Izrada interaktivnog web mjesta za učenje sviranja gitare

Rajn, Karlo

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:122:590866

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-08-03**

Repository / Repozitorij:

University North Digital Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Sveučilište
Sjever

Završni rad br. 721/MM/2021

Izrada interaktivnog web mjesta za učenje sviranja gitare

Karlo Rajn, 0841/336

Varaždin, srpanj 2021. godine

Sveučilište
Sjever

Odjel za Multimediju, oblikovanje i primjenu

Završni rad br. 721/MM/2021

Izrada interaktivnog web mjesta za učenje sviranja gitare

Student

Karlo Rajn

Mentor

mr.sc. Vladimir Stanistavljević, dipl. ing.

Varaždin, srpanj 2021. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL Odjel za multimediju

STUDIJ preddiplomski stručni studij Multimedija, oblikovanje i primjena

PRISTUPNIK Karlo Rajn | MATIČNI BROJ 0841/336

DATUM 10.06.2021. | KOLEGIJ Programski alati 2

NASLOV RADA Izrada interaktivnog web mjesta za učenje sviranja gitare

NASLOV RADA NA ENGL. JEZIKU Creation of a web site for learnig playing guitar

MENTOR mr.sc. Vladimir Stanisavljević | ZVANJE Viši predavač

ČLANOVI POVJERENSTVA

1. mr.sc. Dragan Matković, v.pred - predsjednik
2. mag.mus. Predrag Krobot, pred. - član
3. mr.sc. Vladimir Stanisavljević, v.pred. - mentor
4. dipl.ing. Snježana Ivančić Valenko, v.pred. - rezervni član
5.

Zadatak završnog rada

BROJ 721/MM/2021

OPIS

Učenje pomoću video lekcija jedan je od vrlo značajnih doprinosa Interneta sveopćem obrazovanju. Izradi specijaliziranih web stranica za učenje koje sadrže video materijale može se pristupiti na razne načine. U Internetskim standardima HTML5 donosi nove mogućnosti neposrednog ugrađivanja videa u stranice. Popularne platforme za objavu video stranica također nude brojne mogućnosti izrade i postavljanja video tutorijala.
U radu je potrebno:
• analizirate mogućnosti prikaza video sadržaja pomoću nekih popularnih JavaScript razvojnim okvirima te na odabranim mrežnim stranicama poput YouTube-a
• usporediti mogućnosti pojedinih pristupa i kompatibilnost suvremenih internetskih preglednika
• izraditi demo stranicu na kojoj će se prikazati korištenje neko odabranog načina izrade stranice za učenje sviranja (gitare)
Detaljno opisati sve korištene tehnologije i korake u radu potrebne da bi se ostvarilo traženo te detaljno opisati stečena iskustva i postignute rezultate.

ZADATAK URUČEN 23.06.2021. | POTPIS MENTORA

REPUBLIKA HRVATSKA
SVEUČILIŠTE SJEVER

Sažetak

U ovom radu prikazan je način izrade interaktivnih videa u svrhu učenja sviranja gitare. Za izradu interaktivnih videa testirano je nekoliko metoda te se korištenje JavaScript-a zajedno s YouTube-om pokazala najboljom. Osim programskih alata objašnjene su metode učenja sviranja, glazbena teorija te alati za snimanje i obradu videa i glazbe.

Ključne riječi: interaktivni videi, edukacija, audio i video obrada

Abstract

This paper presents a way of making interactive videos for the purpose of learning to play the guitar. Several methods have been tested for creating interactive videos, and the use of JavaScript together with YouTube has proven to be the best. In addition to software tools, methods of learning music, music theory and tools for recording and processing video and music are explained.

Keywords: interactive videos, education, audio and video processing

Popis korištenih kratica

HTML HyperText Markup Language

Opisni jezik za opisivanje strukture web stranice.

CSS Cascading Style Sheets

Jezik za stiliziranje web stranice.

CMS *Content Management System*

Sustav za upravljanje sadržajem web mjesta.

WebVTT Web Video Text Tracks

Standard za prikaz tempiranog teksta u vezi s HTML5

API Application Programming Interface

Pravila za korištenje usluga i resursa operacijskog sustava ili programa

JS JavaScript

Skriptni programski jezik

Sažetak

1. 1

2. 3

2.1. Pristup za nelinearno učenje 5

2.2. Karakteristike web mjesta 5

2.3. Usporedba literature iz glazbene teorije: 5

3. Izrada sadržaja za web mjesto 6

3.1. Audio obrada 7

3.1.1. Alati za obradu gitara 7

3.1.2. Izrada glazbene podloge za video 9

3.2. Videoobrada 11

4. Web mjesto 13

4.1. Izrada web mjesta 13

4.1.1. CMS 14

4.1.2. Wordpress 14

4.1.3. Instalacija Wordpressa 15

4.1.4. Wordpress *pluginovi* 15

4.1.5. Instalacija WordPress *pluginova* 17

4.1.6. Stranice 17

4.2. Izrada interaktivnog videa 19

4.2.1. Izrada video *chaptera* 19

4.2.2. WebVTT 20

4.2.3. HTML kod 25

4.2.4. Testiranje metode 27

4.2.5. Prijenos videa na YouTube 28

4.2.6. Izrada poglavlja u YouTube videu 28

4.2.7. Spajanje s WordPress-om 30

4.2.8. Testiranje 31
4.2.9. Izrada JS skripte 32
4.2.10. JavaScript 32
4.2.11. iframe 32
4.2.12. API 33
4.2.13. Funkcija 33
4.2.14. Testiranje 36
4.3. Izgled web mjesta 37
5. Zaključak 40
6. Literatura 41
7. Popis slika 43

1. Uvod

U svoje slobodno vrijeme bavim se glazbom. Gitaru sviram oko 14 godina te sam većinom samouk zahvaljujući internetu. Razvojem interneta postepeno se povećavala količina sadržaja za učenje glazbe. Pogotovo u nazad zadnjih par godina guitar community je toliko narastao da je moguće naučiti svaku sitnicu vezanu uz gitaru. Tečaja ima pregršt, primjetio sam da skoro svaki veći YouTube *content creator* ima vlastiti tečaj iako nauka nije u fokusu njegovih videa. Uglavnom tečaji ovakvih *content creatora* su namjenjeni za početnike ali uz njih i postoje kanali koji su striktno posvećeni samo online lekcijama i svoje usluge nude potpuno besplatno. Tada bi se moglo postaviti pitanje zašto bi se plaćao tečaj kada postoje besplatne opcije?

Kada sam pohađao srednju školu u jednom trenutku sam odlučio krenuti ozbiljno vježbati gitaru. Htio sam se posvetiti učenju glazbene teorije jer sam primjetio da su svi gitaristi koji su bolje svirali od mene poznavali do neke mjere glazbenu teoriju. Raspitao sam se koliko koštaju privatne instrukcije jer je u to vrijeme postojala opcija za online učenje ili učenje jedan na jedan. Ubrzo sam shvatio da su lekcije uživo veoma skupe te sam isprobao svoju sreću s online učenjem. Besplatna opcija za mene nije funkcionirala pošto nisam imao nikakvih predznanja o glazbenoj teoriji i iako je postojalo na stotine literatura, nije mi bila od koristi pošto nisam znao ni gdje početi. U konačnici sam krenuo na privatne lekcije koje sam prestao pohađati nakon manje od godinu dana. Razlog tome je što sam savladao osnove te sam tada mogao vidjeti što mi je potrebno za „sljedeću razinu“. I besplatan online sadržaj je imao smisla.

Jedna od prvih lekcija kada sam pohađao instrukcije 1 na 1 bila je razlika u nomenklaturi tonova. Naime, u našoj hrvatskoj litaraturi postoji ton H, dok u većini ostale svjetske literature ne postoji. Hrvatska nije jedina na svijetu koja koristi H kao ime tona pošto sam primjetio da i u nekih drugim europskim državama ponekad ispliva H umjesto B. Ova mi je razlika u početku znala uzrokovati nejasnoće pošto nisam u početku znao na što se ton B referencira pošto u našoj i stranoj literaturi B označava 2 različita tona. No na sreću imao sam nekoga da mi pojasni kada nisam bio siguran pošto sam pohađao instrukcije.

Kasnije kada sam prestao pohađati instrukcije znao sam često posjećivati web mjesta sa literaturom za glazbenu teoriju. Literature ima za sve razine znanja i u svim mogućim oblicima. Videolekcije, knjige, note, TAB zapisi, vježbe itd, ali sva literatura je bila na engleskom. Jednom prilikom sam iz znatiželje išao googlati literaturu na hrvatskom da vidim koliko materijala je dostupno našem govornom području. Na žalost jako je malo web mjesta tada

postojalo. Bilo je nekolicina web stranica koje su imale materijale za učenje gitare ali opširnost i kvaliteta istih nije se niti mogla mjeriti sa stranim web mjestima. Također sve stranice su bile otprilike na istoj razini znanja. Upravo iz tog razloga sam bio potaknut napraviti svoje web mjesto gdje bih objašnjavao koncepte koje nisu pokrile ostale stranice te objasnio razlike u glazbenoj teoriji kako bi prelazak na stranu literaturu bio lakši. Također moja literatura bi bila tako konstruirana da korisnici ne moraju kronološki učiti lekcije već bi svaka lekcija bila zasebno poglavlje koje ne zahtjeva znanje prijašnjih.

U sljedećem poglavlju naveo sam i objasnio postojeće metode za glazbenu edukaciju i kako su se razvijale tokom povijesti. Za svoj pristup učenja držao sam se principa pedagogije moderne glazbe.

2. Metode učenja

Dalcroze metoda

Dalcroze metoda je razvijena u ranom 20. stoljeću od strane Švicarskog glazbenika i predavaća Émile Jaques-Dalcroze-a. Metoda je podijeljena u 3 područja – solfeggio, improvizacija i eurhythmics. Eurhythmics se još naziva ritmičkom gimnastikom pošto obuhvaća učenje ritma, strukture i glazbenog izražaja uz pomoć pokreta što čini Dalcroze metodu prepoznatljivom. Prema Dalcroze metodi glazba je temeljni jezik ljudske svijesti te je tako bitan činbenik naše osobnosti. [1]

Kodály metoda

Zoltán Kodály bio je mađarski učitelj i kompozitor, njegova metoda se zasnivala na zabavnom učenju osnova glazbene teorije i notacije. Glavni cilj je stvoriti cijeloživotnu ljubav prema glazbi. Smatrao je da je glazbena edukacija potrebna u svim školama. Danas se u praksi koriste neke od Kodályovih metoda kao što pokazni znakovi za solfeggio, standardna glazbena notacija te verbalizacija ritma. [2]

Orff Schulwerk

Orff Schulwerk se smatra pristupom u glazbenom edukaciji nezvan prema Carl Orffu – Njemačkom kompozitoru. Edukacija započinje uključivanjem učenika u osnovne ritmičke i melodijske vježbe. Orff je vjerovao da je ljudsko tijelo poput prekusivnog instrumenta te da učenici imaju potencijal razviti svoje jedinstvene glazbene sposobnosti. Njegov pristup potiče samostalan razvoj i improvizaciju te smanjuje pritisak u glazbenom razvoju. Orff je također izumio i modificirao nekoliko perkusivnih instrumenata poput *glockenspiel, xylophone* i *metallophone*. [3]

Suzuki metoda

Shinichi Suzuki je izumio tu metodu po završetku drugog svjetskog rata. Svoje glazbeno obrazovanje je koristio kako bi podigao moral svojih učenika. Suzuki je smatrao da se sva djeca mogu glazbeno educirati u idealno okruženju. Nastojao je stvoriti okruženje u kojemu bi postigao koncentraciju koja imaju osobe dok uče svoj materinji jezik. Njegova metoda se sastoji od velike količine primjera, vježbe, pohvale, poticanja te jedinstvenom pristupu prema svakom učeniku za tehniku koju uči. [4]

Gordonova metoda

Edwin Gordan je smatrao da se glazbena edukacija treba temeljiti na opsežnom istraživanju i terenskim testiranjima. Njegova metoda potiče diskriminacijsko učenje i zaključivanje. Diskriminacijsko učenje potiče analizu jesu li dva elementa ista ili ne, putem verbalne asocijacije, djelomične sinteze, simboličke povezanosti i složene sinteze. Uz učenje zaključivanja učenici sami uče prepoznavati, stvarati i improvizirati. Također postoji i teorija učenja novorođenčadi i male djece. [5]

Svjetska glazbena pedagogija

Svjetsku glazbenu pedagogiju proučavala je Patricia Shehan Campbell kako bi opisala svjetske glazbene sadržaje i praksu u glazbenim programima osnovnih i srednjih škola. Cilj je bio osmisliti model učenja glazbe različitih razina i specijalizacija. Pedagogija potiče uporabu ljudskih resursa te kontonuirano slušanje. [6]

Govorni solfeggio

John M. Feierabend je pod utjecajem Kodály i Gordonove metode razvio govorni solfeggio. Bio je predsjednik Hartt School-a te University of Hartford-a. Glazba se smatra bitnijom od same notacije iste. Program se sastoji od dvanaest faza u kojima se uči slušanje, pjevanje i dekodiranje glazbe, a kasnije i sam zapis glazbe. [7]

Carbo-cone metoda

Metodu je razvila violinistica Madeleine Carabo-Cone. Metoda se također ponekad naziva osjetilno-motorni pristup glazbi. Njezina metoda uključuje korištenje rekvizita, kostima i igračaka kako bi djeca naučila osnovne glazbene koncepte. [8]

MMCP

Manictanville Music Curriculum Project je razvijen 1965. godine. Osnovan je kako bi se podigla zanimacija sve manjeg broja polaznika glazbene edukacije. Cilj metode je da učenici ne shavaćaju glazbu kao statički sadržaj koji terba savladati. Metoda potiče samostalno istraživanje i proučavanje. Učitelj daje određeni glazbeni problem koji učenici u grupama trebaju riješiti putem brainstorminga. [9]

Pedagogija popularne glazbe

Razvijena 1960-ih metoda se također naziva i rock glazbena pedagogija. Metoda uči rock glazbu te potiče grupne improvizacije umjesto treadicionalne individualne. [10]

2.1. Pristup za nelinearno učenje

Kada sam odlučio izraditi web mjesto s programom prilagođenim za naše govorno područje bio sam vođen mišlju kako bi bilo korisno kada bi korisnik smjesta mogao birati lekciju koja ga zanima, a ne da treba prvo savladati prethodne kako bi došao do željene. U takvim situacijama učenici u većini slučajeva gube entuzijazam za učenjem.

Zamisao je da svaka lekcija nema veze sa prijašnjom ili sljedećom u smislu da ne zahtjeva informaciju koja je prethodno bila navedena. U takvom scenariju moguće je napraviti pristup gdje će korisnik moći birati lekcije koje su njemu prioritet ili samo one koje nije savladao.

