

# Interaktivna knjiga primjenom web tehnologija

---

**Klarić, Stjepan**

**Undergraduate thesis / Završni rad**

**2022**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:962108>

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

*Download date / Datum preuzimanja:* **2024-11-26**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**INTERAKTIVNA KNJIGA PRIMJENOM WEB  
TEHNOLOGIJA**

**Završni rad**

**Stjepan Klarić**

**Osijek, 2022.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac ZIP - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 14.09.2022.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Stjepan Klarić
<b>Studij, smjer:</b>	Preddiplomski sveučilišni studij Računarstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	R4221, 25.07.2018.
<b>OIB Pristupnika:</b>	12870049633
<b>Mentor:</b>	Izv. prof. dr. sc. Irena Galić
<b>Sumentor:</b>	Dr. sc. Hrvoje Leventić
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Interaktivna knjiga primjenom web tehnologija
<b>Znanstvena grana rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rad:</b>	Zadatak završnog rada je istražiti i opisati pristupe izradi interaktivnih knjiga. Fokusirati se na postojeće alate i opisati njihovu primjenu, prednosti i nedostatke. Za praktični dio rada potrebno je izraditi web aplikaciju koja će omogućiti korisnicima izradu interaktivne knjige. Tehnologija: web razvojni okvir po izboru Tema rezervirana za: Stjepan Klarić Sumentor s FERIT-a: Hrvoje Leventić
<b>Prijedlog ocjene završnog rada:</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	14.09.2022.
<b>Datum potvrde ocjene od strane Odbora:</b>	21.09.2022.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O ORIGINALNOSTI RADA**

Osijek, 22.09.2022.

<b>Ime i prezime studenta:</b>	Stjepan Klarić
<b>Studij:</b>	Preddiplomski sveučilišni studij Računarstvo
<b>Mat. br. studenta, godina upisa:</b>	R4221, 25.07.2018.
<b>Turnitin podudaranje [%]:</b>	4

Ovom izjavom izjavljujem da je rad pod nazivom: **Interaktivna knjiga primjenom web tehnologija**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Irena Galić

i sumentora Dr. sc. Hrvoje Leventić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.  
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

<b>1. UVOD.....</b>	<b>1</b>
1.1. Zadatak završnog rada .....	1
<b>2. ALATI ZA IZRADU INTERAKTIVNIH KNJIGA .....</b>	<b>2</b>
2.1. Sribbble .....	2
2.2. Kotobee.....	3
2.3. Canva .....	4
2.4. Adobe InDesign.....	5
2.5. Microsoft Word i Google Docs .....	6
2.6. Razlozi za razvijanje samostalnog rješenja.....	6
<b>3. RAZVOJNI OKVIRI .....</b>	<b>8</b>
3.1. Korištene tehnologije .....	8
3.2 PHP .....	9
3.3. MySQL.....	10
3.4. MVC .....	11
<b>4. REALIZACIJA INTERAKTIVNE KNJIGE.....</b>	<b>12</b>
4.1. Kreiranje projekta .....	13
4.2. Laravel Mix i Tailwind .....	13
4.3. Registracija .....	13
4.4. Objave .....	16
4.5. Povratne informacije .....	18
4.6. Urednik za tekst .....	19
4.7. Pregled korisnika .....	20
4.8. Sadržaj i sortiranja .....	21
4.9. Najbolje priče i autori.....	22
4.10. Krajnji rezultat .....	24

<b>5. ZAKLJUČAK.....</b>	<b>25</b>
<b>LITERATURA .....</b>	<b>26</b>
<b>ŽIVOTOPIS .....</b>	<b>27</b>
<b>SAŽETAK .....</b>	<b>28</b>
<b>ABSTRACT.....</b>	<b>29</b>
<b>PRILOZI.....</b>	<b>30</b>

# 1. UVOD

Poznavanje osnova web tehnologija u današnjici otvara vrata mnogim novim mogućnostima jer je internet postao broj jedan izvor za informacije, te nudi mogućnosti moderne izvedbe nekih starih koncepta koji prije nisu bili moguće ostvarivi. Pojam web tehnologija odnosi se na razne alate i tehnike koji se koriste u procesu komunikacije između različitih tipova uređaja preko interneta. Primjenom tih alata i tehnika, moguće je na lakši način ostvariti ideje, te s obzirom da je internet danas pristupačan skoro svakom čovjeku, sama izvedba će doći do više ljudi.

U ovom radu proučeni su razni postojeći alati za izradu interaktivnih knjiga, ukazane su njihove prednosti, mane, način korištenja, te neka rješenja koja bi ovaj rad nudio u odnosu na postojeća rješenja. Dalje, opisan je alat pomoću kojeg je ostvaren sam praktični dio rada, gdje su ukazane njegove prednosti. Cilj ovog rada je privući veći broj mlađe populacije ka pisanju književnih djela, gdje je uz postojeće prilagođene koncepte ostvaren zanimljiv način pisanja djela.

## 1.1. Zadatak završnog rada

Zadatak završnog rada je istražiti i opisati pristupe izradi interaktivnih knjiga. Fokus je na postojećim alatima uz njihov opis i primjenu, prednosti i nedostatke. Za praktični dio rada potrebno je izraditi web aplikaciju koja će omogućiti korisnicima izradu interaktivne knjige.

