezik koji se koristi za opisivanje prezentacije dokumenata napisanih u „HTML-u“. On opisuje kako se elementi trebaju prikazati na ekranu, na papiru, u govoru ili na drugim medijima.^[11] Jedan je od temeljnih jezika otvorenog „web-a“ i standardiziran je prema svim preglednicima.

Koristi se za stiliziranje i raspored „web“ stranica kao što je: pozadinska slika,boja, promjena fonta, veličine, boje, centriranost itd. Oznaka „<script> </script>“ daje do znanja „Visual Studio Code-u“ da koristimo „CSS“. Na sljedećoj slici 4. je prikazana „CSS-ova“ struktura.



The screenshot shows the Visual Studio Code interface with a Vue.js component file open. The code defines a component with a `<style>` block containing CSS rules for background styling. The CSS rules are: `.background { background-image: url("../assets/background_image.jpg"); background-size: 100%; background-position: center; margin: 0px; padding: 0px; }`. The terminal at the bottom shows a successful compilation and the application running on localhost:8080.

```
export default (await import('vue')).defineComponent({
  components: {
    navbar,
  },
});
</script>

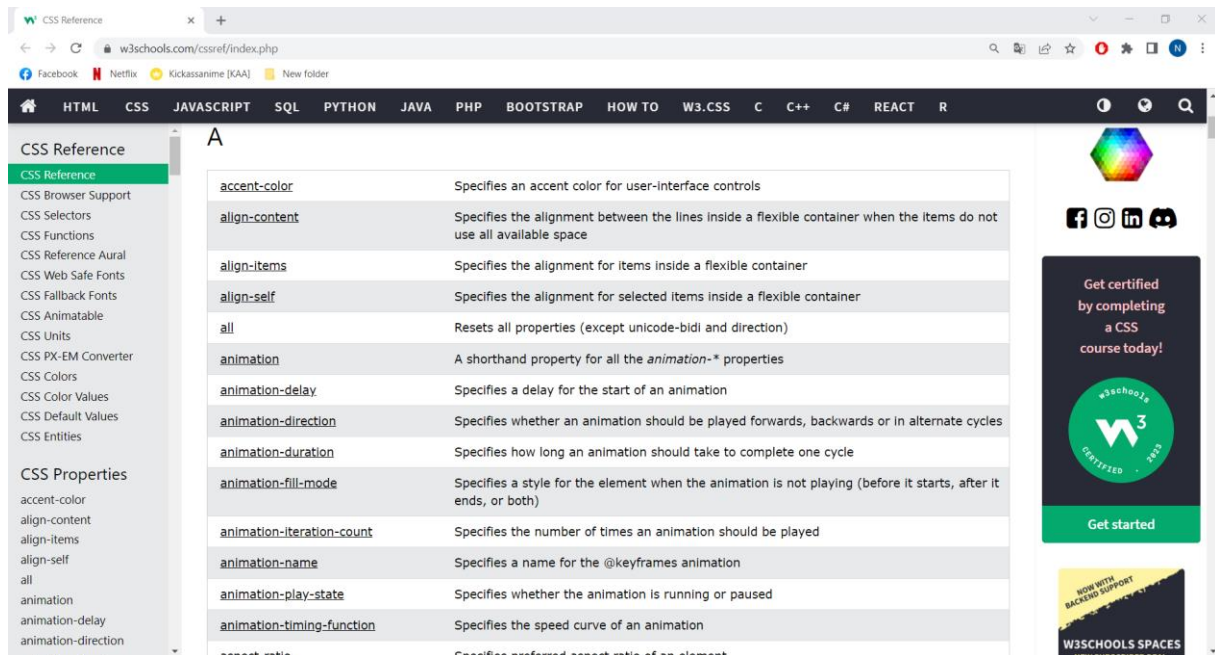
<style>
.background {
  background-image: url("../assets/background_image.jpg");
  background-size: 100%;
  background-position: center;
  margin: 0px;
  padding: 0px;
}
</style>
```

Terminal output:
Compiled successfully in 238ms
App running at:
- Local: http://localhost:8080/
- Network: http://192.168.18.15:8080/

Slika 4. Prikaz „CSS-ove“ strukture u kodu

Izvor: Djelo autora rada

Sljedeća slika 5. prikazuje popis dio komanda koje „CSS“ sadržava.



Slika 5. Prikaz dio komanda koje se koriste za „CSS“

Izvor: <https://www.w3schools.com/cssref/index.php>

Jedna od prednosti „CSS-a“ jest kaskadnost i specifičnost, odnosno „CSS“ slijedi kaskadni model i model specifičnosti zbog kojih se na isti element može primijeniti više stilova. To omogućuje stvaranje složenih i nijansiranih stilova za „web“ stranice. „CSS“ koristi model za opisivanje rasporeda elemenata na web stranici, odnosno svaki element se tretira kao okvir s područjem sadržaja, marginom, obrubom i ispunom. Na taj način je olakšano stvaranje izgleda na različitim uređajima i različitim veličinama zaslona. Još jedna prednost su selektori pomoću kojih se cilja određeni element na stranici. Neki od vrsta selektora i selektora elemenata su: klasa, ID selektori i selektori atributa. Uz pomoć selektora omogućuje se primjena stilova na određene elemente ili grupe elemenata na stranici.

4.4. JavaScript

„JS“ (engl. JavaScript) je skriptni jezik koji omogućuje stvaranje sadržaja koji se dinamički ažurira. Također ima kontrolu nad multimedijom, animiranje slika itd. „JS“ omogućuje:

- Pohranjivanje korisnih vrijednosti unutar varijabli.
- Operacije na dijelovima teksta poput pridruživanje nekog niza varijabli.

- Pokretanje koda kao odgovor na određene događaje(engl. events) koji se događaju pri pokretanju „web“ stranice.

Programiranje vođeno događajima(engl. Event-driven programming) je nešto čime „JS“ omogućuje stvaranje responzivnih i interaktivnih „web“ aplikacija. „JS“ je potpuno opremljen objektno orijentirani programski jezik.^[12] Jedna od prednosti „JS-a“ su njegove biblioteke(engl. libraries) i „framework“ jer ima ogroman broj biblioteka i okvira, uključujući popularne kao što su „React“, „Angular“ i „Vue.js“. Uz pomoć biblioteka i „framework-a“ koji su dostupni, pruža snažne pomoćne programe koji olakšavaju izradu složenih aplikacija. Još jedna od prednosti „JS-a“ je njegova kompatibilnost s više platformi. Može raditi na širokom rasponu platformi, uključujući web preglednike, poslužitelje i mobilne uređaje. To omogućuje stvaranje aplikacija koje besprijekorno rade na različitim platformama i uređajima.