Naravno da će predznanje biti potrebno kako bi korisnik shvatio lekcije i kako bi nelinearni koncept bio mogući. Navedeno predznanje je moguće naučiti s ostalih stranica. Na svom web mjestu fokusirao sam se na nastavak gradiva koji nije bio obuhvaćen.

2.2. Karakteristike web mjesta

Moje web mjesto zahtjeva: informacije za što je web mjesto namijenjeno, zbirku videa (lekcija), mogućnost preskakanja videa po poglavljima pošto se radi o dužim lekcijama, zbirku zapisanih koncepata u TAB obliku za korisnike koji to preferiraju te naravno kontakt formu. Same lekcije bi bile podijeljene u tri skupine. Prva skupina lekcija bi služila za učenje metodike improvizacije i poticala bi učenika na vlastito razvijanje prezentirane vještine. U drugoj skupini prezentirale bi se gotovi materijali koji se mogu smjesta koristiti za vlastite improvizacije, te na kraju u trećoj skupini upozorio bih učenike na moguće probleme, poteškoče i loše navike koje se mogu razviti u samom procesu učenja sviranja gitare te kako ih spriječiti.

2.3. Usporedba literature iz glazbene teorije:

Glazbena teorija u principu je ista ali postoje dvije nomenklature tonova. U hrvatskoj literaturi se koristi nomenklatura koja je rijetko prisutna u stranim programima.

Primjer 1:

Strana nomenklatura:

C C# D D# E F F# G G# A A# B

Domaća nomenklatura:

C C# D D# E F F# G G# A A# H

Primjer 2:

C dur ljestvica

Strana nomenklatura:

C D E F G A B

Domaća nomenklatura:

C D E F G A H

U ova dva primjera vidimo kako za dvanaesti ton kromatske ljestvice i sedmi ton durske ljestvice imamo različito ime tona. Nadalje ukoliko ton H snizujemo on će se u nekim situacijima zvati ton B. U tom slučaju ukoliko osoba nije pohađala glazbenu školu mogla bi zamjeniti sniženi H za obični H pošto u stranoj literaturi B je dvanaesti ton a u našoj literaturi jedanaesti ton kromatske ljestvice. Navedeni problem može dovesti do brojnih nesporazuma u komunikaciji ako niste svjesni razlike. [11]

3. Izrada sadržaja za web mjesto

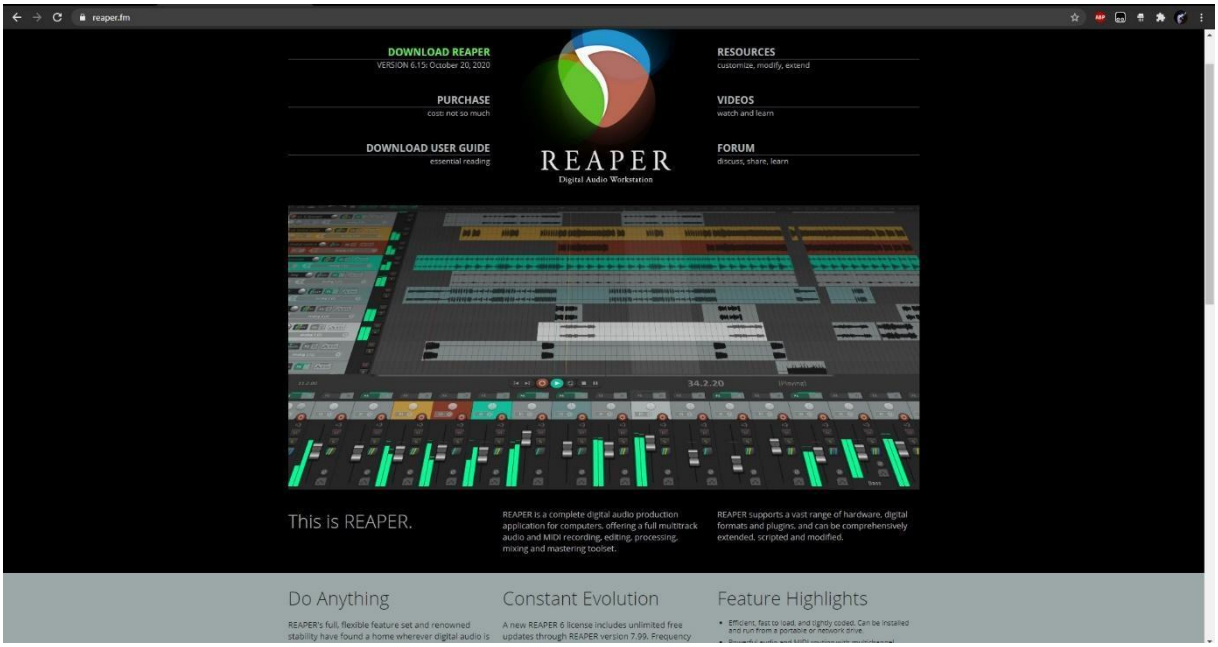
Za izradu materijala koji će biti objavljeni na web mjestu koristit će se: videokamera, Adobe Premiere Pro, Adobe Photoshop, audio interface, DAW (*digital audio workstation*) *software*, gitara, bass gitara i drum *software* u ovom slučaju EZDrummer.

Prilikom snimanja videolekcija zasebno se snima video i zvuk instrumenta. Za snimanje instrumenta potrebni su nam audio interface i DAW *software*. Audio interface je eksterna zvučna kartica. U sebi ima predpojačala i radi konverziju signala iz analognog u digitalni kako bi kvaliteta zvuka bila što veća. Iako postoje USB mikrofoni kod njih je količina šuma veća te ne mogu pružati istu kvalitetu snimljenog zvuka kao eksterna zvučna kartica. Također uz pomoć zvučne kartice u mogućnosti smo snimiti *clean* signal instrumenata te ih je lakše manipulirati u kasnijoj produkciji. [12]

3.1. Audio obrada

Moj odabir za DAW *software* je Reaper. Reaper je digitalna audio radna stanica i program MIDI sekvencera koji je kreirao Cockos. Glavna prednost je što nije potrebno platiti licencu za korištenje te sa programom dolazi većina potrebnih alata za audio obradu.

Reaper je moguće besplatno preozeti sa službene stranice: https://www.reaper.fm/



Slika 3.1. Službena stranica za preuzimanje DAW-a

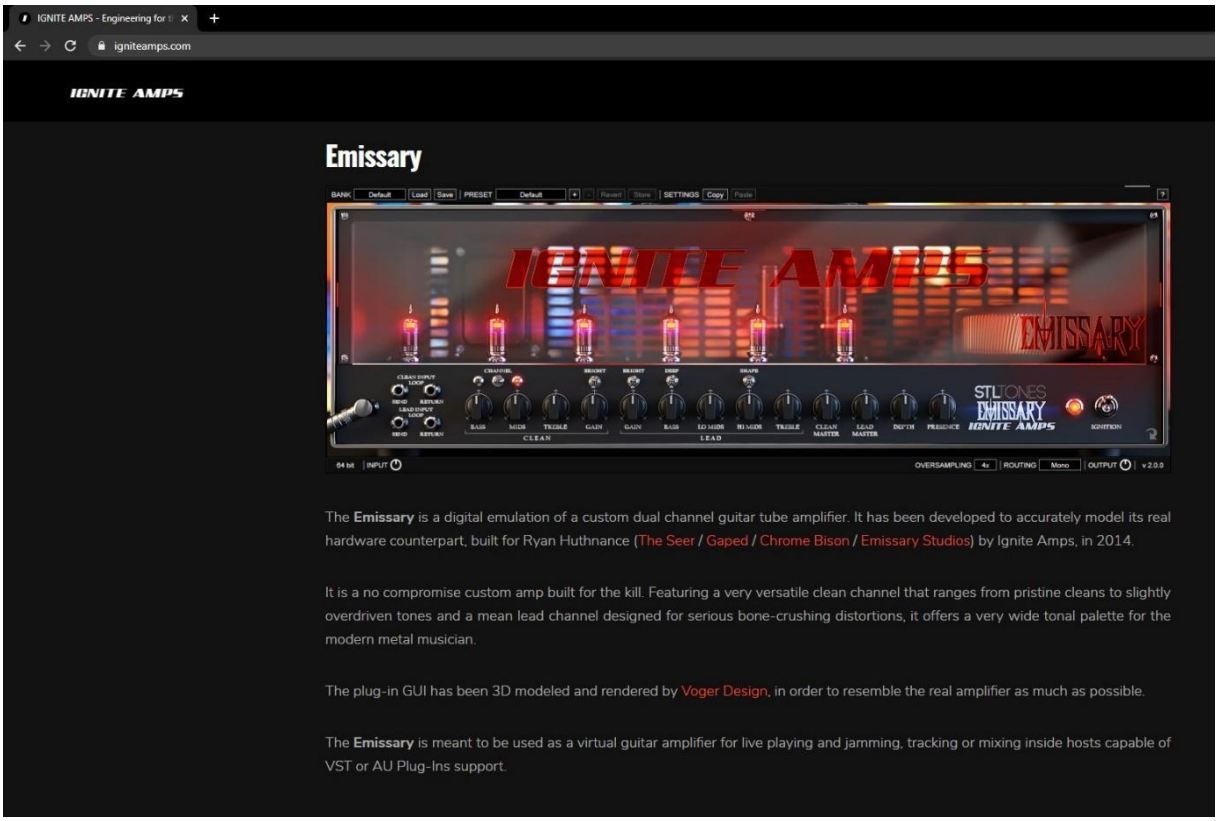
Sljedeći alati potrebni za audio obradu su digitalne simulacije pojačala. Zbog jednostavnost korištenja i naknadne obrade gitara odlučio sam se za digitalnu varijantu. Kada bi snimali pojačala bez digitalnih simulacija osim navedenih alata potrebno bi nam bilo gitarsko pojačalo i mikrofon za ozvučavanje. Snimljena analogna pojačala nije moguće nakon snimanja toliko fleksibilno mjenjati kao digitalna. Razlog tome je što za digitalna pojačala snima se samo „suhi“ ili DI (*direct input*) signal. Taj signal *pluginovi* obrađuju i kontrole za zvuk pojačala se mogu mijenjati u bilo kojem trenutku dok kod pravih pojačala to više nije moguće. Ukoliko želimo promjeniti gain (količinu distorzije u pravom pojačalu) ili EQ pojačala, potrebo je ponovo snimiti gitare. Iz ovih razloga sam odlučio koristiti *pluginove* za procesuiranje gitara te još jedan dodatni razlog je to što danas ima pregršt besplatnih simulacija. [13]

3.1.1. Alati za obradu gitara

Nakon što snimimo DI signal gitara u postprodukciji možemo simulirati zvuk pravog pojačala. Za to su nam potrebni sljedeći VST *pluginovi*; simulacija pojačala, *impulse response*

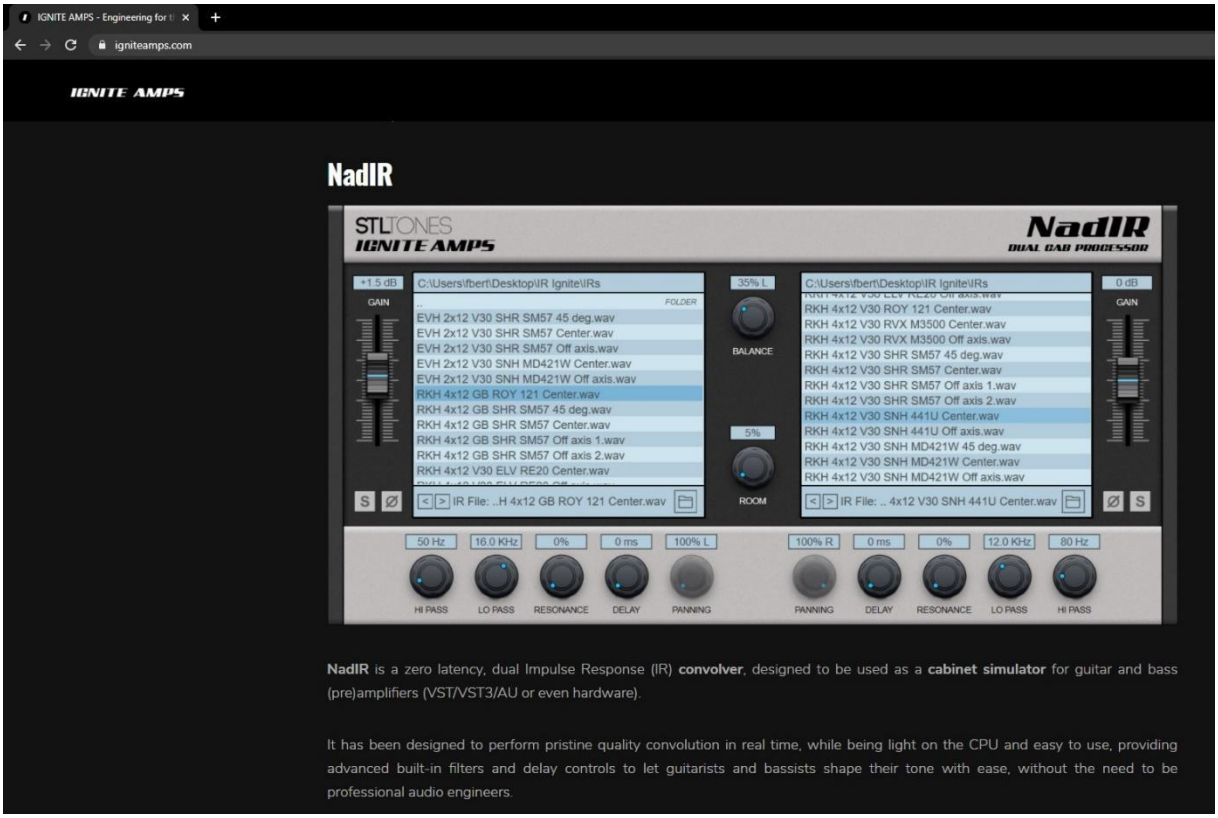
loader i *impulse response file*. Kako bismo shvatili što svaka komponenta predstavlja prvo trebamo objasniti kako izgleda „klasično“ snimanje gitara. Prije korištenja digitalnih simulacija za snimanje osim gitare bili su potrebni pojačalo (glava), zvučna kutija (kabinet) i mikrofon koji je bio uključen u *preamp* ili zvučnu karticu. U našem slušaju kao što ime govori simulacija pojačala simulira zvuk glave a što se tiče kabineta tada je potrebno objasniti što je IR (*impulse response*). IR su sve promjene koje su se desile na *imput* signalu djelovanjem vanjskih čimbenika. Gitarski IR-ovi su rađeni na način da se kabinet ozvuči sa mikrofonom i kroz njih pusti *white noise*. *White noise* je zvučni zapis svih svih frekvencija u istoj glasnoći. Nakon što se *white noise* pusti kroz kabinet i mikrofon glasnoća frekvencija će se promjeniti na specifičan način s obzirom na komponente kabineta, mikrofona i pozicije mikrofona na kabinet. Iz te promjene koju daju kabinet i mikrofon moguće je definirati koje frekvencije i za koliko bi promjenili bilo koji signal. Laički rečeno IR je zapravo digitalni snimak kabineta i mikrofona u određenoj poziciji snimanja. Svaki IR je zasebna datoteka koja ima jedinstveni snimak kobinacije kabineta, mikrofona i ostalih čimbenika koji su utjecali na mjenjanje signala, drugim riječima svaki IR je unikatan. [14]