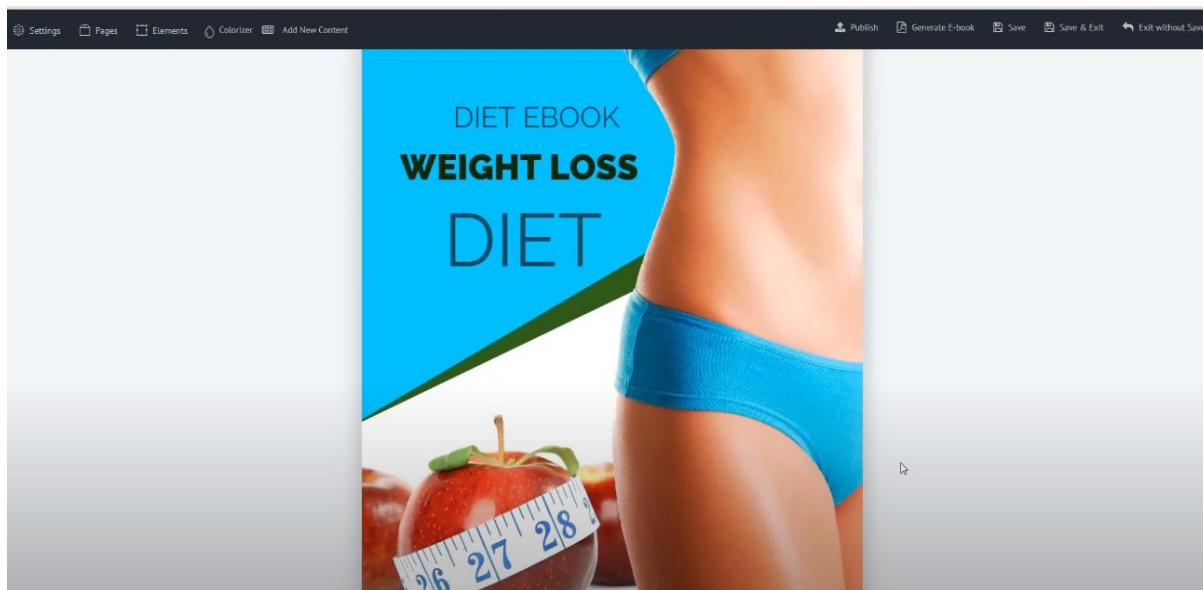
## 2. ALATI ZA IZRADU INTERAKTIVNIH KNJIGA

Interaktivna knjiga je prenosiva i djeljiva datoteka, koja sadrži značajke kao što su slike, zvuk, interaktivna pitanja, mogućnost uređivanja teksta, mogućnost dodavanja poglavlja, sortiranje sadržaja i slično. Sam pojam interaktivna knjiga nije poznat većini ljudi, no kada se bolje prouče mogućnosti koje nudi interaktivna knjiga, te koncepti koji se mogu ostvariti pomoću nje, shvati se da je to snažan alat koji može unaprijediti stari način pisanja knjiga. U ovom poglavlju proučit će se razni alati koji već postoje za izradu interaktivnih knjiga, alternativni alati pomoću kojih se interaktivna knjiga može ostvariti, te naposljetku dati osvrt i ideje za implementaciju nekih već postojećih značajki za sam praktični dio rada.

### 2.1. Scribble

Scribble je web alat za izradu interaktivnih knjiga. Pri kreiranju novog projekta dobije se mogućnost biranja velikog broja predefiniраних dizajnova, te nudi se opcija pretpregleda svakog dizajna. Nakon odabira dizajna dobije se gotov predložak kojeg je moguće urediti na željen način. (Slika 2.1) Ovisno o kvaliteti odabranog predložka, on može sadržavati različite značajke koje se mogu ukloniti ako nisu potrebne. Sadrži mogućnosti kao što je dodavanje elemenata, mijenjanje boja određenih elemenata, te dodavanje samostalnih elemenata. Elementi se dodaju na povuci i ispusti (engl. *drag and drop*) principu. Također, sadrži značajku koja nudi mogućnost da se s druge stranice uveze tekst. Mana je što ova značajka ne radi u svim slučajevima, dosta je ovisna o kojem se formatu radi za uvoženje. Još jedna značajka koju nudi su predefiniрани artikli koji sadrže određen sadržaj na traženu temu u kolekciji. Na kraju je moguće generirati kopiju knjige u PDF formatu, te postoji mogućnost pretvorbe knjige u slikovnicu. Cijena ovog alata je 67 američkih dolara, no postoje bolje verzije koje nude više mogućnosti, ali se moraju nadoplatiti. Još jedna mana ovog alata je manjak ažuriranja, značajke nisu promijenjene (unaprijeđene) već duže vrijeme.