5. Izrada „web“ aplikacije

5.1. Instalacija „Vue.js-a“ i kreiranje projekta

Za izradu „Vue.js“ „web“ stranice potrebno je napraviti instalaciju „node.js“(<https://nodejs.org/en/download>) i „Vue.js“ te naravno imati „Visual Studio Code“(<https://code.visualstudio.com/download>). Potrebno je napraviti jednu datoteku gdje će taj projekt biti. Kad je datoteka kreirana, potrebno ju je otvoriti u „cmd-u“(engl. command prompt) i upisati komandu za instalaciju „Vue.js“ koja je vidljiva na slici 6.


```

C:\WINDOWS\system32\cmd.exe
C:\Users\Niko Novak\Desktop\webstranica>npm i -g @vue/cli
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated url@0.10.3: Please see https://github.com/lydell/url#deprecated
npm WARN deprecated apollo-datasource@3.2.2: The "apollo-datasource" package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-errors@3.1: The "apollo-server-errors" package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the "@apollo/server" package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated apollo-server-types@3.0.0: The "apollo-server-types" package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the "@apollo/server" package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-plugin-base@7.2: The "apollo-server-plugin-base" package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the "@apollo/server" package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated apollo-server-express@3.12.0: The "apollo-server-express" package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the "@apollo/server" package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-reporting-protobuf@3.4.0: The "apollo-reporting-protobuf" package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the "@apollo/reporting-protobuf" package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-env@2.1.1: The "apollo-server-env" package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the "@apollo/utils.fetcher" package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated subscriptions-transport-ws@11.0: The "subscriptions-transport-ws" package is no longer maintained. We recommend you use "graphql-ws" instead. For help migrating Apollo software to "graphql-ws", see https://www.apollographql.com/docs/apollo-server/data/subscriptions/#switching-from-subscriptions-transport-ws. For general help using "graphql-ws", see https://github.com/enisdenjo/graphql-ws/blob/master/README.md
npm WARN deprecated apollo-server-core@3.12.0: The "apollo-server-core" package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the "@apollo/server" package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
added 3 packages, removed 1 package, changed 855 packages, and audited 859 packages in 42s
65 packages are looking for funding
  run npm fund for details

0 vulnerabilities (1 moderate, 5 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run "npm audit" for details.

C:\Users\Niko Novak\Desktop\webstranica>vue -v
Usage: vue <command> [options]

Options:
  -V, --version          output the version number
  -h, --help             display help for command

Commands:
  create [options] <app-name>      create a new project powered by vue-cli-service
  add [options] <plugin> [pluginOptions]  install a plugin and invoke its generator in an already created project
  inspect [options] [paths...]      inspect the webpack config in a project with vue-cli-service
  serve                             alias of "npm run serve" in the current project
  build                             alias of "npm run build" in the current project
  ui [options]                     start and open the vue-cli ui
  init [options] [template] <app-name>  generate a project from a remote template (legacy API, requires @vue/cli-init)
  config [options] [value]          inspect and modify the config
  outdated [options]                (experimental) check for outdated vue cli service / plugins
  upgrade [options] [plugin-name]    (experimental) upgrade vue cli service / plugins
  migrate [options] [plugin-name]    (experimental) run migrator for an already-installed cli plugin
  info                               print debugging information about your environment
  help [command]                   display help for command

Run vue <command> --help for detailed usage of given command.

```

Slika 6. Prikaz „cmd-a“ kreiranje datoteke

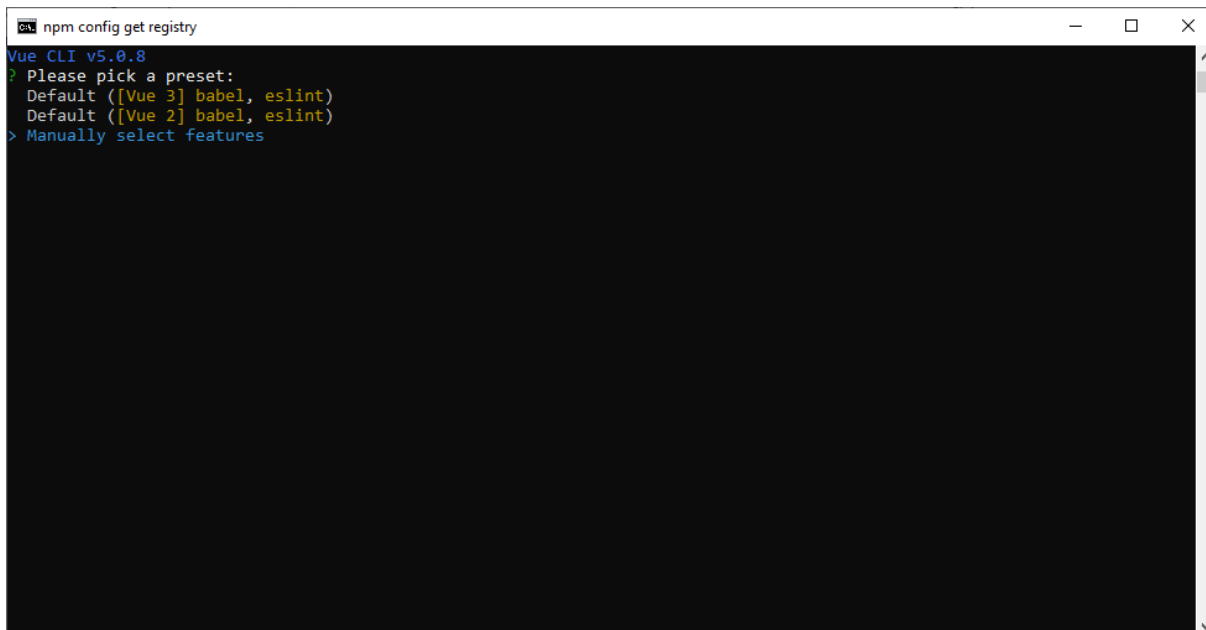
Izvor: Djelo autora rada

Prva komanda: **npm i -g @vue/cli**

- **npm i** → instalacija „npm-a“.
- **-g** → znači da želimo da komanda „Vue“ bude dostupna na cijelom računalu, globalno.
- **@vue/cli** → pruža naredbu „Vue“ u terminalu.

Druga komanda: **vue -v** → pomoću ove komande provjeravamo ako i koja verzija je „vue.js-a“ instalirana.