Moj izbor za digitalno pojačalo je Ignite Amps Emissary koje je moguće besplatno preuzeti sa: https://www.igniteamps.com/



Slika 3.1.1. Digitalna simulacija pojačala

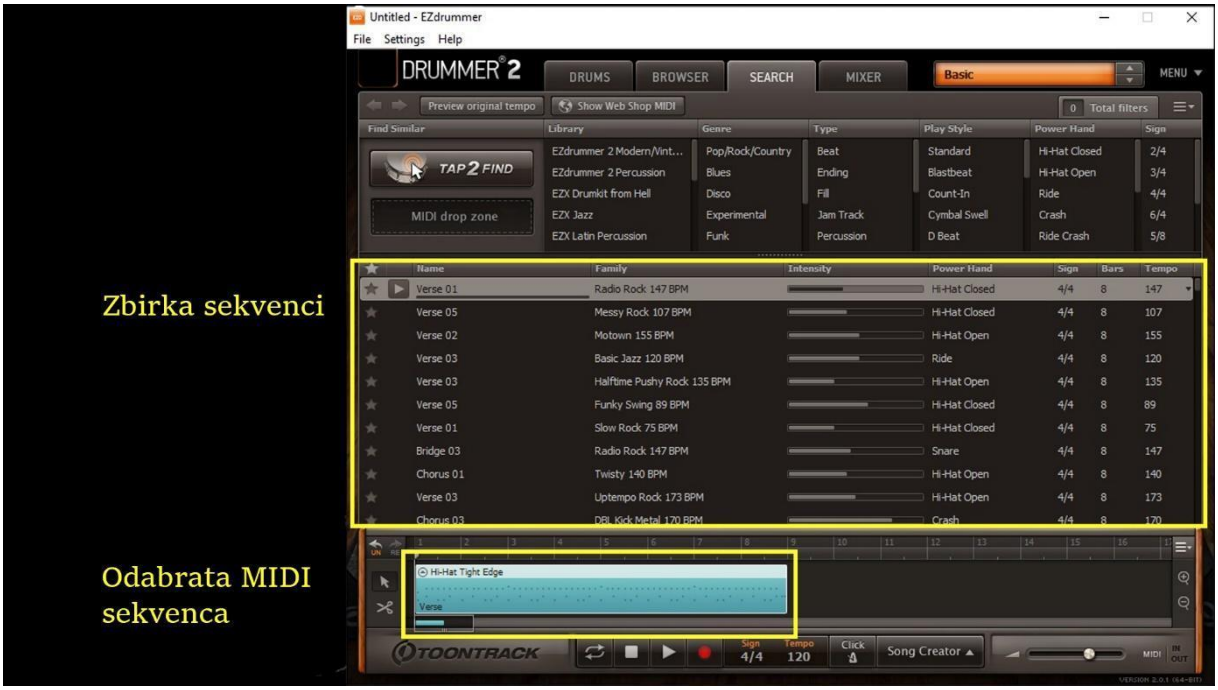
Za IR loader koristio sam Ignite Amps NadIR. IR loader nam omogućava učitavanje IR-ova koji procesiraju signal simulacije pojačala. Bez toga zvuk bi bio veoma grub i često neugodan za slušanje.



Slika 3.1.1. Impulse response loader

3.1.2. Izrada glazbene podloge za video

Prvi korak u izradi glazbene podloge je bio izraditi ritam za bubnjeve. Na sreću danas više nije potrebno imati akustični set bubnjeva i svu opremu za snimanje već je dovoljno imati drum *software* poput EZDrummer-a. EZDrummer sadrži zbirku profesionalno snimljenih udaraca te se uz pomoć MIDI-a može programirati željena sekvenca udarca. MIDI je skraćeno od Musical Instrument Digital Interface te je standard koji opisuje komunikacijski protokol, digitalno sučelje i električne konektore. On zapravo služi kao jezik kojim možemo opisati koji ton želimo da instrumet ili *software* odsvira i na koji način. Srećom EZDrummer ima već unaprijed spremljene sekvence koje možemo izabrati i dodatno modificirati tako da nije potrebno ručno programirati sekvence ukoliko nađemo nešto zadovoljavajuće u zbirci sekvenci. [15]



Slika 3.1.2. EZdrummer software

Kada smo izradili sekvencu za bubnjeve potrebno je snimiti gitare i bass. Instrumente snimamo na način da u zvčnu karticu uključimo instrumentalni kabel te odredimo količinu *gaina* na kartici. Svijetlo na *inputu* ne smije prelaziti u crveno jer tada je signal preglasni što rezultira u *audio clippingu*, drugim riječim u distorziji signala. Na slici niže je prikazano kako to izgleda.

Slika 5.1.2. Prikaz signala koji clippa

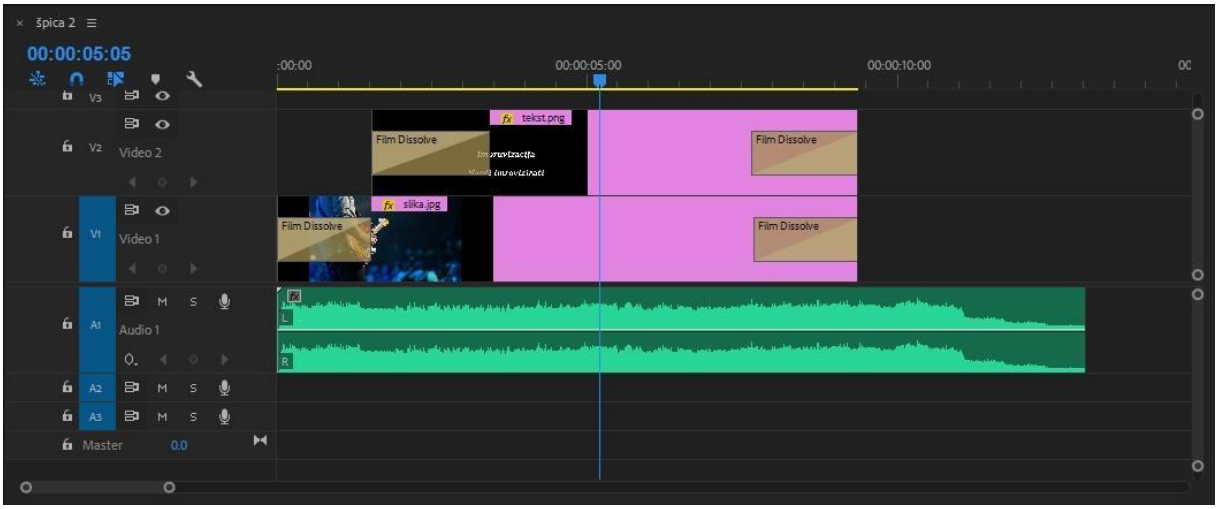
Po završetku kada smo snimili sve instrumente potrebno je napraviti mix. Gitare stavljam u digitalne simulacije pojačala kako je prije bilo objašnjeno, na bass gitare *compressor* da izjednači dinamičke oscilacije, a na bubnjeve nije potrebno stavljati ništa pošto su EZDrummerovi *sampleovi* „*mix ready*“. U suštini potrebno je samo podesiti glasnoće, a na gitare sam stavio high pass filter koji reže nepotrebne niske frekvencije kako bi se bass gitara lakše probila u miksu.

3.2. Videoobrada

Za video obradu koristio sam Adobe Premiere Pro. Prvi korak bio je izraditi uvodnu špicu za početak svakog videa. Za špicu sam snimio jednu kratku frazu na gitari koju sam obradio u Reaperu na isti način kao što je opisano u prethodnom poglavlju. Zatim sam trebamo sliku koja će služiti za pozadinu. Sliku sam preuzeo sa https://pixabay.com/ . Pixabay je jedno od najpopularnijih web mjesta za preuzimanje slika, videa, ilustracija, vektorske grafike i glazbe slobodne za uporabu. Nakon što sam našao sliku preuzeo sam je i obradio u Adobe Photoshopu. Ondje sam definirao veličinu dokumenta 1920 x 1080p kako bi slika odgovarala formatu videa. Sliku sam stavio u prvi layer, a tekts u drugi. Prvi layer sam zasebno spremio

kao JPG format, a tekst u PNG kako bi pozadina mogla biti transparentna. Razlog tome je što sam želio da u uvodnoj špici napravim animaciju na taj način da prvo slika *fade in-a*, a zatim tekst. Da sam eksportao sliku zajedno sa tekstom to ne bi bilo moguće.

Kada sam izradio sav materijal za uvidnu špicu pokrenu sam Adobe Premiere Pro i učitao zvučni zapis, pozadinsku sliku i transparenti tekst. Montažu sam napravi na način da sam kreirao dvije video trake. U prvoj se nalazi pozadinska slika, a u drugoj (iznad prve) se nalazi transparentni tekst. Druga traka s tekstom je kraća i počinje kasnije kako bi se prvo prikazala slika u videu, a zatim tekst te završavaju u isto vijeme kako bi zajedno nestale. Na početku i kraju svake trake je efekt za tranziciju. Video efekti se postavljaju na način da u kartici „*Effects*“ odaberemo mapu „Video transitions“. U toj mapi imamo zbirku podmapa s različitim vrstama tranzicija. U ovom slučaju izabrao sam „*Dissolve*“ tranzicije i stavio „Film dossolve“. Kada odaberemo efekt koji želimo koristiti jednostavno ga povučemo na željenu traku. Dakle pošto želimo efekt na početku i kraju clipa terbamo izabrati željeni efekt i svaki efekt povući iz mape na početak i kraj clipa. Istu radnju ponovimo za svaki clip. Kada je efekt stavljen na clip mišom možemo kontrolirati kada efekt počinje i završava. U nastavku je slika koja prikazuje poredak slika i efekata.



Slika 3.2. Izrada uvodne špice

Po završetku montaže špicu sam izvezao tako da za svaku buduću montažu lekcije imamo gotovu uvodnu špicu koju mogu učitati u projekt. Što se tiče montaža video lekcija princip je jednak za sve. U svaki projekt potrebno je učitati uvodnu špicu, snimljeni video materijal i audio zapise koje smo zasebno snimili i procesirali u Reaperu. Prvo je potrebno pregledati snimljeni video materijal i kratiti eventualne greške ili ponavljanja koja su se desila tokom snimanja. Materijal se krati na način da se stavi kursor na željeno vrijeme te se s Ctrl + K *cropa*

clip te ga izbrišemo tipkom Delete. Kada završimo *cropanje clipova* na početak trake stavimo uvodnu špicu i nakon nje obrađeni video materijal. Nakon toga potrebno je staviti obrađeni *audio* u video. To radim tako da u posebno *audio* traku stavim obrađeni zvuk. Nakon toga potrebno je sinkronizirati zvuk s videom. Na sreću postoji funkcija u Premieru koja to automatski rati tako da nije potrebno ručno poravnavati audio s *audiom* s kamere. Označio sam obje *audio* trake te kliknuo desni miš i odabrao naredbu „synchronize". Tada je potrebno samo u ponuđenim opcijama kliknuti da želimo sinkronizirati audio i kliknemo OK. Na poslijetku isključim traku u kojoj se nalazi svuk sa kamere i ostavim procesirani zvuk koji je sada sinkronizirani sa videom. Prije nego *export*am video na kraj sekvence stavim dissolve efekt kao što sam stavljao u uvodnoj špici. Za kraj *export*am video naredbom File > *Export* > Media. U prozoru imamo puno različitih parametara za *export* videa ali Premier ima već spremljene presetove za različite platforme. Za svoje lekcije izabrao sam preset „Youtube HD 1080" i klikuno „*Export*".

4. Web mjesto

Web mjesto sam zamislio da sadrži 3 kartice; početnu, video lekcije i kontakt. Na početnoj stranici objasnio bih čemu služi web mjesto i što ga čini jedinstvenim. Na kartici lekcije stavljao bih video materijali koji bi bili kategorizirani. Materijal sam odlučio podijeliti u 3 različite cijeline jer svaka cijelina ima različiti pristup. Prva cijelina je nazvata „koncepti". U ovoj cijelini objašnjavao sve koncepte koji su potrebni za napredno znanje impovizacije, glazbena teorija, analiza stilova, pristup melodijama, frziranju, modusima idt. Svaka lekcija bi imala zasebne primjere ali bi se bazirala na tome da se uz prikazani materijal objasni kojim principom korisnik može implementirati znanje za vlastiti glazbeni rast. Drugim riječima svrha videa je da korisnik nakon lekcije ne vježba samo date primjere već da sam može izgratiti svoj stil. Druga kategorija je nazvana „fraze". Ova kategorija je u neku ruku suprotni princip od prijašnje. Ova kategorija služi za širenje glazbenog vokabulara. U ovim lekcijama svaki video obrađuje jednu frazu koja se može implementirati za improvizaciju. Treća i zadnja kategorija je „razgovarajmo". U ovoj kategoriji pričam o principia vježbanja gitare. Ne dajem nikakve konkretne vježbe već pričam mogućim navikama koje učenici mogu steći, a da ih ne sputavaju u daljem napredovanju i kako se izvući iz tih situacija.

4.1. Izrada web mjesta

Za prvu verziju web mjesta koristio sam Wordpress CMS zboj jednostavnosti i mogućnosti brze izrade stranice. U nastavku ću razjasniti što su CSM sustavi i koje su njihove prednosti.

4.1.1. CMS

Content managment system je sustav koji omogučava klasifikaciju, organizaciju i povezivanje sadržaja. CMS služi za lakše upravljanje sadržajem. U današnje vrijeme najčešća se prikjenjuje za programska rješenja koja upravljaju velikim informacija. CMS sustavi se koriste pri sinkronizaciji podataka iz više izvora, za izvršavanje kolaborativnih projekata, za organizaciju rada u korporacijskim okruženjima i slično. Temeljna primjena CMS-a danas je u dinamičkom kreiranju web-stranica nove generacije. U korporacijskom kontekstu, CMS sustavi mogu se koristiti za integraciju različitih podatkovnih baza kao što su podaci o zaposlenicima, kupcima, proizvodima, radnim zadacima i sl. Koristeći CMS rješenja, svima njima se može upravljati s jednog mjesta. Jednako tako, putem zajedničkog sučelja zaposlenici mogu pristupati različitim dokumentima i podacima koje mogu koristiti za različite svrhe. [16]

Brzim razvojem i širem Interneta, web-stranice postale su najveći i najpristupačniji izvor informacija. Zbog toga se CMS sustavi danas uglavnom koriste za upravljanje informacijama na Internetu.