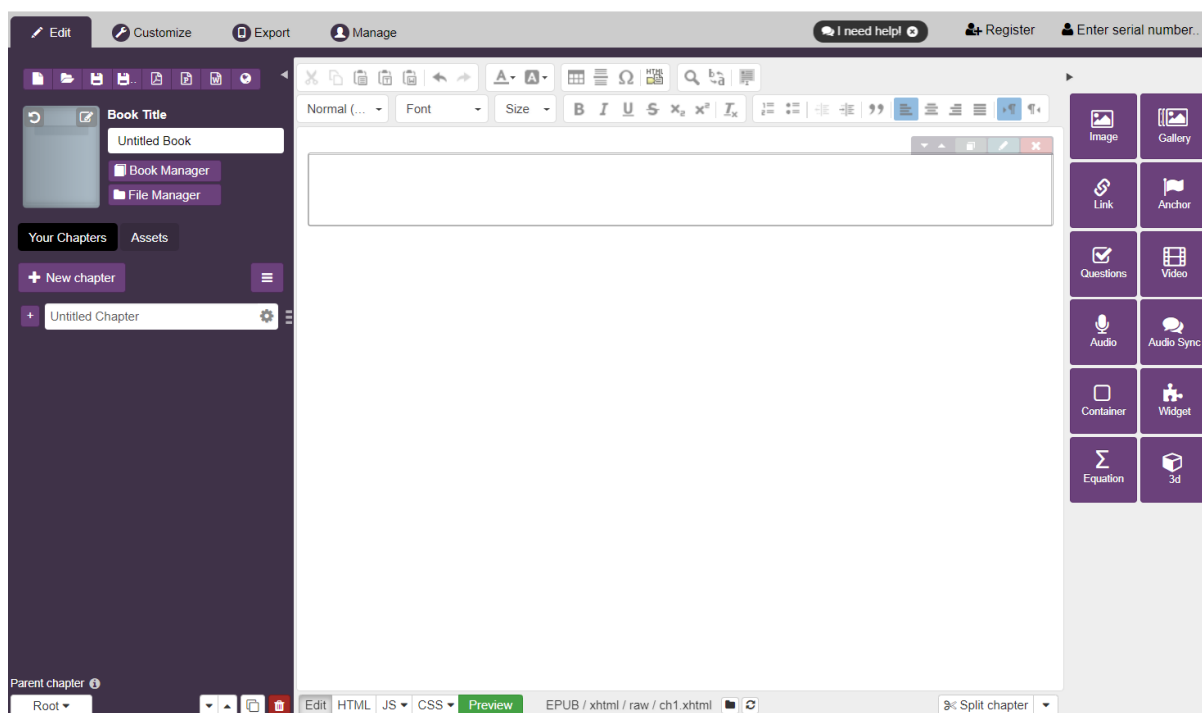




Sl. 2.1 Prikaz Sribbble sučelja.

## 2.2. Kotobee

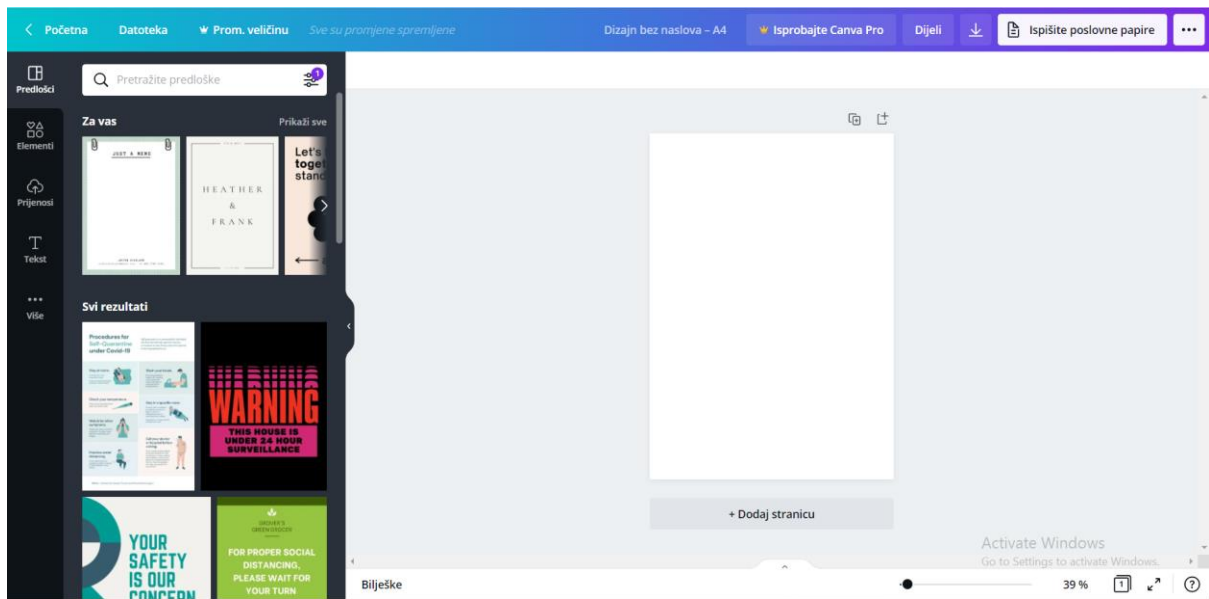
Kotobee je digitalna platforma za kreiranje interaktivnih knjiga za Android, Ipad, Iphone itd. Korisničko sučelje se sastoji od četiri glavna prozora: *Uredi*, *Prilagodi*, *Izvoz* i *Upravljač*. (Slika 2.2) U *Uredi* prozoru piše se sadržaj koji će knjiga sadržavati. Knjigu je moguće raditi od nule ili pak postoji mogućnost uvoza PDF, docx ili HTML datoteka. S desne strane, *Uredi* prozora nalaze se interaktivni elementi, kao što su video, slike, te postavljanje pitanja što je jedna od posebnih značajki koje ovaj alat nudi. *Prilagodi* prozor omogućuje uređivanje i prilagođavanje izgleda same knjige. Kotobee nudi formatiranje datoteke u svim tipovima dokumenata, koji se nalaze u *Izvoz* prozoru. U *Upravljač* prozoru moguće je postaviti knjigu kao privatnu, te dodati korisnike pomoću njihovog emaila kako bi mogli pristupiti samoj knjizi. Prednost ovog alata je što nudi najviše značajki za izradu interaktivne knjige od svih navedenih alata u ovom poglavlju, te svaku zamisao je moguće ostvariti pomoću ovog alata. Nedostatak je što za korisnike koji prvi put koriste ovaj alat, postoji previše značajki za koje je potrebno dugotrajno korištenje ovog alata kako bi se prilagodili na njih. Također cijena za ovaj alat je jako visoka, 150 američkih dolara, iako licenca traje doživotno.



Sl. 2.2 Prikaz Kotobee sučelja.