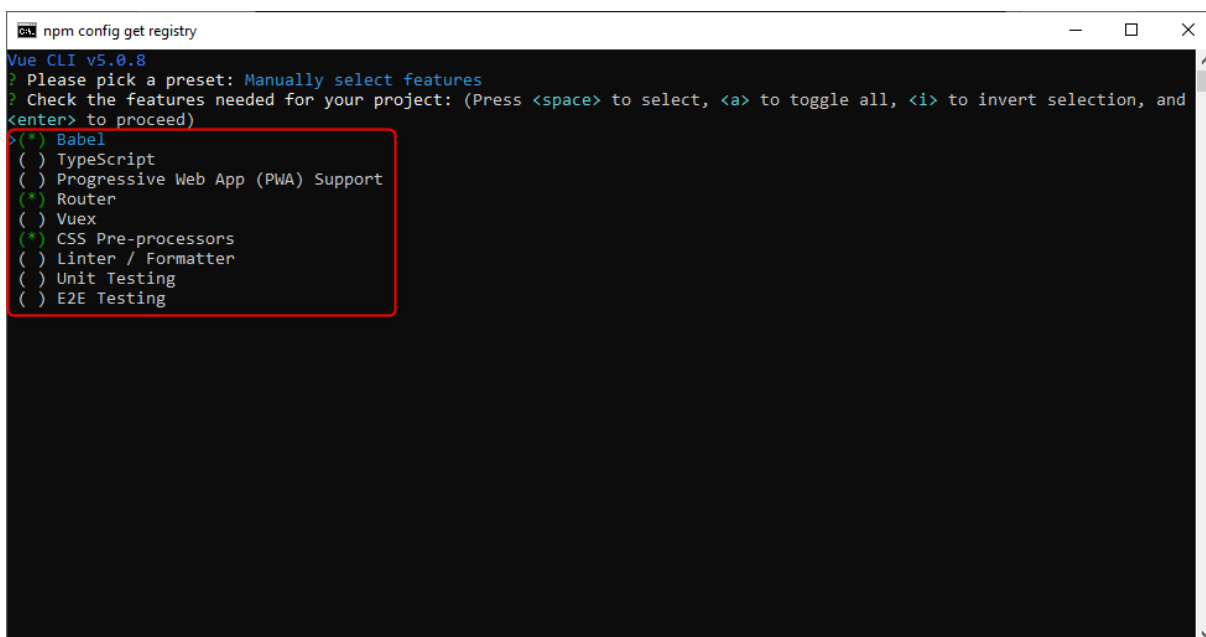
Nakon što su odrađene prve dvije komande, sada je potrebno kreirati projekt sa naredbom: **vue create <naziv_projekta>**. Nakon te komande u „cmd-u“ pokrenuta je konfiguracija gdje definiramo koje opcije želimo koristiti vidljivo na slici 7.



Slika 7. Prikaz konfiguracije „Vue.js-a“

Izvor: Djelo autora rada

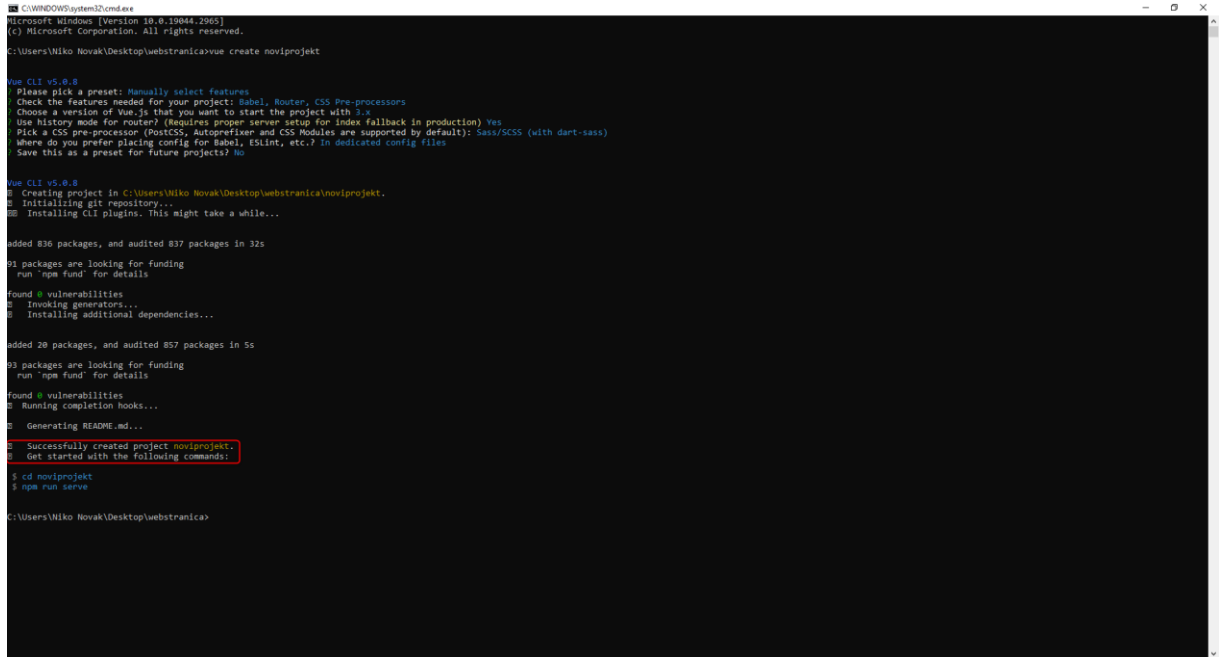
Potrebno „strelicom dolje“ odabrati „Manually select features“ kao na slici 7. i pritisnuti „enter“. Nakon toga odabrati tipkom razmak(engl. space): „Babel“, „Router“ i „CSS Pre-processors“ kao na slici 8.