Komercijalni CMS sustavi:

ColdFusion, Kentico, Sitefinity, EasyEdit, Accrisoft, Jalios, Corvus, Webiny, SharePoint…

Besplatni CMS sustavi:

Open Source, Drupal, Joomia!, Wordpress, TYPO3, Magma, sNews, DotNetNuke, eZ Publish…

4.1.2. Wordpress

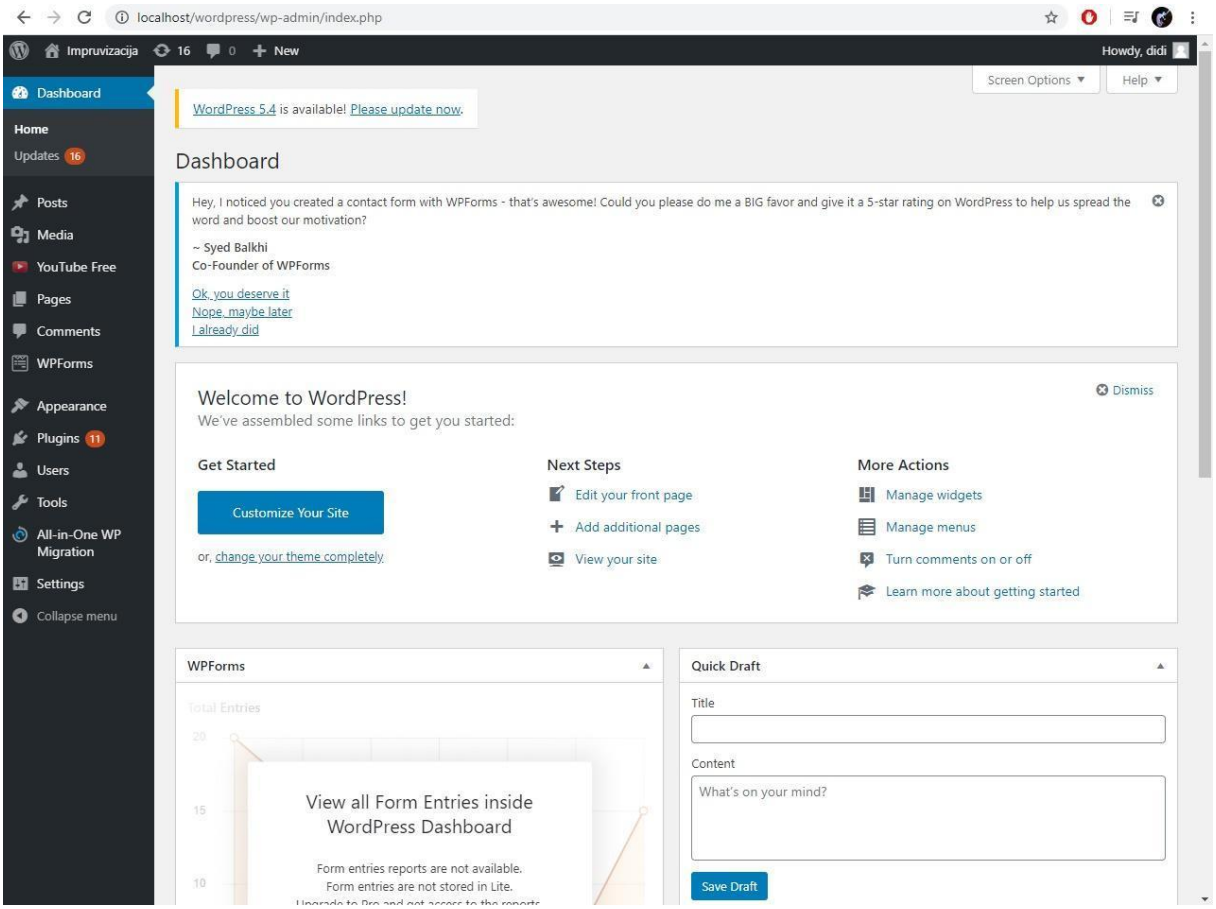
WordPress je trenutno najpopularniji open source *Content Management System*. Najčešće se korisiti za jednostavnu i brzu objavu *blogova*. Poduprt je PHP-om i MySQL-om te omogućava korišrenje plug-inova. [17]

Od 1 000 000 najvećih web stranica na svijetu njih 13% je izrađenu korištenjem wordpressa. Bio je objavljen 27. svibnja 2003. a verzija 2.9 je preuzeta preko 10.3 milijona puta. [18]

Wordperss je započet kao *blog,* ali se s vremenom razvio u potpuno razvijeni i najkorišteniji CMS sustav. [17]

4.1.3. Instalacija Wordpressa

Prvo je potrebno preuzeti instalaciju s stranice https://bitnami.com/stack/wordpress te nakon toga instalirati. Nakon što je instalacija završena log in vršimo putem linka: http://localhost/wordpress/wp-login.php . Ovdje unesemo ime korisničkog računa i lozinku koje smo definirali u instalaciji. Po prijavi otvara se korisničko sučelje admina stranice koje izgleda ovako:



Slika 4.1.3. WordPress sučelje

Sučelje se sastoji od kartica koje omogućuju prikaz izrađene stranice, dodavanje multimedijskih sadržaja, izrada novih podstranica i pisanje *blogova*. Također u padajućem izborniku su prikazani svi instalirani *pluginovi*.

4.1.4. Wordpress *pluginovi*

Pluginovi su aplikacije koje jednostavnim instaliranjem dodavaju nove mogućnosti i funkcionalnosti na wordpress stranici. Putem WordPress.org moguće je instalirati više od 48000 *pluginova* besplatno. Na stranicama poput GitHub-a moguće je naći i tisuće drugih

potpuno besplatnih *pluginova*. Također postoje i premium WordPress-*pluginovi* koje izrađuju nezavisni programeri i firme. Oni se naravno plaćaju.

WordPress je 2003. godine pokrenut kada je bio tek jednostavna aplikacija za brzu izradu *blogova,* no tijekom godina se razvijao te je danas jedan od najmoćnijih CMS sustava za izradu puno složenijih stranica uz pomoć korištenja *pluginova*.

Pluginovi su aplikacije koje se integriraju i pokreću povrh WordPress softvera. **Na primjer;**

- WooCommerce plugin omogućava izradu internetske trgovine u WordPressu,
- Job Manager plugin omogućava izradu stranice za zapošljavanje,
- Envira plugin omogućava izradu web stranice za fotografije,
- Knowledge base plugin omogućava izradu wiki web stranice, itd.
- U WordPress zajednici postoji popularna izreka „za to postoji dodatak“.

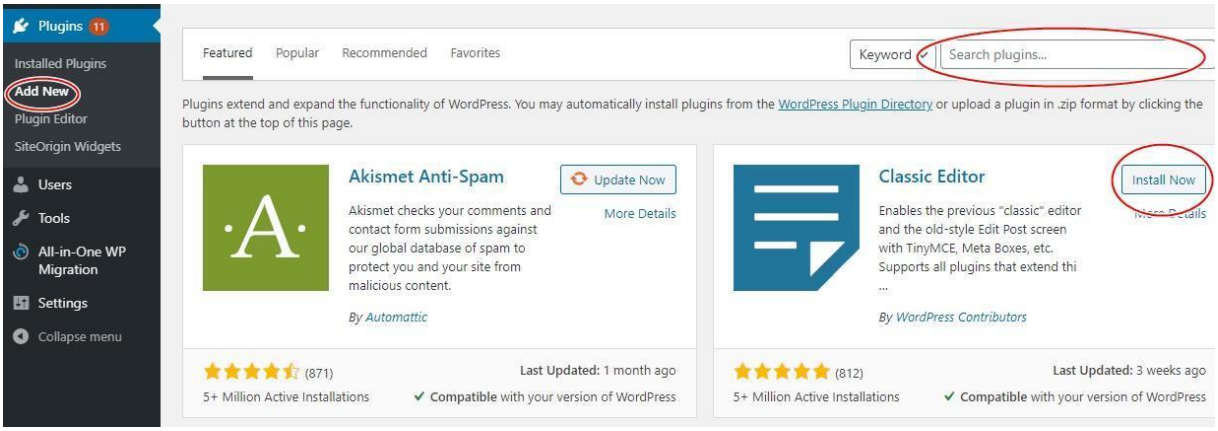
Što god ste zamislili uz pomoć WordPressa moguće je izraditi te ako značajka nije dostupna za nju sigurno postoji plugin. Postoje *pluginovi* za WordPress koji poboljšavaju SEO, performanse, dodaju kontakt obrasce, gumbe na društvenim mrežama, stvaraju galerije i tako još mnogo toga. *Pluginovi* mogu biti mali i nude samo jednu malu značajku (poput dodavanja slike na bočnu traku). Ili također mogu biti prilično veliki i također se ponašaju kao vlastita platforma. [19]

Način rada *pluginova*

WordPress je napisan na način da drugi programeri mogu dodati svoj kôd u njega. WordPress plugin API nudi robustan skup kuka i filtera koji omogućuju programerima da mijenjaju postojeću WordPress funkcionalnost ili dodaju novu funkcionalnost. Programerima je također omogućeno da pohranjivaju vlastite podatke u WordPress bazu podataka. *Pluginovi* mogu upotrebljavati WordPress vrste sadržaja, taksonomije i prilagođena polja koja korisnicima omogućuju pohranjivanje različitih vrsta sadržaja, a ne samo objave i stranice. Svaki WordPress plugin instaliran na vašoj web lokaciji registriran je u vašoj WordPress bazi podataka. Možete ih uključiti i deaktivirati u bilo koje vrijeme. [19]

4.1.5. Instalacija WordPress *pluginova*

Jednostavno kliknemo na „Plugins“ »“Add new“ i pretražite upisivanjem imena dodatka ili funkcionalnosti koju želite dodati.



Slika 4.1.5. Instalacija pluginova

WordPress će pretražiti *pluginove* te ispisati popis dostunih *pluginova* za instalaciju. Nakon što prođete popis *pluginova* i nađete ono što ste tražili kliknite „Install Now“. Međutim, trebate biti svijesni da WordPress nudi besplatne *pluginove* koji nisu nužni toliko korisni kao što su oni koji se plaćaju. Početnim korisnicima bi potražnja idealno plugin-a mogla biti zbunjujuća. Jedna od mnogih web stranica koja omogućava instalaciju WordPress *pluginova* je www.WPBeginner.com

4.1.6. Stranice

Sadržaj na wordpressu se može objavljivati u obliku posta ili stranice. U slučaju da se sadržaj objavljuje za *blog*, izabrat ćete opciju posta. Postovi u zadanim postavkama objavljuje sadržaj obrnutim kronološkim redoslijedom. Stranice za razliku od postova sadrže uvijek relevantne informacije kao npr. „Kontakt“, „Opis“, „Početna“… Iz tog razloga stranice se mogu organizirati kako bi se upravljalo sadržajem na web mjestu. Stranice i postovi su vrlo slični u smislu da oboje imaju naslov i sadržaj te estetski isto izgledaju kako bi se održao doslijedan izgled na cijelom web mjestu. Ipak, stranice imaju nekoliko ključnih razlika koje ih odjeljuju od postova:

Stranice su:

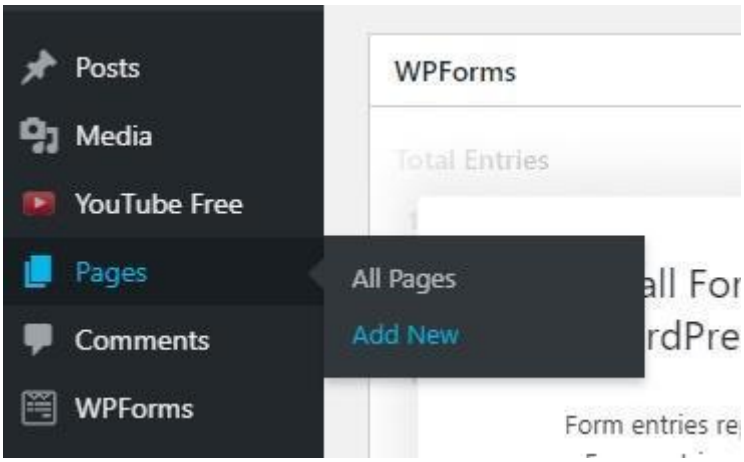
- Sadržaj koji nije vremenski ovisan
- Mogu se organizirati u stranice i podstranice
- Mogu koristiti različite predloške stranica, uključujući datoteke predložaka, oznake predloška i drugi PHP kod
- Sofisticiranije teme mogu pružiti širi raspon prilagodbi ili opcija prikaza za pojedine stranice
- Moguće je napraviti web stranicu pomoću WordPressa koja sadrži samo stranice

Stranice nisu:

- Pojavljuju se u vremenski strukturiranim prikazima unutar *blog*ovskog odijeljka
- Dopuštaju udruživanje taksonomije
- Uključene su u „feedove“ web mjesta [20]

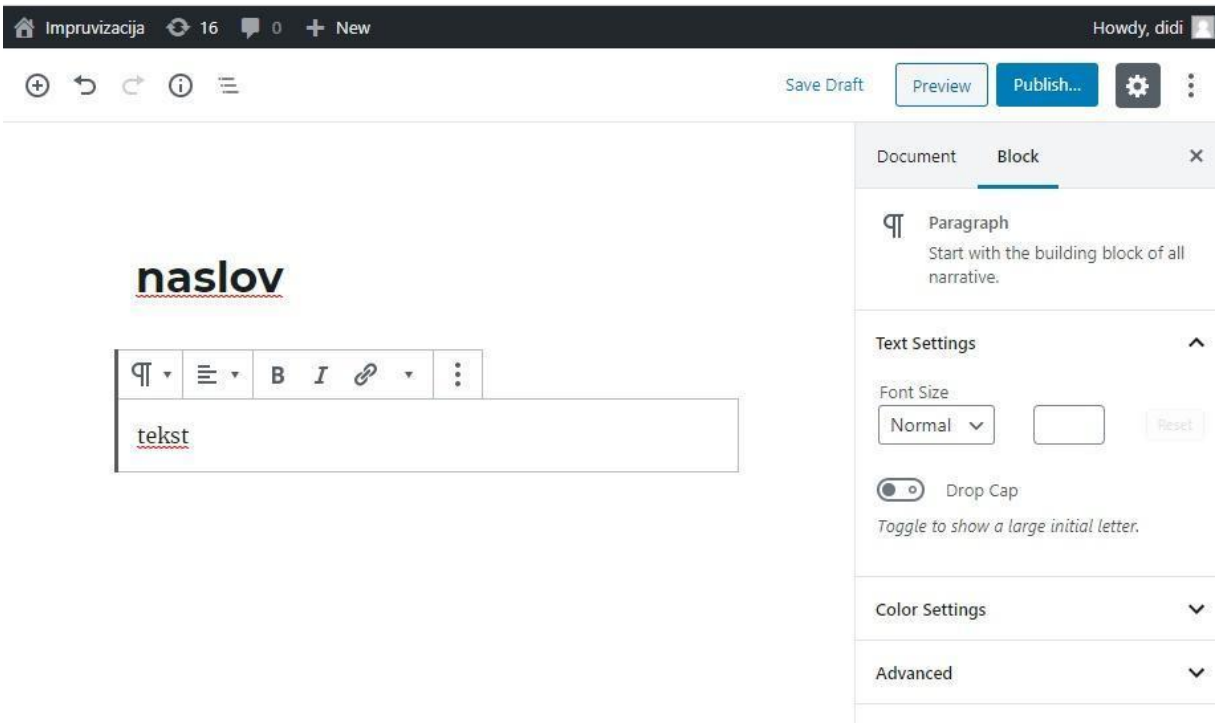
Kreiranje stranice

U padajućem izborniku kliknemo; „Pages“ □ “Add New“



Slika 4.1.6. Izrada stranice

Zatim nam se otvara prozor za izradu web stranica. U ovom prozoru možemo unijeti željeni naslov i tekst te uređivati sadržaj. Alati nam daju mogućnost poravnavanja i podebljavanja teksta te unos linka.