## 2.3. Canva

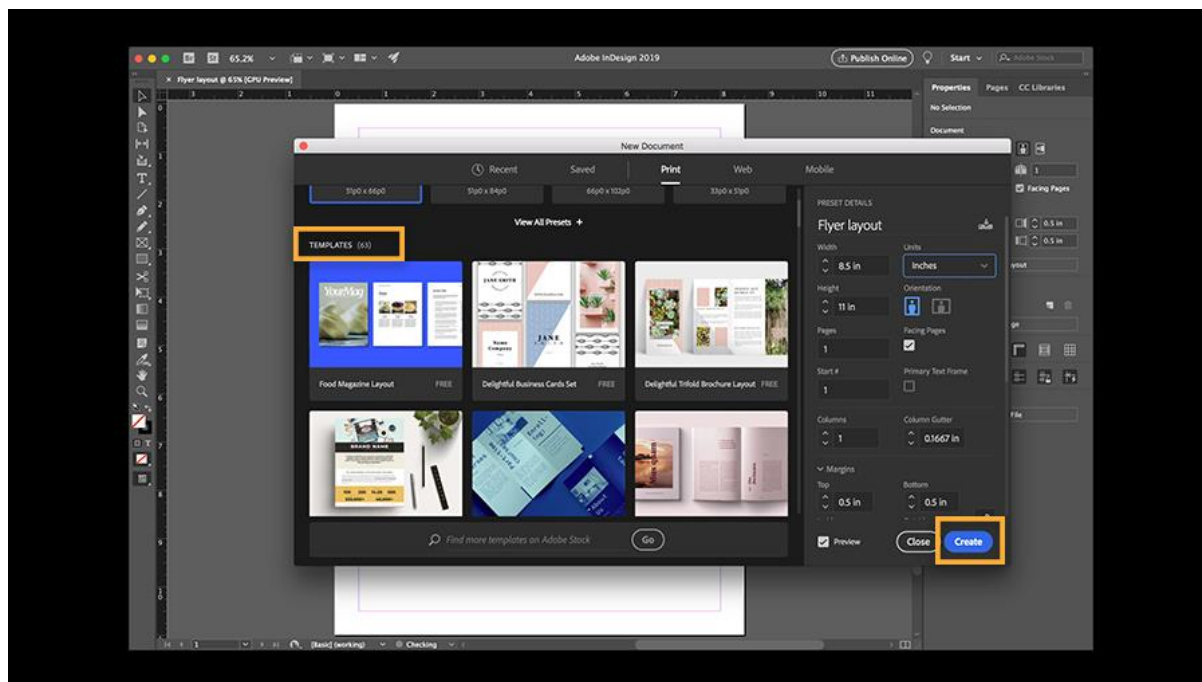
Canva je platforma za grafički dizajn, no moguće je u njoj kreirati i web knjige. Sustav radi na principu *drag and drop*, gdje je moguće dodati slike, videozapise, zvuk, te neke dodatne elemente kao simbole i ikone. Canva nudi gotove predloške koje omogućuju korisniku gotov predefiniрани dizajn, tako da se ne mora krenuti od nule. Dobra strana ovog alata je što je svaka komponenta fleksibilna i nije ograničena gdje je postavljena. Nedostatak alata je što se može napraviti maksimalno 100 stranica, te pošto Canva primarno nije alat za kreiranje knjiga ili sustav za objavljivanje dokumenata, nema ugrađen sustav za numeriranje stranica, pa se svaka stranica mora posebno numerirati od strane korisnika. Osnovna verzija je besplatna za koristiti, za razliku od Pro verzije koja košta 119,99 američkih dolara godišnje, no nudi se mogućnost probe od 30 dana besplatno.



Sl. 2.3 Prikaz Canva sučelja.

## 2.4. Adobe InDesign

Adobe InDesign je softverska aplikacija koja nudi mogućnosti izrade plakata, letaka, brošura, novina, prezentacija, te knjiga. Adobe InDesign spada u profesionalnu kategoriju alata za kreiranje dokumenata. Iako je ovo alat koji nudi najviše mogućnosti od svih, jako je težak za korištenje i potrebno dosta predznanja kako bi se mogao postići željeni rezultat, što se protivi konceptu ovog rada, koji teži ka jednostavnosti i mogućnosti korištenja bez potrebe za ikakvim predznanjem. Također spada u skuplju kategoriju alata gdje je mjesečna nakana 20,99 američkih dolara. Ovim alatom se većinom koriste ljudi koji se bave dizajniranjem knjiga. Sadrži jako puno značajki koje su nevezane za pravljenje interaktivne knjige. Velika prednost je što dolazi od uspješne kompanije Adobe, te rješenja za većinu problema je moguće pronaći na internetu.



Sl. 2.4 Prikaz Adobe InDesign sučelja. [1]

## 2.5. Microsoft Word i Google Docs

Microsoft Word je program za obradu teksta, te sastavni je dio programskog paketa Microsoft Office, proizveden od Microsofta. Google Docs je internetski alat za uređivanje teksta napravljen od Googlea. Ovi alati poznati su velikoj većini ljudi, korišteni su u školskom sustavu i vrlo su jednostavni za koristiti. Iako primarno nisu napravljeni za izradu interaktivnih knjiga, nude sve osnove koje su potrebne za kreiranje jedne. Velika prednost ovih alata je što je vrlo jednostavno implementirati sadržaj, te podjelu po poglavljima. Sadrže ugrađen sustav za numeriranje stranica, te besplatni su za učenike i studente, što je i ciljana populacija ovog rada. Nedostatak ovih alata je da nemaju velike mogućnosti raspoređivanja teksta i slika, vrlo su ograničeni po tom pitanju, te nemaju nekakve dodatne značajke koje bi doprinijele pravljenu interaktivne knjige.

## 2.6. Razlozi za razvijanje samostalnog rješenja

Navedeni alati nude jako puno mogućnosti, no za ljude koji se prije nisu bavili kreiranjem interaktivnih knjiga to može na prvu ruku izgledati jako komplicirano, te naposljetku ih spriječiti da kreiraju svoju knjigu. Cilj završnog rada je kreirati knjigu koja nudi neke značajke prije