Slika 8. Prikaz konfiguracije „Vue.js-a“ i njenih značajki

Izvor: Djelo autora rada

Pritisnuti tipku „enter“ nakon što su točne značajke(engl. features) odabrane. I dalje samo pritiskivati tipku „enter“ odnosno prihvaćamo zadanu(engl. default) konfiguraciju nakon koje kreće instalacija projekta. Na slici 8. je prikazana gotova instalacija projekta.



```
Microsoft Windows [Version 10.0.19044.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Miko Novak\Desktop\webstranica> vue create noviprojekt

Vue CLI v5.0.8
Please pick a preset: Manually select features
Check the features needed for your project: Babel, Router, CSS Pre-processors
Choose a version of Vue.js that you want to start the project with: 3.x
Use history mode for router? (Requires proper server setup for index fallback in production) Yes
Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): SASS/SCSS (with dart-sass)
Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files
Save this as a preset for future projects? No

Vue CLI v5.0.8
Creating project in C:\Users\Miko Novak\Desktop\webstranica\noviprojekt.
  Initializing git repository...
  Installing CLI plugins. This might take a while...

added 836 packages, and audited 837 packages in 32s
91 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
  Invoking generators...
  Installing additional dependencies...

added 20 packages, and audited 857 packages in 5s
93 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
  Running completion hooks...
  Generating README.md...

Successfully created project noviprojekt.
Get started with the following commands:

$ cd noviprojekt
$ npm run serve

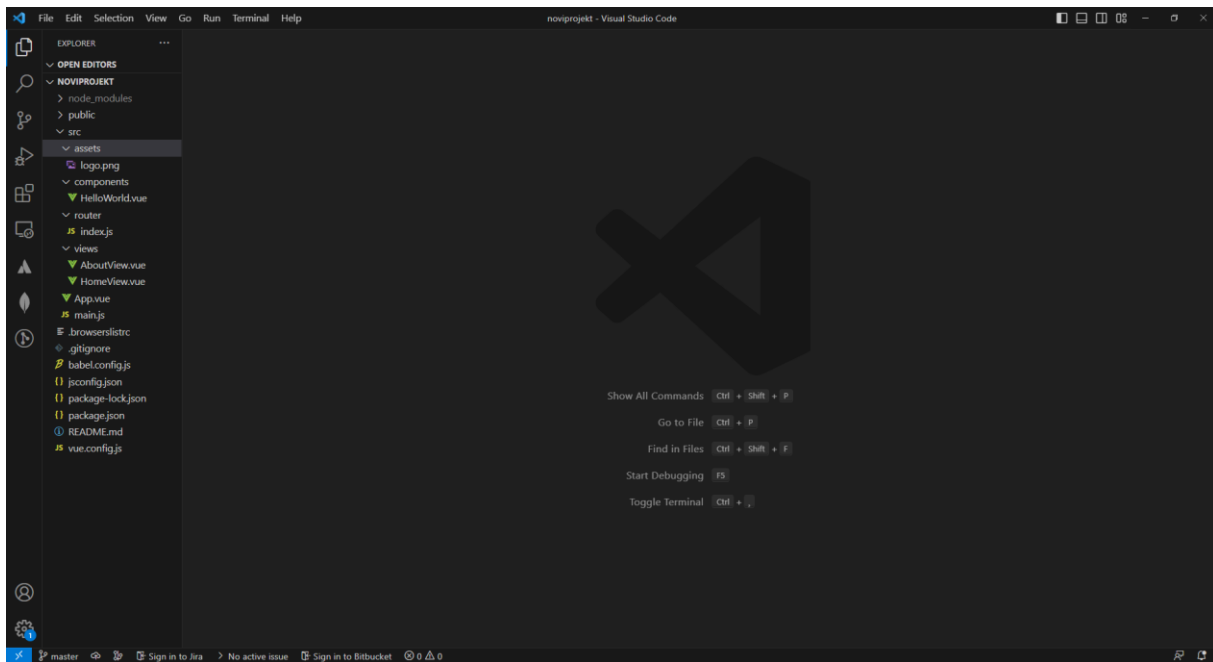
C:\Users\Miko Novak\Desktop\webstranica>
```

Slika 9. Prikaz uspješne instalacije projekta

Izvor: Djelo autora rada

Potrebno je otvoriti projekt uz „Visual Studio Code“. To se radi tako da se otvori kreirani projekt u „cmd-u“ te napiše komanda: **code** .

Na slici 10. se nalaze sve potrebne datoteke(engl. file).



Slika 10. Prikaz uspješno kreiranog projekta

Izvor: Djelo autora rada

5.2. Definiranje ruta „index.js“

Prvo što je potrebno bilo napraviti jest rute između različitih pogleda (engl. views). Pomoću „index.js“ datoteke, definiramo rute pomoću kojih se korisnicima prikazuju različiti pogledi. U ovom slučaju kreirana su tri pogleda, jedan pogled se naziva „Home.vue“, drugi „Skripta.vue“ i treći „Video.vue“. To znači da je u „index.js“ datoteci potrebno imati tri specificirane rute, a prikazane su na slici 11. zajedno sa kodom.

```

1 import { createRouter, createWebHistory } from 'vue-router'
2 import HomeView from '../views/Home.vue'
3 import Skripta from '../views/Skripta.vue';
4 import Video from '../views/Video.vue';
5
6 const routes = [
7   {
8     path: '/',
9     name: 'home',
10    component: HomeView
11  },
12  {
13    path: '/skripta',
14    name: 'Skripta',
15    component: Skripta,
16  },
17  {
18    path: '/video',
19    name: 'Video',
20    component: Video,
21  },
22 ]
23
24 const router = createRouter({
25   history: createWebHistory(process.env.BASE_URL),
26   routes
27 })
28
29 export default router
30

```

Sav kod "index.js" datoteke.

Terminal output: `Done Compiled successfully in 2991ms`

App running at:
 - Local: <http://localhost:8080/>
 - Network: <http://192.168.18.15:8080/>

Slika 11. Prikaz koda „index.js“ datoteke

Izvor: Djelo autora rada

Za definiranje jedne rute potrebno je unutar „const routers = [{ }]“ definirati: „path: ,name: ,component: “. „Path“ odabiremo naziv sami, no standard za početnu stranicu je '/'. „Name“ je većinom onako kako je i „component“, a to je naziv komponente odnosno datoteke. Nakon definiranja ruta unutar „const routers“, slijedi kod:

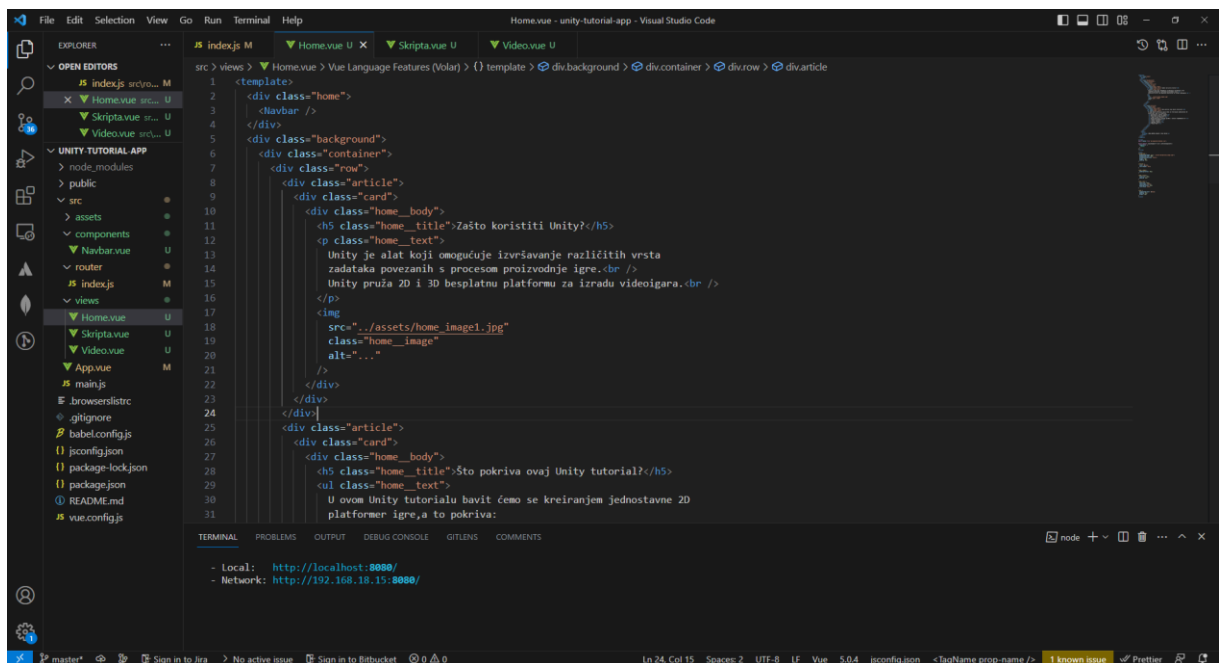
- Const router = createRouter({ → stvara novu instancu routera i dodjeljuje je konstantnoj varijabli „router“.
- history: createWebHistory(process.env.BASE_URL), → konfigurira način povijesti usmjerivača za korištenje funkcije „createWebHistory“ koji je određen varijablom „process.env.BASE_URL“. Time je postavljeno da „router“ koristi „HTML5 History API“ za obradu promjena „URL-a“.
- routes]] → specificira niz objekata rute koji definiraju rute za aplikaciju. Te rute se trebaju prikazati kada korisnik ide na različite „URL-ove“.

5.3. Komponente

Pogled je „web“ komponenta aplikacije koja predstavlja vizualnu prezentaciju podataka korisniku. Pružaju vizualno sučelje s kojima korisnici komuniciraju kada koriste neku „web“ aplikaciju, a sadržavaju elemente korisničkog sučelja kao što su „gumbi“, obrasci, tablice, slike itd.

5.3.1. „Home.vue“

Na slici 12. je prikazan kod prvog kreiranog pogleda nazvanog „Home.vue“.

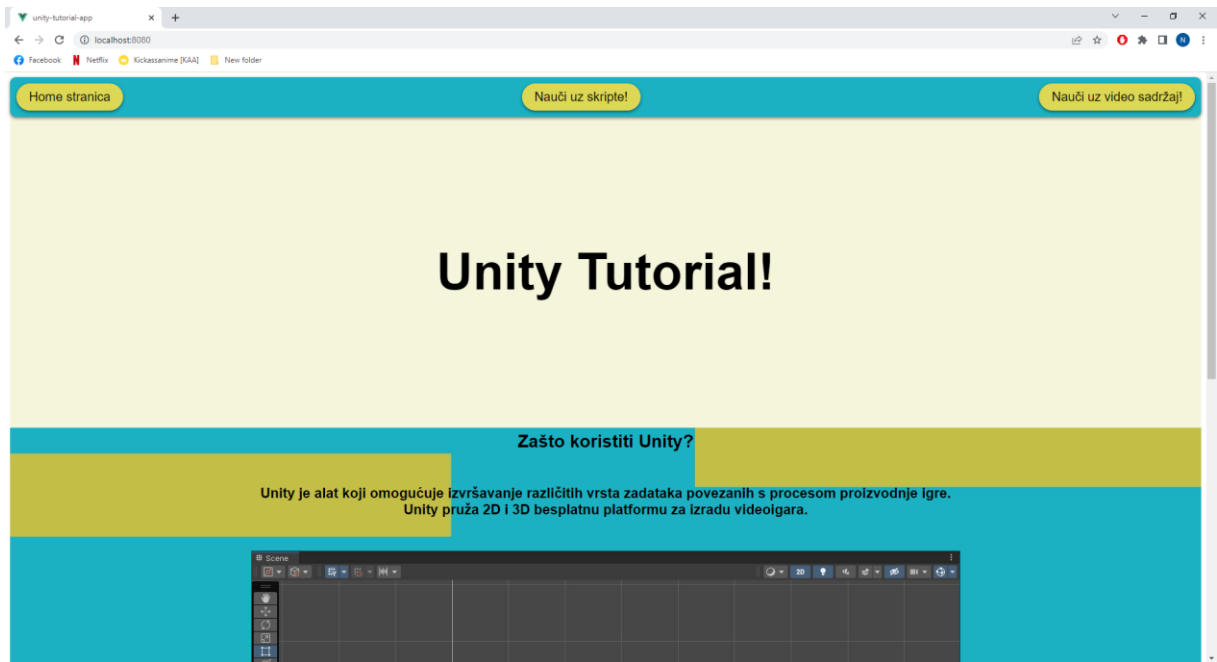


```
1 <template>
2 <div class="home">
3 <navbar />
4 </div>
5 <div class="background">
6 <div class="container">
7 <div class="row">
8 <div class="article">
9 <div class="card">
10 <div class="home_body">
11 <h5 class="home_title">Zašto koristiti Unity?</h5>
12 <p class="home_text">
13 Unity je alat koji omogućuje izvršavanje različitih vrsta
14 zadataka povezanih s procesom proizvodnje igre.<br />
15 Unity pruža 2D i 3D besplatnu platformu za izradu videoigara.<br />
16 </p>
17 
22 </div>
23 </div>
24 </div>
25 <div class="article">
26 <div class="card">
27 <div class="home_body">
28 <h5 class="home_title">Što pokriva ovaj Unity tutorial?</h5>
29 <ul class="home_text">
30 <li>U ovom Unity tutorialu bavit ćemo se kreiranjem jednostavne 2D
31 platformer igre, a to pokriva:
```

Slika 12. Prikaz koda pogleda „Home.vue“

Izvor: Djelo autora rada

Sastoji se od dvije cjeline kojima je klasa (engl. class) nazvana „article“. Svaki „article“ sastoji se od „home__title“ gdje se nalazi naslov cjeline, a zatim „home__text“ gdje se nalazi sadržaj te cjeline. Pogled „Home.vue“ ima rutu '/'. Na slici 13. je prikaz pogleda „Home.vue“.