Slika 4.1.6. Uređivanje stranice

S desne strane nalaze se opcije za objavu stranice. Kada završimo uređivanje stranice možemo izabrati hoće li stranica biti vidljiva, kada će vremenski biti objavljena, pregled stranice prije objave i druge napredne postavke. Po završetku izrade u gornjem desnom kutu pliknemo na „Publish“ kako bismo objavili stranicu.

4.2. Izrada interaktivnog videa

Rješenje za izradu interaktivnih html videa našao sam na: http://thenewcode.com/977/Create-Interactive-HTML5-Video-with-WebVTT-*Chapter*s?fbclid=IwAR0Wfpz1CoqAQYHbd4zr1yGeY2C6xM3HHDsJc9fDssI_2Hp1J1TdxNhfLkg

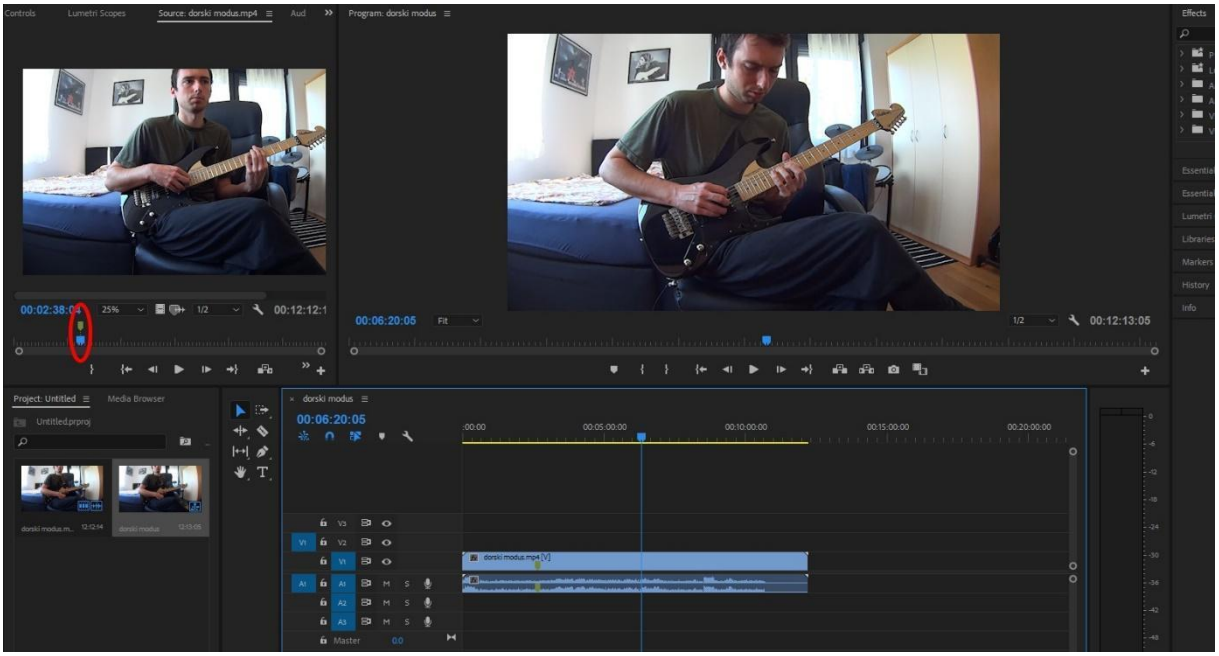
Metoda se bazira na izradi video datoteka koje sadrže markere ili *chaptere* koji dijele video u više cjelina. Uz video je potrebno napraviti VTT datoteku koja sadrži podatke o vremenu u kojeme počinju i završavaju *chapteri*. Funkcija tada učitava VTT datoteku te iz nje vadi informacije o video *chapteri*ma te na temelju tih informacija generira gumbe koji omogućavaju pregled videa u definiranim poglavljima.

4.2.1. Izrada video *chaptera*

Glavni sadržaj na web mjestu su video lekcije. Svaka lekcija sadrži više točaka koje se obrade u lekciji što naravno utječe na duljinu videa. Imajući to na umu, zamisao je bila stvoriti

video koji je podijeljen u poglavlja tako da korisnik može lakše navigirati kroz sadržaj. Ako u nekom trenutku korisnik ne razumije misao koja se nadovezuje na trenutnu lakše će naći traženu informaciju nego da ponovo gleda proteklih 5 minuta video lekcije.

Učitamo izrađeni video u projekt i povučemo datoteku na video traku, U gornjem lijevom prozoru pomoću kursora odaberemo vrijeme u kojemu želimo imati marker za video *chapter*. Kada smo odredili vrijeme jednostavno kilknemo tipku „M“ na tipkovnici.



Slika 4.2.1. Izrada video markera

Ovdje možemo vidjeti da se iznad kursora pojavila zelena oznaka za video marker. Kada se video *export*a na tom mjestu će se nalaziti novo poglavlje videa. Nakon što smo podijelili video u poglavlja potrebno je kreirati dokument iz kojega će funkcija moći izvući informacije o postojećim poglavljima.

4.2.2. WebVTT

WebVTT je format za prikaz vremenskih zapisa s vremenskim ograničenjem (poput titlova ili opisa) pomoću elementa <track>. On služi za dodavanje teksta u <video> tag. WebVTT se mora kodirati pomoću UTF-8. Struktura WebVTT-a zapisa se sastoji od:

- Izborna oznaka redoslijeda bajtova (BOM).
- Niz "WEBVTT".
- Izborno zaglavlje teksta desno od WEBVTT.
- Prazan redak, što je ekvivalent dva uzastopna nova retka.
- Nula ili više znakova ili komentara.

- Nula ili više praznih redaka. [21]

Primjer najmanje moguće WebVTT datoteke:

WEBVTT

Primjer jednostavne WebVTT datoteke s naslovom:

WEBVTT – Naslov

Primjer WebVTT datoteke s znakovima:

WEBVTT – datoteka s znakovima

1

00:01:34.521 --> 00:01:36.972

- tekst 1

- tekst 2

2

00:01:39.906 --> 00:01:42.146

- tekst 3

- tekst 4

Struktura WebVTT datoteke

Uzmimo sljedeći primjer da objasnimo strukturu WebVTT datoteke:

WEBVTT

00:01.000 --> 00.02.000

- ovo je tekst 1

00:06.000 --> 00.09.000

- ovo je tekst 2

U ovom slučaju:

- Prva linija počinje sa vremenom koje označava trenutak početka prikaza teksta
- U istoj liniji imamo *string* „-->“
- Liniju završamo drugim vremenom, koje označava kraj prikaza teksta
- Tada možemo imati jednu ili više linija teksta koje započinjemo s crticom (-)
- Svaka crtica (-) sadrži dio teksta koji će biti prikazan

U naše .vtt datoteke možemo unositi komentare koji će nam pomoći zapamtiti bitne informacije u našoj datoteci. Komentari trebaju biti zasebne linije koje započinju sa *string*om NOTE.

Bitno je ne koristiti prazne linije između vremena i teksta pošto će svaka prazna linija prekinuti vezu između vremena u teksta. [21]

WebVTT komentari

Komentari su neobavezna komponenta za unos informacija u WebVTT datoteku. Komentari nisu prikazani u video datoteci već se vide sam u kodu .vtt datoteke. Oni mogu imati više redova ali ne smiju biti prekinuti s praznim redom, u tom slučaju prazan red označava kraj komentara. Također komentari ne smiju sadržavati *string* „-->“ i znakove „&“ i „<“.

Komentar čine tri stvari:

- *String* NOTE
- Space ili nova linija
- Nula ili više znakova osim znabranjenih znakova koje smo naveli iznad [22]

Primjer WebVTT komentara:

NOTE Ovo je komentar

Primjer WebVTT komentara u više redaka:

NOTE
Ovaj komentar se sastoji

od dvije linije.

NOTE Komentar se može sastojati

od više linija.

WebVTT titlovi:

Najčešća upotreba WebVTT tehnologije je pri izradi titlova. Titlovi su tekst u filmu, najčešće prijevod dijaloga. Njih čini:

- Vrijeme početka
- *String* „-->“
- Vrijeme završetka
- Jedna linija teksta [23]

Primjer WebVTT titlova:

WEBVTT

00:15:01.001 --> 00:56:05.678

<v Ovo je prvi titl.

01:58:15.895 --> 02:05:19.947

<v Ovo je drugi titl.

02:06:01.125 --> 02:09:20.769

<v Ovo je treći titl.

02:25.34.786 --> 02:27.25.345

<v Ovo je četvrti titl.

WebVTT poglavlja:

Video poglavlja dijele video u dva ili više dijela. U slučaju da korisnik vidi prvi dio videa i kasnije želi nastaviti ili ponovo pogledati određenu sekciju, video poglavlja omogućavaju automatski skok u željeno područje.

Poglavlja čine:

- Redni broj sekvence povlavlja
- Vrijeme početka poglavlja
- *String* „-->“
- Vrijeme završetka poglavlja
- Opisni tekst poglavlja

Pravila pisanja vremenski oznaka poglavlja:

- Vrijeme završetka poglavlja mora biti veće od početka poglavlja
- Vrijeme se piše u obliku hh:mm.ss.ttt
- Hh su sati, moraju sadržavati namanje dvije brojike
- Mm si minute, moraju biti između 00 i 59
- Ss su sekunde, moraju biti između 00 i 59
- Ttt su milisekunde, moraju biti između 000 i 999 [21]

Primjer pisanja vremenskih oznaka:

00:15:01.001 --> 00:56:05.678

01:58:15.895 --> 02:05:19.947

02:06:01.125 --> 02:09:20.769

Primjer WebVTT poglavlja:

WEBVTT

NOTE

Ovo je datoteka korištena za učenje sviranja gitare.

1

00:00:00.000 --> 00:00:16.152

Uvod

2

00:00:19.272 --> 00:02:12.574

Vježba 1

3

00:02:15.621 --> 00:05:32.336

Vježba 2

.

4.2.3. HTML kod

Kada smo pripremili video i .vtt datoteku spremni smo testirati funkcionira li na webu. Ovdje je prikazani jednostavni kod koji učitava video zajedno s .vtt datotekom na web stranicu.

```
<figure>
     <video controls id="dorski modus">
               <source src="dorski modus.mp4">
          <source src="dorski modus.webm">
          <track kind="chapters" label="Locations"
src="chapters.vtt" srclang="en" default
onload="displayChapters(this)">
     </video>
     <figcaption>
          <ol id="chapters">
          </ol>
     </figcaption>
</figure>
```

Oprezniji pristup bio bi dodavanje ovih oznaka na stranicu putem JavaScript-a ako preglednik podržava HTML5 video i WebVTT. Budući da većina preglednika sada radi oboje, tako je napravljeno.

Naručeni popis u oznaci bit će ispunjen stavkama popisa koje sadrže opise poglavlja iz .vtt datoteke, s istaknutim opisom trenutne scene. Za to će nam trebati CSS, zajedno s izjavama za prikazivanje videozapisa i korisničkog sučelja: [24]

```
figure {
     font-size: 0;
     position: relative;
     background: #000;
}
figure video {
     width: 75%;
     height: auto;
     display: inline-block;
}
figure figcaption {
     position: absolute;
     right: 0; top: 0;
```

```
     background: #222;
     width: 25%;
     font-size: .8rem;
     color: #666;
     height: 100%;
     overflow-y: scroll;
}
figure figcaption ol {
     position: relative;
     list-style-type: none;
     margin: 0; padding: 0;
}
figure figcaption ol li {
     padding: .7rem 1rem;
     border-bottom: 1px dashed #000;
     transition: .3s;
}
.current {
     background: hsl(45,80%,50%);
     color: #000;
}
```