navedenih alata, ali tako da te značajke budu prilagođene za ovaj rad. Pod tim se misli da knjiga koja će se izraditi bude prilagođena tako da samo objavljivanje priče radi na principu objava nekih društvenih mreža, npr. Twitter. To je upravo iz razloga što prije navedeni alati na prvu izgledaju kompleksno, te korisniku treba neko vrijeme da otkrije sve značajke koje takav alat nudi. Pristupom objava privlači se mlađa populacija jer je već upoznata tim sučeljem, te nije potrebno nikakvo predznanje kako bi se koristilo. Način na koji bi se knjiga ostvarila pomoću objava je jednostavan, svaka objava predstavlja jednu stranicu knjige. To omogućava da se jednostavno imitira sadržaj pomoću naslova svake objave, te za samo pisanje objave, što je u ovom slučaju kratak dio sveukupne priče, nije potrebno izdvojiti puno vremena kako bi se postigla knjiga. Potrebno je malo uloženog vremena od strane više korisnika kako bi se postigla knjiga, što je idealno za mlađu populaciju. Dakle cilj ovog rada je da svaki korisnik napiše svoj dio priče gdje se na kraju dobije imitacija knjige koja će se ostvariti pomoću web tehnologija. Još jedna stvar koja bi se uvela su povratne informacije za svakog korisnika. Svaki korisnik imat će mogućnost čitanja dijelova priče od drugih korisnika, te u slučaju da im se sviđa, označiti sa „Sviđa mi se“ gumbom. Tako će svaki korisnik dobiti brzu povratnu informaciju o dijelu koji je napisan, te time postoji mogućnost da se iskažu jako talentirani pojedinci, koje će se ovim putem potaknuti na ulaganje više vremena u pisanje djela. Još jedna stavka prije navedenih alata je da se većinom moraju platiti kako bi se koristili, što opet ne ide u korist mlađoj populaciji. Ovaj rad je besplatan za korištenje.

### 3. RAZVOJNI OKVIRI

Razvojni okvir (engl. *framework*) je platforma za razvoj softverskih aplikacija, prema [2]. Nudi temelje pomoću kojih softver programeri mogu izraditi program za specifičnu platformu. Razvojni okvir sadrži predefinirane klase i funkcije koje se mogu koristiti za procesiranje unosa, upravljanje hardverskih uređaja i interakciju sa softverskim sistemom. To pojednostavljuje proces razvoja jer programeri ne moraju svaki put kada prave novu aplikaciju pisati ponovno isti kod. Omogućuje standardni način za izgradnju i implementaciju aplikacija i predstavlja univerzalno softversko okruženje koje pruža posebne funkcije kao dio veće softverske platforme za olakšavanje razvoja softvera aplikacije, proizvoda i različitih rješenja, prema [3]. Razvojni okvir također može sadržavati knjižnice kodova, prevoditelj(engl. *Compiler*) i druge programe korištene u softverskom procesu izgradnje.

Tipovi razvojnih okvira:

- UI (engl. *User Interface*)/CSS (engl. *Cascading Style Sheets*) razvojni okviri – Bootstrap, Materialize, Bulma, Tailwind itd.
- *Front end* JS(JavaScript) razvojni okviri – React, Vue, Angular itd.
- *Back end* razvojni okviri – Laravel, Ruby on Rails, Express, Django itd.

#### 3.1. Korištene tehnologije

Kako bi se ostvario cilj ovog rada, izrada interaktivne knjige, postoji mnogo alata pomoću kojih se to može izvesti. Za ovaj rad odabran je Laravel razvojni okvir iz razloga što je pomoću njega moguće ostvariti ovaj projekt na lagan, ali učinkovit način, a da se pri tome prikažu mnoge mogućnosti ovakvog alata. Laravel je MVC(engl. *Model-View-Controller*) web razvojni okvir pisan u PHP-u. Dizajniran je kako bi unaprijedio kvalitetu softvera smanjujući troškove prvobitne proizvodnje, troškove održavanja, te unapređuje iskustvo rada s aplikacijom na način da nudi izražajnu sintaksu i jezgru koja sadrži set funkcionalnosti pomoću koje se ušteduje jako puno vremena koje bi inače bilo potrošeno na samoj implementaciji. Laravel je dizajniran s filozofijom korištenja konvencije iznad konfiguracije. To znači da koristi pametne pretpostavke o tome što se želi postići u određenim situacijama i realizaciju cilja uz pisanje puno manje koda. Laravel je dizajniran na način da je što minimalniji, ali funkcionalan. Lakše je razumjeti manje kodne baze i Laravelov fokus je implementacija rješenja na način koji je čist, jednostavan i elegantan. Laravel je jedan od rijetkih razvojnih okvira koji nudi istintsku kodnu modularnost. To je postignuto kroz

kombinaciju *drivera* i njegovih sistemskih paketa. *Driveri* dozvoljavaju da se jednostavno promijeni i produži predmemoriranje, sesija, baza podataka i funkcionalnost autentifikacije. Koristeći pakete, sav kod koji je vlastito napisan može se upakirati pa ponovno koristiti za vlastite svrhe, ili pak pružiti Laravel zajednici. Ovaj koncept je jako bitan jer sve što je moguće napisati u Laravelu, može se upakirati, od jednostavnih biblioteka pa do cijelih web aplikacija. Laravel-bundle web stranica omogućuje pretraživanje različitih paketa koji su izgrađeni od strane zajednice, te omogućuje da se pokažu samostalno izrađeni paketi. To je jako vrijedan „*third-party*“ podsistem i biblioteka koja može drastično olakšati proces izrade web aplikacije. Laravel također nudi najsvremenije alate za interakciju s bazom podataka. Migracija baze podataka(engl. *Database migration*) omogućava jednostavno dizajniranje i izmjenjivanje baze podataka tako da je neovisno o platformi.

Laravelov tečni graditelj upita(engl. *Fluent Query Builder*) sažima sve različitosti između različitih tipova baza podataka. *ActiveRecord* je oblikovni obrazac koji pohranjuje objektne podatke iz memorije u relacijsku bazu, prema [4]. Laravelova *ActiveRecord* implementacija zove se Eloquent. Interakcija s bazom podataka na objektno orijentiran način je moderni standard. S Eloquentom, moguće je kreirati, vratiti, ažurirati i brisati zapise iz baze. Laravel također dolazi s ugrađenim sučeljem za komandu liniju zvan Artisan. S Artisanom, programer može interagirati s aplikacijom da pokrene radnje kao što su migracija, testovi ili pak neki zakazani zadaci. Artisan je također u potpunosti proširiv tako da je moguće napisati bilo kakvu funkcionalnost koja je potrebna. Laravelov jednostavno upravljiv sistem usmjeravanja(engl. *routing system*) omogućuje da se lako upravlja URL-ovima vlastite stranice. Koristeći ugrađeni HTML pomoćnik, moguće je kreirati linkove unutar vlastite stranice koji će se automatski ažurirati ako se promijeni URL, što uvelike olakšava posao održavanje stranice.