Slika 13. Prikaz pogleda „Home.vue“

Izvor: Djelo autora rada

5.3.2. „Skripta.vue“

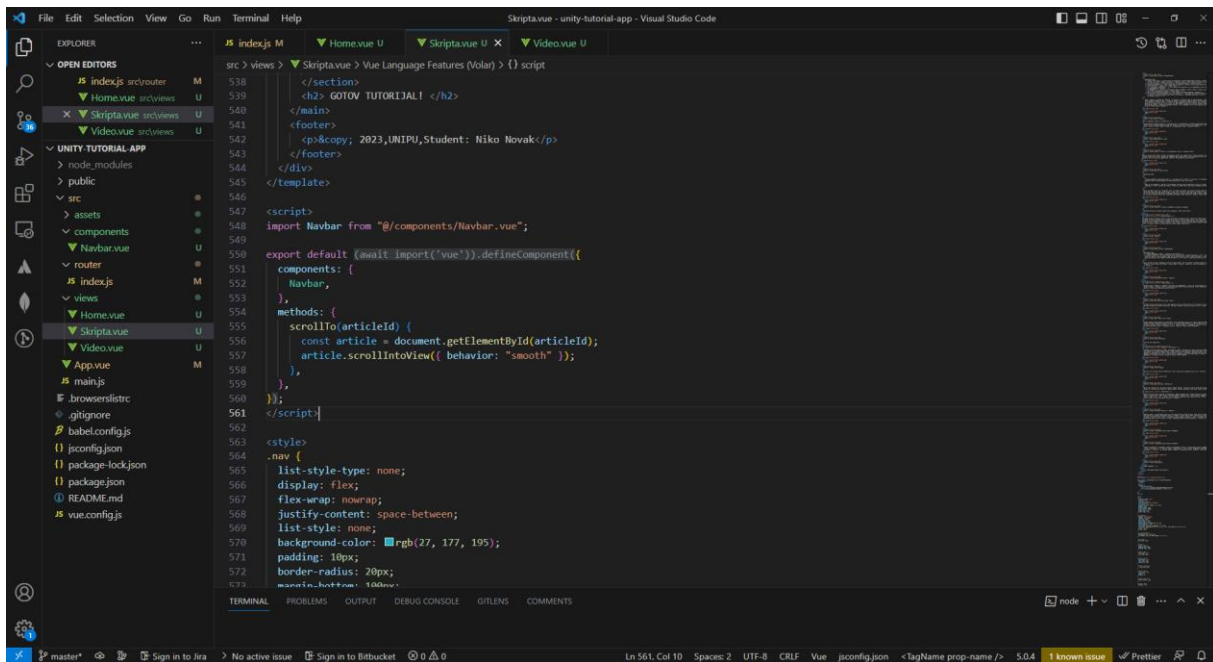
Druga kreirana komponenta je nazvana „Skripta.vue“. U njoj se nalazi „Unity Tutorial“, gdje korisnici mogu naučiti kako napraviti „2D Platformer“ igru. Ovaj pogled se sastoji od 6 članaka, te navigacijske trake (engl. navigation bar) uz pomoć koje korisnik sam odabire koje poglavlje želi proučiti. Na slici 13. je prikazan kod pogleda „Skripta.vue“.

```
1 <template>
2   <div class="container">
3     <header>
4       <nav>
5         <Navbar />
6       </nav>
7     </header>
8     <h2>Navigacijska traka za skriptu!</h2>
9     <ul class="nav">
10      <li class="nav_links">
11        <button class="nav_button" @click="scrollTo('article1')>
12          1.Uvod i instalacija alata "Unity"
13        </button>
14      </li>
15      <li class="nav_links">
16        <button class="nav_button" @click="scrollTo('article2')>
17          2."Import-anje Asset-a"
18        </button>
19      </li>
20      <li class="nav_links">
21        <button class="nav_button" @click="scrollTo('article3')>
22          3.Funkcionalnosti "gameObject-a"
23        </button>
24      </li>
25      <li class="nav_links">
26        <button class="nav_button" @click="scrollTo('article4')>
27          4."Camera movement"
28        </button>
29      </li>
30      <li class="nav_links">
31        <button class="nav_button" @click="scrollTo('article5')>
32          5."Scene loading"
33        </button>
34      </li>
35      <li class="nav_links">
36        <button class="nav_button" @click="scrollTo('article6')>
```

Slika 14. Prikaz koda pogleda „Skripta.vue“

Izvor: Djelo autora rada

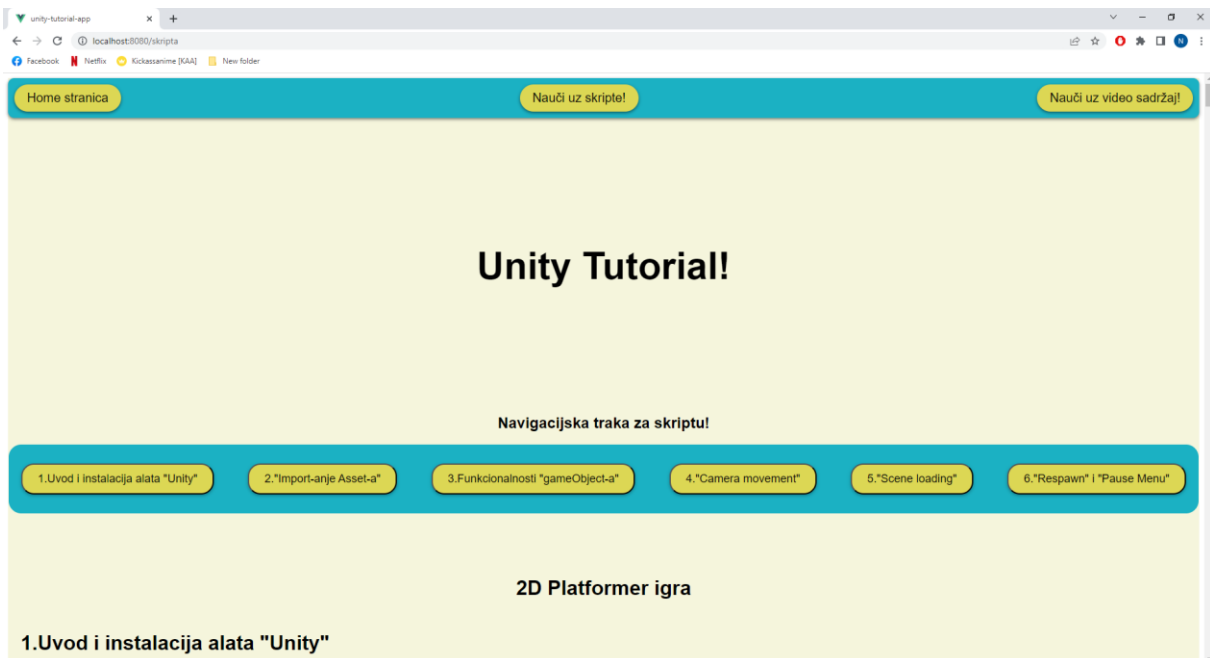
Prikazan je način na koji je navigacijska traka za skriptu kreirana. Postoji u „JavaScript“ dijelu metoda „scrollTo()“ koja pomoću „document.getElementById(articleId)“ definira na koji članak „gumb“ prebacuje korisnika. Iz tog razloga je svaki članak nazvan <article+broj>. Na slici 14. je prikazana metoda „scrollTo()“.