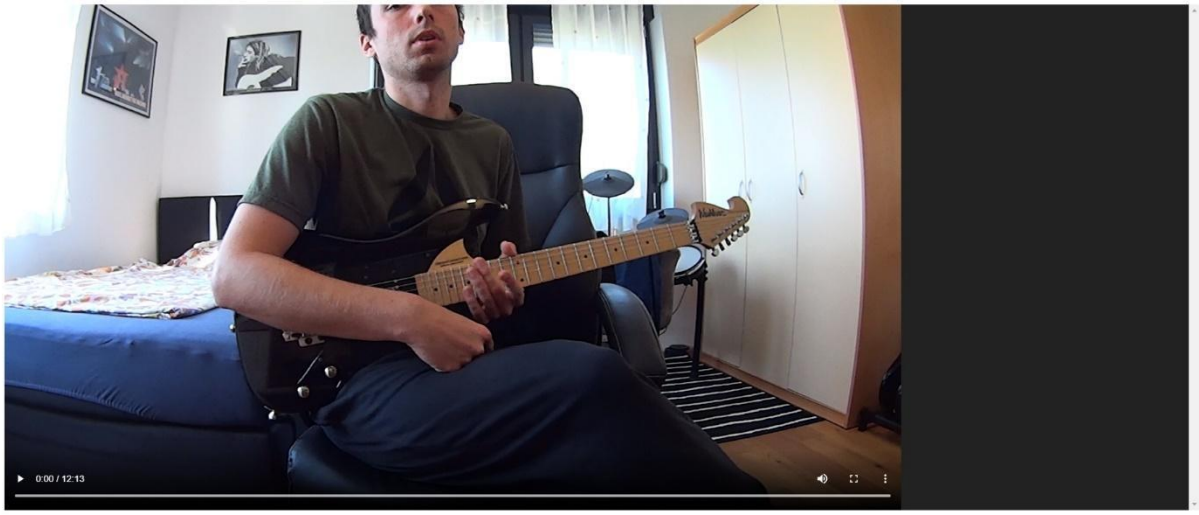
Na kraju očekujem da će preglednici razviti izvorno korisničko sučelje za bavljenje navigacijom poglavlja, ali niti jedan još nema ovu značajku. Trenutno će Chrome naznačiti da je videozapis s informacijama o poglavlju natpis, ali naslove prikazuje samo kao titlove. Dakle, naš prvi posao bit će sakriti ovaj zaslon, a zatim stvoriti korisničko sučelje za rukovanje poglavljima. To se postiže dodavanjem skripte na dnu stranice u jednoj funkciji koja se poziva kada se učita .vtt datoteka: [24]

```
function displayChapters(trackElement){
     if ((trackElement) && (textTrack = trackElement.track)){
          if (textTrack.kind === "chapters"){
               textTrack.mode = 'hidden';
               for (var i = 0; i < textTrack.cues.length; ++i)
{
                    cue = textTrack.cues[i],
                    chapterName = cue.text,
                    start = cue.startTime,
                    newLocale = document.createElement("li");
                    newLocale.setAttribute('id', start);
                    var localeDescription =
document.createTextNode(cue.text);
                    newLocale.appendChild(localeDescription);
                    locationList.insertBefore(newLocale);
                    newLocale.addEventListener("click",
                    function() {
                         video.currentTime = this.id;
```

```
                    },false);
                }
            textTrack.addEventListener("cuechange",
                function() {
                    var currentLocation =
this.activeCues[0].startTime;
                    if (chapter =
document.getElementById(currentLocation)) {
                        var locations =
[].slice.call(document.querySelectorAll("#chapters li"));
                        for (var i = 0; i < locations.length;
++i) {

      locations[i].classList.remove("current");
                        }
                        chapter.classList.add("current");
                        locationList.style.top = "-
"+chapter.parentNode.offsetTop+"px";
                    }
                },false);
            }
        }
}
```

4.2.4. Testiranje metode

Nakon testiranja koda u 4 različita preglednika (Chrome, Firefox, Microsoft Edge i Opera) došao sam do zaključka da kod iz rješenja nađenog na webu na žalost ne radi. Video je bio prikazan samo u Chrome-u ali bez funkcije premotavanja željenih poglavlja. U ostalim preglednicima niti video nije radio.



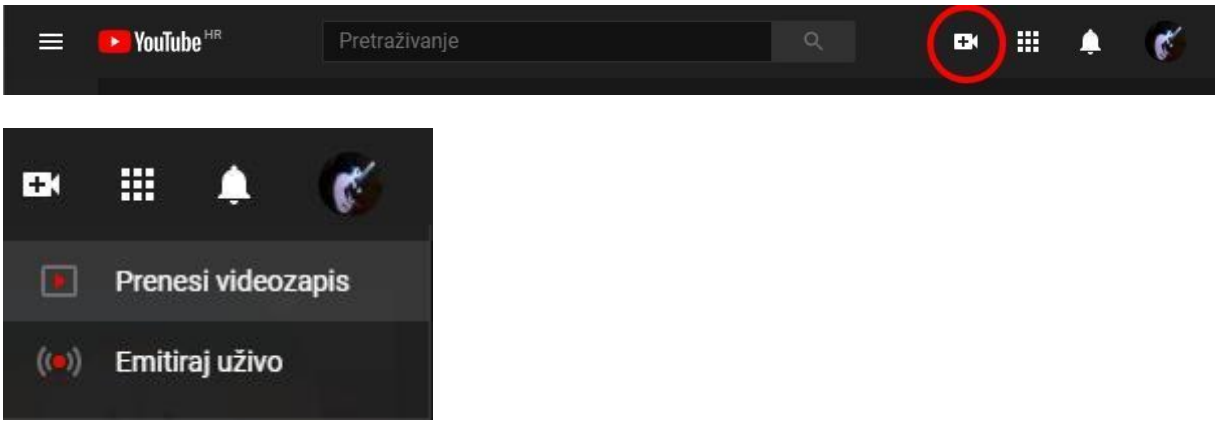
Slika 4.2.4. Testiranje prikaza video chaptera

Chapteri su trebali biti prikazani s desne strane videa, međutim prikaz nije radio. Iako rješenje nije radilo dobio sam ideju iskoristiti YouTube za prikaz *chaptera*. YouTube omogućava izradu video *chaptera* jednostavnim upisom vremena i imena *chaptera* u opisu videa. U nastavku ću prikazati kako se to radi.

4.2.5. Prijenos videa na YouTube

Prije svega prijavimo u postojeći Google račun i otvorimo YouTube stranicu na: https://www.youtube.com/

U gornjem desnom kutu kliknemo na kameru u kojoj je "+" i kliknemo "Prenesi videozapis".



Slika 4.2.5. Prijenos videa na YouTube

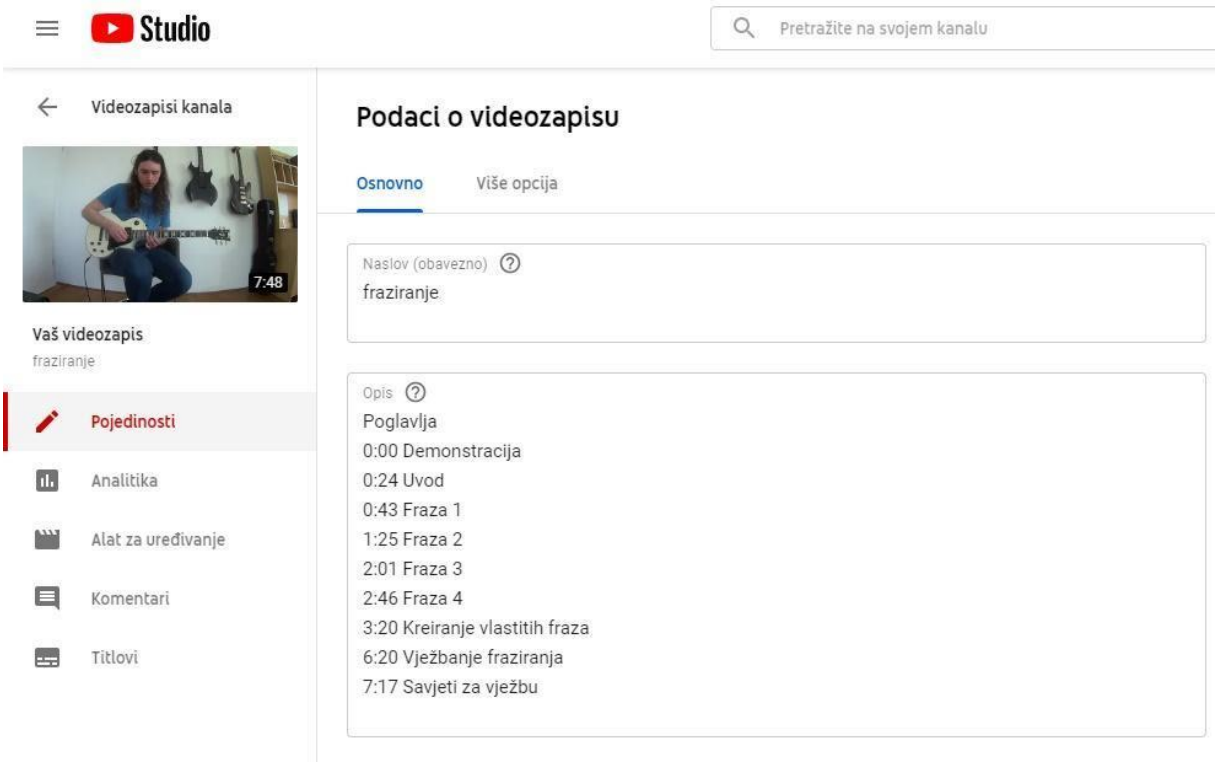
Za vrijeme prijenosa videa, odaberemo željene opcije kao što su vidljivost videa, mogućnost titlova, moguće zabrane itd. Kada se video prenese kliknemo "objavi" i u prozoru će se pojaviti link na kojem se nalazi video.

4.2.6. Izrada poglavlja u YouTube videu

Po prijenusu videa na početnoj stranici YouTube-a (https://www.youtube.com/) kliknemo na sliku računa i odaberemo "YouTube studio". U studiu kliknemo na karticu "videozapisi" s lijeve strane i odaberemo video koji želimo urediti. Poglavlja izradimo tako da u prostor za opis upišemo vrijeme početka *chapter* i njegov naziv u sljedećem obliku:

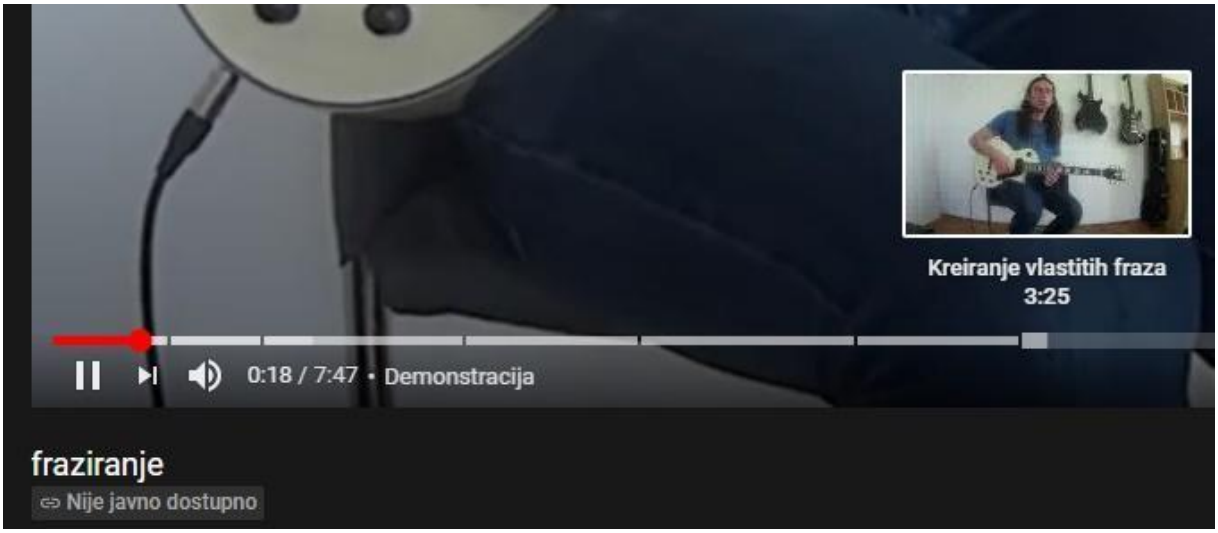
mm:ss naziv

Opis mora imati najmanje 3 poglavlja, svako poglavlje mora trajati minimalno 10 sekundi te prvo poglavlje počinje na 0: 00 u videu.



Slika 4.2.6. Izrada poglavlja

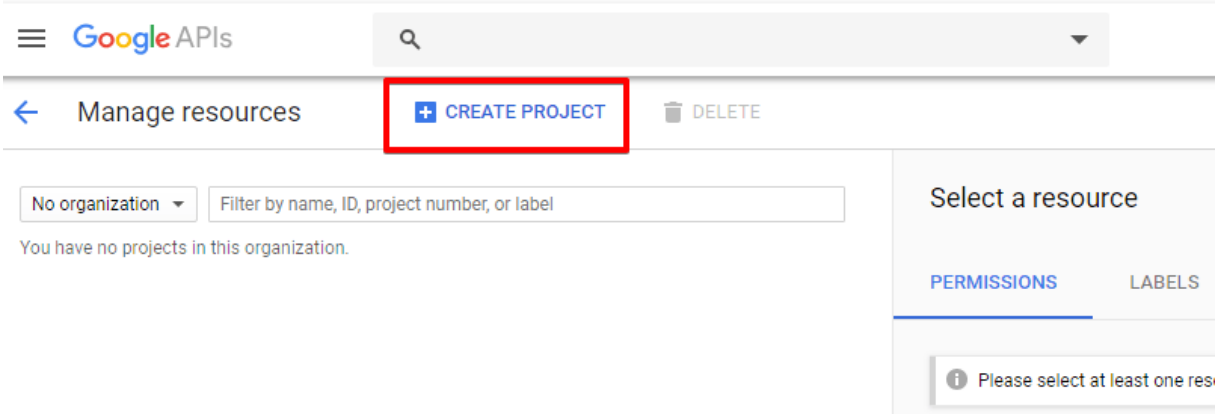
Sada kada stavimo miš na video crtom su odjeljena poglavlja tog video zapisa. Također, u opisu videa imamo vremenske linkove pomoću kojih možemo brzo navigirati kroz video.



Slika 4.2.6. Prikaz poglavlja

4.2.7. Spajanje s WordPress-om

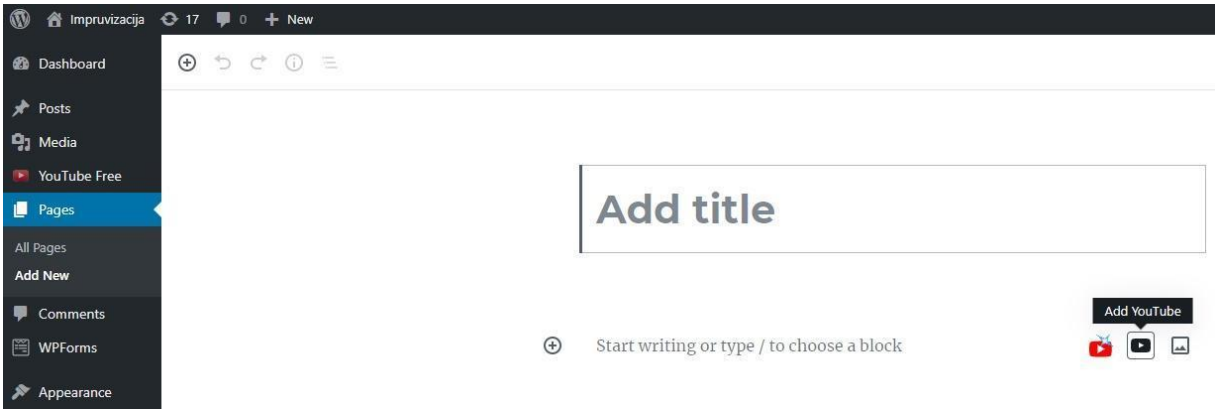
Otvorimo WordPress te instaliramo “YouTube Free” plugin. Instalacija je ista kao i za sve ostale *pluginove*. U instalaciji će biti potrebno unesti API *key*, a do njega dolazimo sljedeći način. Posjetimo https://console.cloud.google.com i prijavimo se u postojeći google račun.