## 3.2 PHP

S obzirom da je Laravel razvojni okvir pisan pomoću PHP-a, važno je znati njegove osnove. PHP je skriptni jezik dizajniran posebno za web, prema [5]. Unutar HTML stranice, moguće je ugraditi PHP kod koji će se izvršiti svaki put kada je stranica posjećena. PHP kod je interpretiran kod web servera i generira HTML ili drugi izlaz kojeg posjetitelj vidi. PHP je začeo 1994. i originalno je rad jednog čovjeka, Rasmusa Lerdorfa. Već od listopada 2002. godine korišten je u više od devet milijuna domena diljem svijeta i taj broj raste ubrzo. PHP je proizvod otvorenog koda(engl. *open source*). Svaki čovjek ga može koristiti, mijenjati i redistribuirati bez novčanih naknada. PHP

originalno je stajalo za engl. *Personal Home Page*, ali poslije je promijenjeno u engl. *PHP Hypertext Preprocessor*. Neki glavni PHP-ovi natjecatelji su Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) i Allaire ColdFusion. Uspoređujući s tim proizvodima, PHP ima mnogo prednosti, uključujući sljedeće:

- Visoke performanse
- Sučelja za mnoge različite baze podataka
- Ugrađene knjižnice za mnogo općih web zadataka
- Niska cijena
- Lagan za učenje i korištenje
- Prenosiv
- Dostupnost *source* koda

### 3.3. MySQL

MySQL je vrlo brz, snažan sustav za upravljanja relacijskom bazom podataka. Baza podataka nudi učinkovitost spremanja, traženja, sortiranja i dohvaćanja podataka. MySQL server kontrolira pristup podacima kako bi osigurao da više korisnika može raditi istovremeno na bazi, osigurava brzi pristup, te osigurava da samo ovlaštene osobe mogu pristupiti bazi. Osim toga, MySQL je višekorisnički i višenitni(engl. *multithreaded*) server. Koristi SQL (engl. *Structured Query Language*), standardni upitni jezik za baze podataka, prema [5]. MySQL je javno dostupan od 1996., ali povijest razvoja još od 1979. MySQL je dostupan pod licencom otvorenog koda(engl. *Open source*), ali komercijalne licence su također dostupne ako su potrebne. Neki od glavnih MySQL-ovih konkurenata su PostgreSQL, Microsoft SQL Server i Oracle. MySQL ima mnoge prednosti, uključujući i sljedeće:

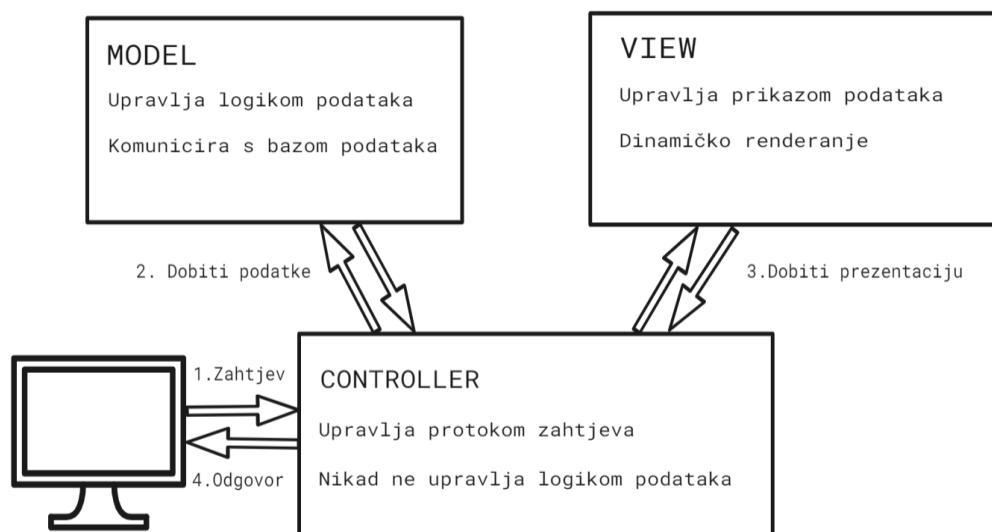
- Visoke performanse – MySQL je neupitno brz, rezultati nekih mjerenja su dostupni na glavnoj MySQL stranici
- Niske cijene – MySQL je dostupan bez plaćanja, pod licencom otvorenog koda, ili pod niskom komercijalnom cijenom
- Lagan za konfigurirati i naučiti – većina modernih baza podataka danas koriste SQL, u slučaju korištenja nekih drugih sličnih izvedbi lako se prilagoditi SQL-u, te MySQL je lakši za postaviti nego neki slični proizvodi
- Prenosiv – MySQL se može koristiti na razim Unix sustavima kao i na Microsoft Windows



- Izvorni kod je dostupan – kao što je slučaj i kod PHP-a, moguće je dobiti i mijenjati izvorni kod za MySQL