Slika 15. Prikaz „scrollTo()“ metode

Izvor: Djelo autora rada

Na slici 15. je prikaz kako izgleda pogled „Skripta.vue“, zajedno sa njenom rutom '/skripta'.



Slika 16. Prikaz pogleda „Skripta.vue“

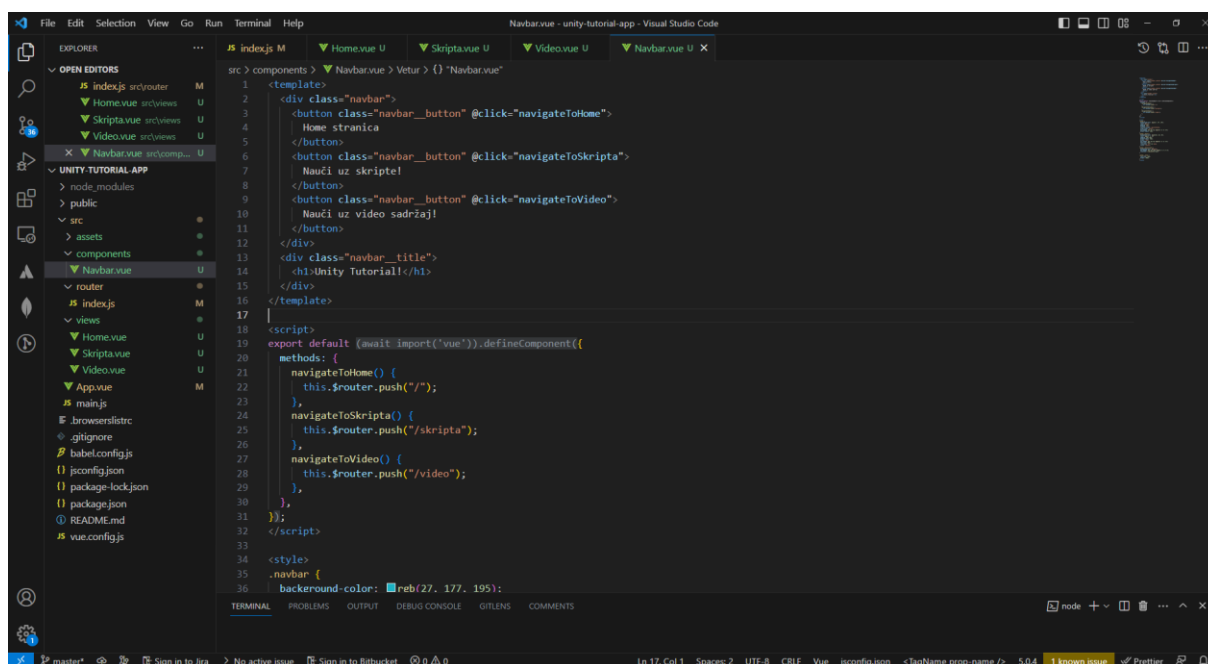
Izvor: Djelo autora rada

5.3.4. „Video.vue“

Posljednji kreiran pogled je „Video.vue“, koji sadržava kao i „Skripta.vue“ svoju posebnu navigacijsku traku, te dva videa. Jedan video je vezan za instalaciju alata „Unity“, a drugi video je vezan za kreiranje novog projekta. Linkovi za „Youtube“ nalaze se na „web“ aplikaciji.

5.3.5. „Navbar.vue“

Navigacijska traka koja vodi na različite poglede nazvana je „Navbar.vue“. Kreirana je kao zasebna komponenta iz razloga što je potrebna na svakom pogledu, te na ovaj način je skraćen kod. Sve što je potrebno da se pozove komponenta na pogled je: „<Navbar />“. Na slici 16. prikazan je kod komponente.



```
1 <template>
2   <div class="navbar">
3     <button class="navbar__button" @click="navigateToHome">
4       Home stranica
5     </button>
6     <button class="navbar__button" @click="navigateToSkripta">
7       Nauči uz skriptel
8     </button>
9     <button class="navbar__button" @click="navigateToVideo">
10      Nauči uz video sadržaj!
11    </button>
12  </div>
13  <div class="navbar_title">
14    <h1>Unity Tutorial!</h1>
15  </div>
16 </template>
17
18 <script>
19 export default (await import('vue')).defineComponent({
20   methods: {
21     navigateToHome() {
22       this.$router.push("/");
23     },
24     navigateToSkripta() {
25       this.$router.push("/skripta");
26     },
27     navigateToVideo() {
28       this.$router.push("/video");
29     },
30   },
31 });
32 </script>
33
34 <style>
35 .navbar {
36   background-color: #e627_177_195;
```

Slika 17. Prikaz komponente „Navbar.vue“

Izvor: Djelo autora rada

U „HTML“ dijelu komponente „Navbar.vue“ kreirana su 3 gumba klase „navbar__button“, koji imaju „onClick“(@click) funkciju koja klikom na gumb započinje određenu metodu. Prva metoda je „navigateToHome()“ koja pomoću komande

„this.\$router.push(„/“)“ preusmjeruje na pogled „Home.vue“. I tako za svaki pogled slijedi komanda „this.\$router.push(„naziv_rute“)“ kojom usmjeravamo koji pogled je prikazan. Kako izgleda komponenta „Navbar.vue“ se vidi na vrhu na slikama 13. i 15.