Slika 4.2.7. Kreairanje API ključa

Nakon što smo se prijavili kliknemo na “*create project*”. U gornjem lijevom kutu kliknite vezu Google APIs, te zatim slijedite opciju veze nazvanu "YouTube DATA API" ispod ikone YouTube API-ja. Sada odaberite gumb "Omogući" koji se pojavljuje nakon što kliknete vezu YouTube DATA API. Kliknite plavi gumb krajnje desno uz frazu "Idi na vjerodajnice". Na prvoj opciji odabira koja se prikazuje na ovoj stranici odaberite YouTube Data API v3, a zatim web poslužitelj (kao što je node js. Tomcat) na drugoj opciji. Za treću opciju odaberite javne podatke. Na kraju, kliknite gumb s „koje vjerodajnice trebam“, a zatim prijeđite na sljedeći korak. Nakon što pritisnete gumb "Koje vjerodajnice trebam", Google će stvoriti vaš novi projekt i sada možete kopirati svoj API ključ s ove stranice. Sada možete zalijepiti API ključ koji ste kopirali..

Sada imamo mogućnost izraditi novu stranicu na kojoj je vidljivi YouTube video. Kada radimo novu stranicu s desne strane nam se pojavljuje YouTube ikona.



Slika 4.2.7. Povezivanje videa s WordPressom 1

Kliknemo na YouTube ikonu i u izbornik unesemo link videa koji želimo prikazati.

Po završetku naša stranica izgleda ovako:

Slika 4.2.7. Izgled stranice

4.2.8. Testiranje

Na žalost ovo rješenje ne funkcionira kako je očekivano jer Wordpress plugin za Youtube videe ne dozvoljava kreiranje vremenskih linkova za *chaptere* u videu. Također kada gledamo video ne vidimo vremensku lentu kada počinje i završava poglavlje. Ukoliko kliknemo na link videa i napustim stranicu kako bi video gledali na Youtube-u tada sve funkcije rade. Ovim rješenjem nisam zadovoljan pošto bih htio da korisnik ima sve funkcije direktno na web mjestu.

4.2.9. Izrada JS skripte

Nova ideja je bila napraviti JS skriptu koja bi komunicirala sa YouTube API servisom i dohvaćala informacije iz opisa, vadila iz njega vremenske oznake, tekst u opisu i u konačnici kreirala, na temelju tih informacija, gumbe kojima bi upravljali sa videom.

4.2.10. JavaScript

JavaScript je programski jezik koji zajedno s HTML i CSS-om čini osnovne tehnologije *World Wide Web-a*. JS omogućava izradu interaktivnih web stranica te je neizbježni dio web aplikacija. Velika većina web stranica koristi ga za ponašanje na strani klijenta, a svi glavni web preglednici imaju namjenski JavaScript mehanizam za njegovo izvršavanje. Kao jezik s više paradigmi, JavaScript podržava upravljane događajima, funkcionalne i imperativne stilove programiranja. Ima sučelja aplikacijskog programiranja (API-ji) za rad s tekstom, datumima, regularnim izrazima, standardnim strukturama podataka i objektnim modelom dokumenata (DOM). JavaScript motori izvorno su se koristili samo u web preglednicima, ali sada su ključne komponente ostalih runtime sustava, poput Node.js i Deno. Ti se sustavi koriste za izgradnju poslužitelja, a također su integrirani u okvire, kao što su Electron i Cordova, za stvaranje različitih aplikacija. Iako postoje sličnosti između JavaScript-a i Java-e, uključujući naziv jezika, sintaksu i odgovarajuće standardne knjižnice, ta su dva jezika različita i uvelike se razlikuju u dizajnu. [25]

4.2.11. iframe

ifrane je HTML tag koji određuje in line frame. On se može shvatiti kao prozor na web mjestu koji gleda na sadržaj sa neke druge web stranice. iframe tag sadrži source atribut preko kojeg određujemo lokaciju sadržaja kojeg želimo prikazati. [26]

Primjer iframe taga s w3schools:

```
<iframe src="https://www.w3schools.com" title="W3Schools Free
Online Web Tutorials"></iframe>
```
[27]

Česta primjena iframe-a je prikaz oglasa, YouTube videa, Google karta, social media i RSS feed-ova na web mjestu. Za većinu korisnika iframe je koristan pošto omogućava odvajanje od web mjesta i vanjskog izvora sadržaja koji se prikazuje. Korist ove tehnologije je što se podaci ne trebaju ažurirati. Kada se sadržaj ažurira na matičnoj stranici, automatski će prikazati ažurirani sadržaj na vašoj. [28]

4.2.12. API

Application programming interface ili sučelje za programiranje aplikacija je skup određenih pravila i specifikacija koje programeri slijede tako da se mogu služiti uslugama ili resursima operacijskog sustava ili nekog drugog složenog programa kao standardne biblioteke rutina (funkcija, procedura, metoda), struktura podataka, objekata i protokola. [29]

Laički rečeno API je poštar koji prima zahtjeve korisnika te sustavu govori što korisnik traži. Na poslijetku odgovor sustava proslijeđuje korisniku. U našem slučaju možemo koristiti YouTube API da dohvatimo informacije o rasporedu poglavlja u videu koji je već uploadani na YouTube. Time možemo iskoristiti već uploadane videe koji imaju složene *chaptere* i prikazati ih na web mjestu uz pomoć iframea.

4.2.13. Funkcija

initializeAPIScript()

Ova funkcija kreira script tag na stranici na kojoj god je implementirana.

```
var tag = document.createElement('script');
tag.id = 'iframe-demo';
tag.src = 'https://www.youtube.com/iframe_api';
var firstScriptTag =
document.getElementsByTagName('script')[0];
firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);
```

Znači ovdje dodajemo tag.id s imenom **iframe-demo,** src tag i na kraju kreiramo sami **<script>** tag pri dnu HTML dokumenta.

extractYouTubeVideoID()

Ova funkcija je bavi vađenjem ID elemente iz **iframe-a** koji je postavljen na web.

Prije svega, provjerimo postoji li uopće **iframe** na stranici na kojoj se nalazimo.

```
if(youtubeVideo) {
```

Prvo vadimo cijeli **src** tag:

```
youtubeVideoSource = youtubeVideo.getAttribute('src');
```

A onda između 30-og i 41-og znaka, vadimo ID.

```
youtubeVideoID = youtubeVideoSource.substring(30, 41);
```

U slučaju da **iframe** ne postoji, ispišemo poruku u **else** dijelu.

onYouTubeIframeAPIReady()

U ovoj funkciji kreiramo YouTube player i pod videoID mu predajemo ID koji smo u prethodnoj funkciji izvadili van.

Naknadno, dodajemo mu evente koji su samo funkcije s kojima provjeravamo je li taj player spreman i ispravno postavljen i je li se promijenilo stanje. OnStateChange će registrirati da smo kliknuli na gumb i promijenili sekciju.

```
    player = new YT.Player('video', {
        videoId: youtubeVideoID,
        events: {
            'onReady': onPlayerReady,
            'onStateChange': onPlayerStateChange
        }
    });
```

getYouTubeVideoDescription()

Ova funkcija je zadužena za komunikaciju sa YouTube serverom. Postavljamo po definiranim parametrima i ubacujemo API key u sam url.

```
let url =
"https://www.googleapis.com/youtube/v3/videos?part=snippet&id=
" + youtubeVideoID + "&key=AIzaSyBDE4OwtZzeVDSxo-
jL5hJ7FYXwHRzliZM";
```

Nakon toga radimo poziv sa **fetch** metodom.

Odgovor koji dobijemo, spremimo u generički objekt pod imenom **data**.

U slučaju uspješne konekcije, u **then** dijelu šaljemo podatke u zadnje dvije funkcije.

```
    fetch(url)
    .then(response => response.json())
    .then(data => {
        findTimestamps(data.items[0].snippet.description),

createTimestampList(data.items[0].snippet.description.split("\
n"))
    });
```

findTimestamps()

Ova funkcija nam je namijenjena na pronalasku vremenskih oznaka po regularnoj ekspresiji koju smo definirali pri vrhu skripte.

```
var regex = /(?<=\()([^)]+)(?=\))/g;
```

Taj regex će pronaći brojeve od 0-9 odvojene : znakom. Znači 00:00 ili 0:00.

Ukoliko ih nađem, spremiti će ih u **timestamps** polje.

```
if(match) {
      timestamps.push(match[0]);
}
```

Na kraju te metode pozivamo metodu za pretvaranje dobivenih vremena u sekunde i predajemo joj polja vremenskih oznaka kao parametar. Pošto YouTube radi samo sa sekundama pri mijenjanju sekcija.

```
convertToSeconds(timestamps);
```

createTimestampList()

Ovo je u biti dosta jednostavna funkcija, od descriptiona koji smo predali u getYouTubeVideoDescription funkciji kreiramo HTML elemente, točnije paragrafe. Sa for petljom prolazimo kroz cijeli objekt i od svake linije radimo novi paragraf.

```
for(let i = 0; i < data.length; i++) {
        var paragraph = document.createElement("P");
        paragraph.setAttribute("class", "timestamps");
        let text = document.createTextNode(data[i]);
        paragraph.appendChild(text);

        description.appendChild(paragraph);
    }
```

convertToSeconds()

Zadnja funkcija radi točno ono što joj ime govori. Uzima sve vremenske oznake koje smo spremili, miče dvotočku, lijevu stranu od dvotočke sprema kao minute i množi sa 60 i desnu stranu od dvotočke sprema kao sekunde.

Nakon toga svega, od svake vremenske oznake u polju timestamps, radimo HTML element, točnije <a> element koji će se ponašati kao gumbi kojima ćemo mijenjati *chaptere*.

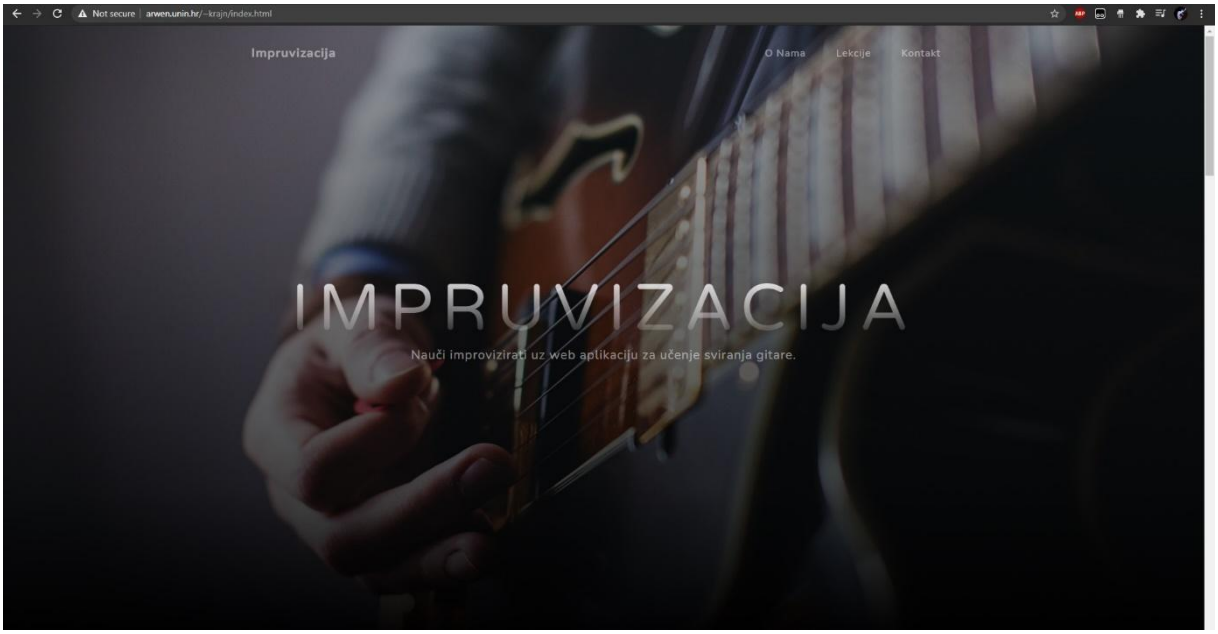
```
var realTimestamps = [];
    for(let i = 0; i < timestampsData.length; i++) {
        let splitTime = timestampsData[i].split(':').join('');
        let seconds = splitTime.slice(1, 3);
        let minutes = splitTime.slice(0, 1) * 60;
        let time = +minutes + +seconds;
        realTimestamps.push(+minutes + +seconds);

        var x = document.createElement("A");
        var t = document.createTextNode(timestampsData[i]);
        x.setAttribute("onClick", "player.seekTo(" + time +
")");
        x.setAttribute("class", "timestamp-btn");
        x.appendChild(t);

        timestampContainer.appendChild(x);
    }
```