### 3.4. MVC

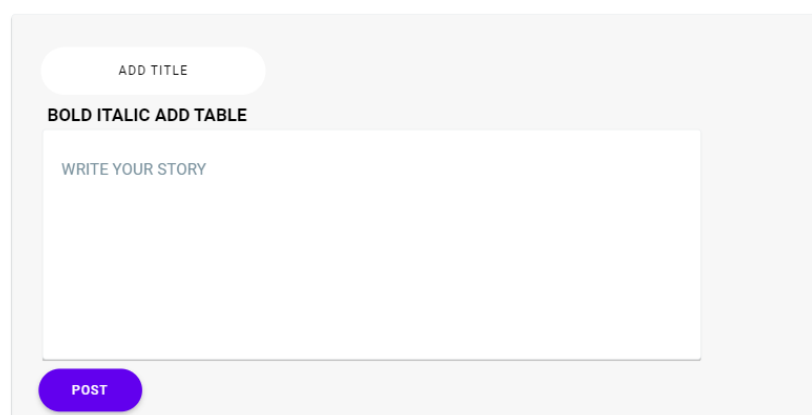
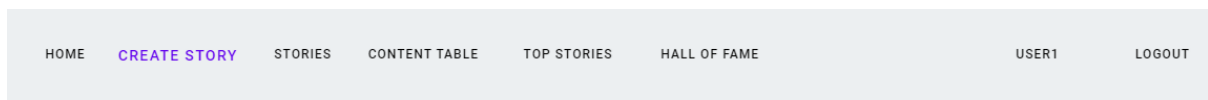
Jako poznati MVC (engl. *Model-View-Controller*) oblikovni obrazac koristan je način da se strukturira vlastiti interaktivni softver sustav. Glavna ideja ovog obrasca je da se odvoji korisničko sučelje od podataka koji se nalaze u pozadini, koje korisničko sučelje na kraju prezentira. U MVC-u pogled(engl. *View*) prikazuje informacije korisniku i zajedno s upravljačem(engl. *Controller*), koji procesira korisnikove interakcije, sadrži aplikacijsko korisničko sučelje. Model (engl. *Model*) je dio aplikacije koji sadrži i informaciju reprezentiranu od strane pogleda i logiku koja mijenja te informacije kao odgovor korisničke interakcije. Koristeći MVC i PAC (engl. *Presentation/Abstraction/Control*) oblikovne obrasce olakšano je razvijati i održavati aplikaciju jer se izgled aplikacije može drastično mijenjati bez da se mijenja struktura podataka i aplikacija može lagano održavati različita sučelja, kao što je više jezika, ili različiti setovi korisničkih dozvola. Kao odgovor promjenama u modelu, upravljač inicira stvaranje novog aplikacijskog pogleda. Tradicionalno korišten za grafička korisnička sučelja, ovaj obrazac postao je popularan za dizajniranje web aplikacija.



Sl. 3.1 Prikaz MVC modela.

## 4. REALIZACIJA INTERAKTIVNE KNJIGE

Koncept ovog rada je da se imitira kreacija knjige pomoću korisničkih objava, gdje bi svaka objava imala neke interaktivne elemente, te činila jednu stranicu knjige. Time se postiže knjiga koja sadrži onoliko stranica koliko ima korisničkih objava. Svaka objava sadržavati će neke mogućnosti uređivanja, implementirati će se sučelje za uređivanje koje sličí sučelju Worda. Tako će korisnici dobiti jednostavno sučelje s kojim su upoznati i nije teško za koristiti. Implementirati će se sustav za registraciju korisnika, što će omogućiti da više korisnika doprinosi sadržaju knjige. Kreirati će se sadržaj na način da će svi naslovi biti ispisani na stranici, te korisnik pritiskom na naslov moći će otići na tu priču. Također uvest će se „Sviđa mi se“ (engl. *like*) gumb, koji će dodati još jednu razinu interaktnosti između korisnika i objave. Pomoću „Sviđa mi se“ gumba, moći će se sortirati priče tako da korisnik dobije informaciju o najuspješnijim pričama, te najboljim autorima, ovisno o tome koliko imaju *likeova*. Interaktivna knjiga koja se želi postići ovim radom pokušaj je pojednostavljenja postojećih alata za izradu interaktivnih knjiga. Cilj je približiti mlađu populaciju k pisanju djela, na način da im se pruži već prije poznato sučelje, koje je jednostavno za koristiti, besplatno, te ne uzima puno vremena.



Sl. 4.1 Ogledni primjer krajnjeg rezultata

## 4.1. Kreiranje projekta

Kreiranje novog projekta vrši se naredbom *laravel new ime\_projekta* koja se piše u terminalu. Nakon kreiranja novog projekta dobijemo datoteku s raznim predefiniranim datotekama, koje sadrže mnoge funkcionalnosti koje sam programer ne mora implementirati, što jako ubrzava posao.

## 4.2. Laravel Mix i Tailwind

Laravel Mix je dodatak koji omogućava upravljanje *assetima*. Tailwind je razvojni okvir koji dolazi s ugrađenim klasama za uređivanje izgleda stranice. Laravel Mix prevodi kod koji je pisan od strane Tailwinda, tako da preglednik razumije kod i može ga prikazati. Laravel to postiže kroz datoteku zvanu *webpack.mix.js*. Dakle, to je datoteka u kojoj se prevodi JS kod. Sada je moguće koristiti predefinirane klase koje nudi Tailwind za uređivanje stranice.

<i>Linija</i>	<i>Kod</i>
14:	<code>mix.js('resources/js/app.js', 'public/js')</code>
15:	<code>.postCss('resources/css/app.css', 'public/css', [</code>
16:	<code>require('tailwindcss')</code>
17:	<code>]);</code>

Sl. 4.2 Webpack.mix.js datoteka.

## 4.3. Registracija

Kako bi se izbjegao problem od pisanja cijelog koda u rute(engl.*routes*) mapi, što bi jako otežalo održavanje aplikacije, kreiran je upravljač(engl. *Controller*) koji će se brinuti za funkcionalnosti registracije. Kreiranje novog upavljača vrlo je jednostavno u Laravelu pomoću linije *php artisan make:controller ImeDatoteke\\ImeControllera* koja se upisuje u terminalu. Kako bi se koristio kreirani upravljač potrebno je u *routes* mapi dodati slijedeću liniju koda:

**Linija Kod**

```
15: Route::get('/register', [RegisterController::class,'index'])->name('register');
```

Sl. 4.3 Prikaz implementacije funkcionalnosti upravljača.