6. Zaključak

Uz pomoć praćenja interaktivnog tutorijala može se naučiti kako kreirati „Vue.js“ „web“ stranicu, te kako kreirati sadržaj da bi privukli korisnike. Tutorijal je o alatu „Unity“ uz pomoć kojeg se mogu napraviti 2D, 3D igrice, koje su u današnje vrijeme vrlo popularne. Iz tog razloga je kreiran tutorijal upravo o „Unity-u“, vrlo je korisno znati kako baratati s njime, te usput naučiti kako kreirati jednostavnu 2D platformersku igru sa svim bitnim objektima poput: glavnog igrača, neprijatelja, završnog cilja, te naravno kako nastaviti na sljedeći level da korisnik može daljnje uživati u igri. Što je još bitnije, upoznavanje sa „HTML-om“, „CSS-om“ i „JavaScript-om“ je veoma ključno za svakog programera koji se namjerava baviti kreiranjem „web“ stranica jer su ti jezici ključni za kreiranje „web“ aplikacija.

7. Popis slika

Slika 1. Prikaz glavnog zaslona "Unity Editor-a"	8
Slika 2. Primjer strukture „HTML-a“ u „Vue.js-u“	11
Slika 3. Primjer strukture „HTML-a“ u „Vue.js-u“ prikazan kao stranica	12
Slika 4. Prikaz „CSS-ove“ strukture u kodu.....	13
Slika 5. Prikaz dio komanda koje se koriste za „CSS“	14
Slika 6. Prikaz „cmd-a“ kreirane datoteke	16
Slika 7. Prikaz konfiguracije „Vue.js-a“	17
Slika 8. Prikaz konfiguracije „Vue.js-a“ i njenih značajki	17
Slika 9. Prikaz uspješne instalacije projekta	18
Slika 10. Prikaz uspješno kreiranog projekta	19
Slika 11. Prikaz koda „index.js“ datoteke	20
Slika 12. Prikaz koda pogleda „Home.vue“	21
Slika 13. Prikaz pogleda „Home.vue“	22
Slika 14. Prikaz koda pogleda „Skripta.vue“	23
Slika 15. Prikaz „scrollTo()“ metode.....	24
Slika 16. Prikaz pogleda „Skripta.vue“	24
Slika 17. Prikaz komponente „Navbar.vue“	25

8. Literatura

1. „Uvod u dizajn i multimediju“. Dostupno na:
https://www.academia.edu/22307903/Uvod_u_dizajn_i_multimediju
[Pristupljeno 18.04.2023.]
2. „Multimedija(MM)“, (2022.), Multimedija. Dostupno na:
https://www.academia.edu/10183143/Multimedija_MM [Pristupljeno 18.04.2023.]
3. „Unity User Manual“(2023.), Unity. Dostupno na:
<https://docs.unity3d.com/Manual/index.html> [Pristupljeno 18.04.2023.]
4. Vue.js Documentation,(2014.), Introduction. Dostupno na
<https://vuejs.org/guide/introduction.html> [Pristupljeno 18.4.2023.]
5. Božidar Kliček,(1996.), „Izgradnja multimedijских sustava“. Dostupno na:
https://www.academia.edu/35672651/Multimedijски_sustavi_skripta
[Pristupljeno 18.04.2023.]
6. w3schools.com(2022.),JavaScript Tutorial. Dostupno na
<https://www.w3schools.com/js/DEFAULT.asp> [Pristupljeno 19.04.2023.]
7. w3schools.com(2022.), HTML Tutorial. Dostupno na
<https://www.w3schools.com/html/> [Pristupljeno 19.04.2023.]
8. w3schools.com(2022.), HTML Styles – CSS. Dostupno na
https://www.w3schools.com/html/html_css.asp [Pristupljeno 19.04.2023.]
9. Olga Filipova(2016), „Learning Vue.js 2“. Dostupno na
https://books.google.hr/books?hl=hr&lr=&id=nszcDgAAQBAJ&oi=fnd&pg=PP1&dq=vue+js&ots=9oMeDkOesK&sig=9P9GsFQwBst8eRCj1-ktbGSSI00&redir_esc=y#v=onepage&q&f=false [Pristupljeno 05.05.2023.]

10. Astari S.(2022.), „What is HTML?“. Dostupno na https://www.hostinger.com/tutorials/what-is-html#How_Does_HTML_Work [Pristupljeno 08.05.2023.]
11. Mozilla.org(2023.), „CSS: Cascading Style Sheets“. Dostupno na <https://developer.mozilla.org/en-US/docs/Web/CSS> [Pristupljeno 08.05.2023.]
12. Mozilla.org(2023.), „What is JavaScript“. Dostupno na https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript [Pristupljeno 10.05.2023.]

9. Sažetak

Tema ovog završnog rada je upute za korištenje alata u obliku interaktivnog web tutorijala. Uvodni dio rada objašnjava što je to multimedija, koje vrste multimedije postoje te koje je najbolje koristiti da korisnici što više zapamte. Nadalje se objašnjava što je to Unity te kako izgleda njegovo korisničko sučelje te se opisuju svi njegovi pomoćni alati unutar tog sučelja. Nakon toga, opisuju se korišteni alati kod izrade web aplikacije, a to su programski jezici: Vue.js, HTML, CSS i JavaScript. Započinje se instalacijom određenih paketa koji su potrebni za Vue.js aplikaciju, nakon čega se objašnjava kod aplikacije. Prvo što se objašnjava su rute i kako ih pravilno postaviti. Nakon ruta, objašnjavaju se pogledi. U projektu postoje 3 pogleda, od kojih je jedan početni zaslon kad otvorimo aplikaciju, jedan pogled gdje bi se nalazili videi, te pogled pod nazivom „Skripta.vue“ na kojem se nalazi sva dokumentacija za izradu 2D platformerske igre. Svaki pogled ima objašnjen kod i objašnjenje za što se koristi taj pogled i kako izgleda.

Ključne riječi: interaktivni web tutorijal, Unity, Vue.js, kod, pogledi

10. Abstract

The topic of this final paper is instructions for using the tool in the form of an interactive web tutorial. The introductory part of the paper explains what multimedia is, what types of multimedia exist and which ones are best used so that users remember as much as possible. It further explains what Unity is and what its user interface looks like, and describes all its auxiliary tools within that interface. After that, the tools used in creating a web application are described, namely programming languages: Vue.js, HTML, CSS and JavaScript. It begins by installing certain packages that are required for a Vue.js application, after which the application code is explained. The first thing that is explained is the routes and how to set them up correctly. After the routes, the views are explained. There are 3 views in the project, one of which is the initial screen when we open the application, one view where the videos would be located, and a view called "Script.vue" which contains all the documentation for creating a 2D platformer game. Each view has code explained and an explanation of what that view is used for and what it looks like.

Keywords: interactive web tutorial, Unity, Vue.js, code, views