4.2.14. Testiranje

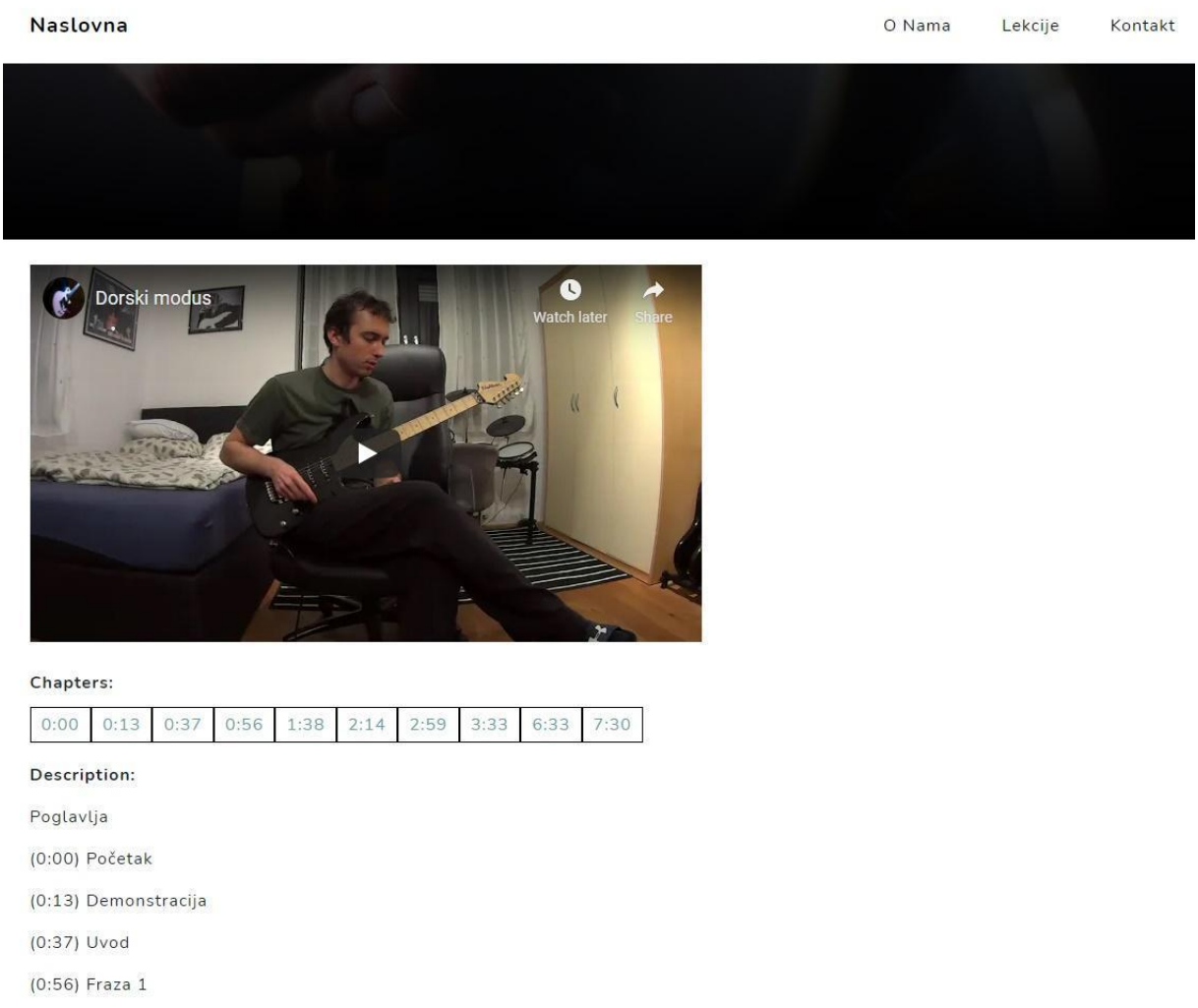
Funkciju sam testirao bez implementacije za wordpress i radila je bez poteškoća. Međutim, kada je funkcija bila ubačena u wordpress nije radila. Razlog tome je da se wordpress mješao u interakciju funkcije te se podaci nisu mogli dohvatiti. U konačnici sam odlučio da je jednostavnije načiniti stranicu da ručno pišem vlastiti HTML. Wordpress je dobro rješenje za brzu izradu web mjesta čak i ako korisnik ne razumije web programiranje ali na žalost kada je potrebna prilagodba wordpressa za izradu stvari za koje već ne postoje *pluginovi* tada nastaju problemi i komplikacije. Na sreću načiniti novo web mjesto nije predstavljalo problem pošto na stranici nije bilo previše sadržaja. Koncepcija web mjesta je ista ostala kao što je bila i na wordpressu. U konačnici potrebno je bilo estetski doraditi kod u kojemu sam testirao skriptu tako da je najteži dio posla bio odrađen unatoč tome što sam WordPress stranicu trebao odbaciti.

4.3. Izgled web mjesta

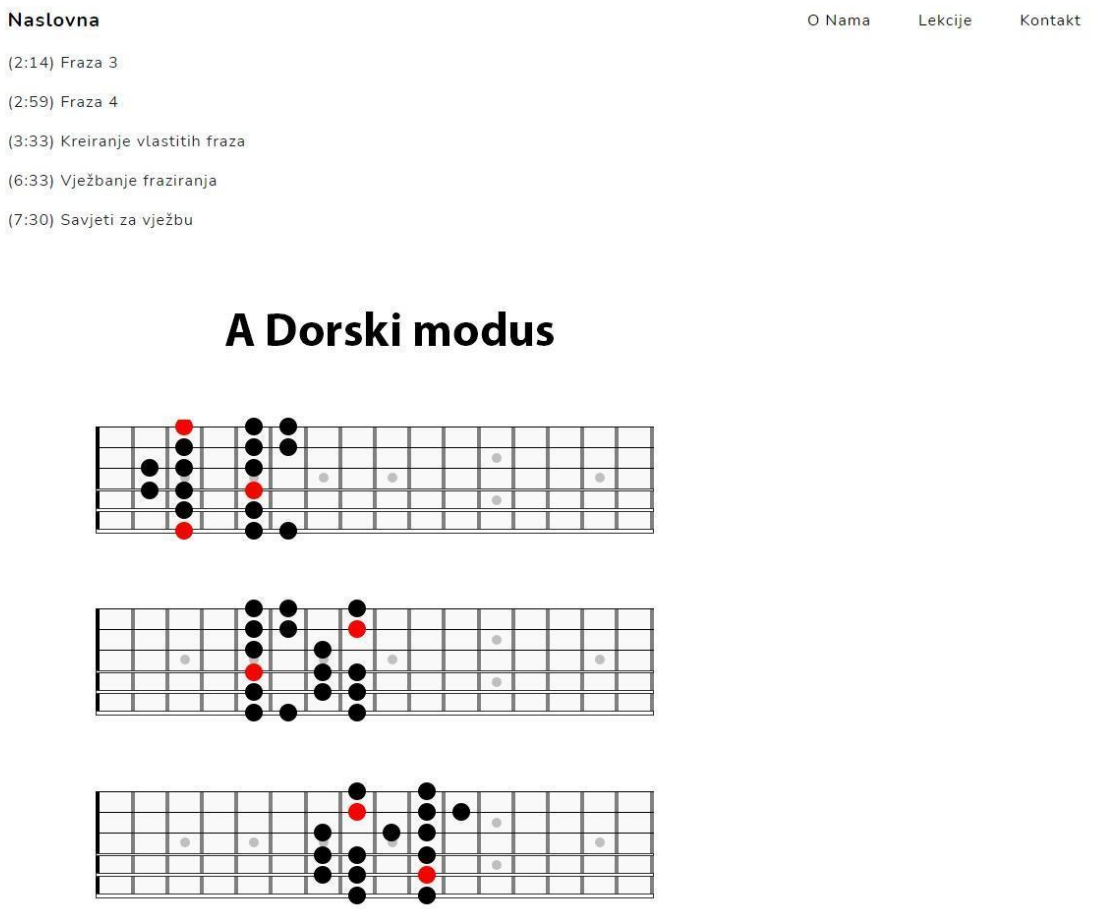
Svaka stranica na web mjestu ima na vrhu sliku na kojoj piše na kojoj podstranici se korisnik nalazi. Slika je na svim podstranicma jednaka kako bi se ostvario konstantni dizajn. Svaka stranica je podijeljena u dva dijela. Prvi dio čini na samom vrhu glavni izbornik te ispod njega pozadinska slika i naslov podstranice. Drugi dio čini sadržaj stranice. Na primjer ako se nalaziom na kontakt stranici spod slike se nalazi kontakt forma, ako smo na odabratoj lekciji ispod slike se nalazi video lekcija, gumbovi za poglavlja u videu i dodatni materijali kao što su TAB notacije.



Slika 4.3. Gordnji dio stranice



Slika 4.3. Video s gumbima za skakanje po poglavljima



Slika 4.3. Dodatni sadržaj za učenje

5. Zaključak

Cilj ovog web mjesta bio je izraditi interaktivni sadržaj za lakše učenje sviranja gitare. Za izradu interaktivnih videa koristio sam tri metode do kad nisam uspio postići željeni cilj.

Prva metoda koja je uključivala WebVTT tehnologiju iz nekog razloga nije radila. Možda je problem u tome da se kod treba optimizirati. Bez obzira u čemu je problem svejedno smatram da to rješenje ne bi bilo praktično pošto bi za svaki video trebalo pisati zasebnu VTT datoteku te ukoliko su potrebne promjene, nije moguće promjene napraviti unutar same VTT datoteke već je potrebno i u videu sve markere promjeniti te ponovo *exportati* video.

Druga metoda u kojoj je Wordpress komunicirao s YouTube API-om je omogućavala prikaz videa s YouTube-a. Međutim, iako su *chapteri* bili vidljivi na YouTube-u, WordPress nije mogao prkiazati *chaptere* na progress bar-u. Pošto je video radio dobio sam ideju za treće rješenje.

Treća metoda bila je zamišljena da uz pomoć YouTube API-a skripta dohvaća informacije o video *chapteri*ma te da na temelju tih informacija generira gumbove za iste. Ovu metodu sam zasebno testirao u HTML kodu bez impelementacije u WordPress te je sve funkcioniralo. Kada sam funkciju implementirao u WordPress isti problemi su se javljali kao i u prijašnjoj metodi. Video je bio prikazan ali *chapteri* nisu. WordPress je dobar alat za izradu stranica gdje nije potrebna njegova prilagodba pošto većinu posla radi „sam“. Međutim kada se rade skripte na vlastitu ruku i pokušavaju ubaciti u WordPress, on će se „umješati“ te će nastajati komplikacije. Iz tog razloga sam odlučio „ručno“ izraditi cijelu stranicu. U konačnici smatram da je treća metoda najbolje rješenje jer ne zahtijeva izradu *chaptera* u videodatoteci već je potrebno samo unijeti podatke o željenom vremenu direktno na YouTube-u. Nadalje update videa je lagani te svaki update je automatski vidljiv te je potrebno samo na jednom mjestu vršiti promjenu, a ona će biti vidljiva na oba mjesta.

U Varaždinu, _____

6. Literatura

[1] E Findlay: Rhythm and movement: Applications of Dalcroze eurhytmics, Summy-Brichard Music, Van Nuys, 1995.

[2] M Houlahan, P Tacka: Kodály today: A cognitive approach to elementary music education, Oxford Unversity Press, New York, 2015.

[3] Orff-Schulwerk: An Integrated Foundation: Music Educators Journal, veljača 1986, str. 41

[4] SJ Rea: The Suzuki flute method: A history and description, The Florida State University, ProQuest Dissertations Publishing, Ann Arbor, 1999.

[5] E Gordon: Learning sequences in music: A contemporary music learning theory, GIA Publications, Chicago 2007

[6] Brent M. Gault, Carlos Abril: Teaching general music: Approaches, issues, and viewpoints, Oxford University Press, 2016.

[7] JM Felerabend: Conversational solfege, IL: GIA, Chicago, 2001.

[8] M Carabo-Cone: A Sensory-Motor Approach to Music Learning, MCA Music, New York, 1968.

[9] RB Thomas: Manhattanville Music Curriculum Program, Manhattanville Coll, New York, 1970.

[10] D Lebier: Popular Music Pedagogy: Peer Learning in Practice, Music Education Research 10 br. 2, 2008, str. 93-213.

[11] http://www.tonalsoft.com/enc/g/german-h.aspx, dostupno 2005.

[12] RJ Burgess: The Art of Music Production: The Theory and Practice, Oxford University Press, New York, 2013.

[13] https://guitargearfinder.com/guides/best-guitar-plugins, dostupno 5.5.2021.

[14] J Pakarinen, DT Yeh, A review of digital techniques for modeling vacuum-tube guitar amplifiers, Computer Music Journal, MIT Press, 2009.

[15] http://reapertv.co.uk/how-to-build-a-song-with-ez-drummer-2/, dostupno 16.9.2016.

[16] B Boiko: Content Management Bible, Wiley Publishing, Indianapolis, 2005.

[17] A Brazell: WordPress Bible, Wiley Publishing, Indianapolis, 2010.

[18] https://wordpress.org/news/2010/06/thelonious/ dostupno 17.6.2010.

[19] https://www.wpbeginner.com/beginners-guide/what-are-wordpress-plugins-how-do-they-work/ dostupno 30.1.2017.

[20] https://wordpress.org/support/article/pages/ dostupno 12.5.2018.

[21] https://developer.mozilla.org/en-US/docs/Web/API/WebVTT_API dostupno 25.2.2021.

[22] T Steainer, PA Champin, H Műhleisen, B Encelle, R Verborgh, Y Prië: Weaving the Web(VTT) of Dana, Proceedings of the 7th Workshop on Linked Data on the Web 2014, str. 1-10.

[23] S Powers: HTML5 Media: Integrating Audio and Video with the Web, O'Reilly Media, Sebastopol, 2011.

[24] http://thenewcode.com/977/Create-Interactive-HTML5-Video-with-WebVTT-*Chapter*s?fbclid=IwAR0Wfpz1CoqAQYHbd4zr1yGeY2C6xM3HHDsJc9fDssI_2Hp1J1TdxNhfLkg dostupno 7.3.2015.

[25] D Crockford: JavaScript: The Good Parts, O'Reilly Media, Sebastopol, 2008.

[26] https://www.hostinger.com/tutorials/what-is-iframe/ dostupno 9.3.2021.

[27] https://www.w3schools.com/tags/tag_iframe.ASP dostupno 5.4.2021.

[28] https://*blog*.logrocket.com/the-ultimate-guide-to-iframes/ dostupno 23.1.2020.

[29] https://www.hubspire.com/resources/general/application-programming-interface/ dostupno 11.5.2021.

7. Popis slika

Slika 3.1. Službena stranica za preuzimanje DAW-a 7
Slika 3.1.1. Digitalna simulacija pojačala 9
Slika 3.1.1. Impulse response loader 10
Slika 3.1.2. EZdrummer *software* 11
Slika 3.1.2. Prikaz signala koji clippa 12
Slika 3.2. Izrada uvodne špice 13
Slika 4.1.3. WordPress sučelje 16
Slika 4.1.5. Instalacija *pluginova* 18
Slika 4.1.6. Izrada stranice 19
Slika 4.1.6. Uređivanje stranice 20
Slika 4.2.1. Izrada videomarkera 21
Slika 4.2.4. Testiranje prikaza video *chaptera* 24
Slika 4.2.5. Prijenos videa na YouTube 25
Slika 4.2.6. Izrada poglavlja 26
Slika 4.2.6. Prikaz poglavlja 26
Slika 4.2.7. Kreairanje API ključa 27
Slika 4.2.7. Povezivanje videa s WordPressom 1 28
Slika 4.2.7. Izgled stranice 28
Slika 4.3. Gordnji dio stranice 34
Slika 4.3. Video s gumbima za skakanje po poglavljima 35
Slika 4.3. Dodatni sadržaj za učenje 36

Sveučilište Sjever

SVEUČILIŠTE SJEVER

IZJAVA O AUTORSTVU I SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, KARLO RAJN *(ime i prezime)* pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom IZRADA INTERAKTIVNOG WEB MJESTA ZA UČENJE SVIRANJA GITARE *(upisati naslov)* te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

Karlo Rajn
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, KARLO RAJN *(ime i prezime)* neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom IZRADA INTERAKTIVNOG WEB MJESTA ZA UČENJE SVIRANJA GITARE *(upisati naslov)* čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)

KARLO RAJN Karlo Rajn
(vlastoručni potpis)