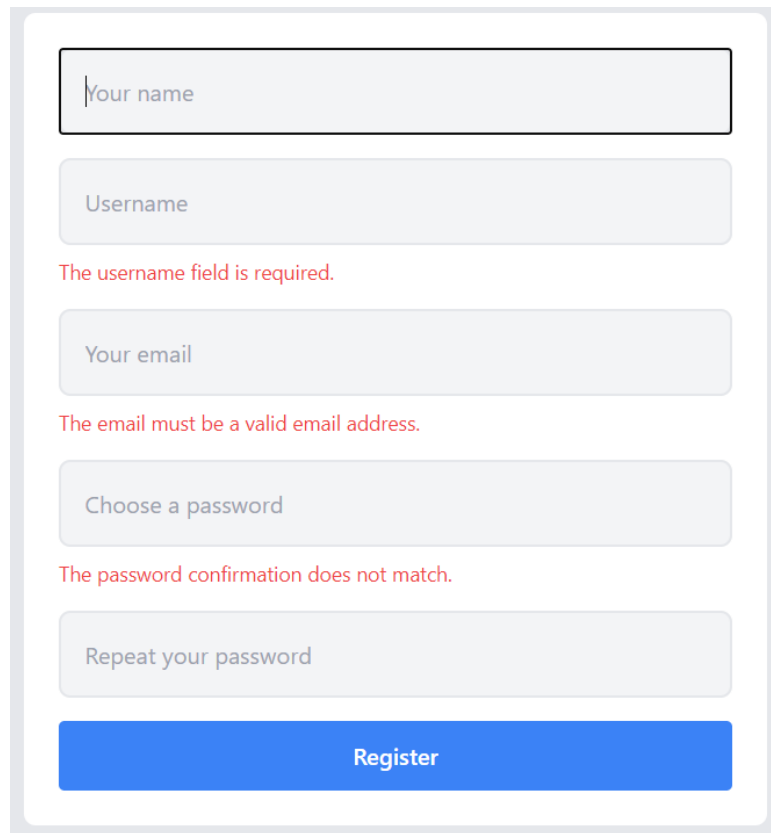
Nakon toga potrebno je implementirati funkcionalnosti za registraciju, te u *RegisterControlleru* kroz funkcija `index`, vraća se odgovarajući pogled (engl. *View*).

**Linija Kod**

```
8: class RegisterController extends Controller
9: {
10:     public function index(){
11:         return view('auth.register');
12:     }
```

Sl. 4.4 Prikaz klase *RegisterController* koja vraća pogled *register*.

Zatim se implementira forma za *View register*, Slika 4.5.

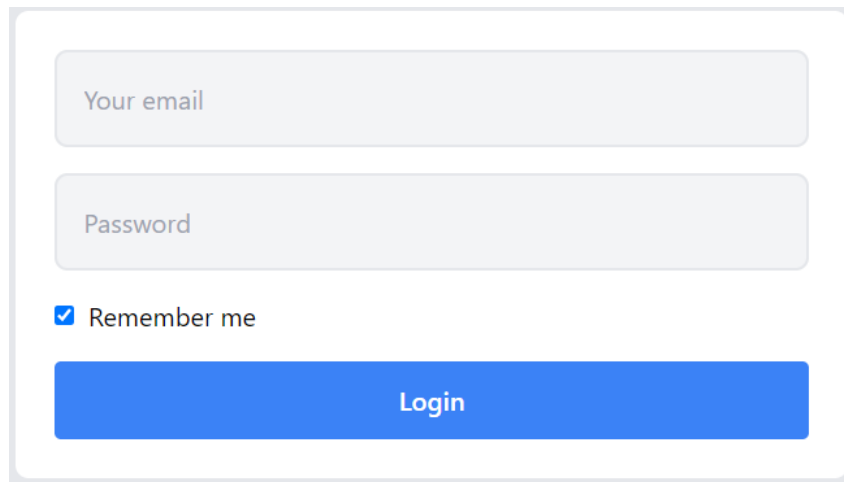


The image shows a registration form with the following elements:

- A text input field labeled "Your name" with a cursor.
- A text input field labeled "Username" with a red error message below it: "The username field is required."
- A text input field labeled "Your email" with a red error message below it: "The email must be a valid email address."
- A text input field labeled "Choose a password" with a red error message below it: "The password confirmation does not match."
- A text input field labeled "Repeat your password".
- A blue button labeled "Register".

Sl. 4.5 Prikaz gotove *register* forme

Dodana su ograničenja i upozorenja u slučaju da određeno polje nije ispunjeno, te korisnik dobiva povratnu informaciju ukoliko nešto nije dobro unešeno. Nakon toga kreirani su *Login* i *Logout* upravljači, te odgovarajuća forma za njih, Slika 4.6 prikazuje *Login* formu, koja sadrži dodatnu opciju „Zapamti me“ (engl. *Remember me*). Opcija „Zapamti me“ generira token u bazi pomoću kojeg možemo dobiti neke podatke o korisniku. U navigaciji *Login* ili *Logout* mogućnosti se prikazuju prikladno s tim da li je korisnik prijavljen ili ne. To je postignuto pomoću ugrađenih *blade* metoda *@auth*, koja provjerava da li je korisnik prijavljen, te *@guest*, koja provjerava da li korisnik nije prijavljen. Nakon kreiranih upravljača i formi za registraciju potrebno je uvesti model i tablicu kako bi se korisnikovi unešeni podaci mogli spremati. Kada se kreira novi projekt u Laravelu, odmah se dobije gotov model korisnika (engl. *User model*) kojeg se lako može prilagoditi za potrebe ovog projekta.

The image shows a login form with a light gray border. It contains four main elements: a text input field labeled 'Your email', another text input field labeled 'Password', a checkbox labeled 'Remember me' which is checked, and a blue button labeled 'Login'.

Sl. 4.6 Prikaz *Login* forme

#### 4.4. Objave

Za kreiranje objava napravljena je posebna forma, Slika 4.7. Korisnik ima mogućnost unositi naslov, te sadržaj priče. Kreiran je upravljač koji upravlja funkcionalnostima forme objave(engl. *post*). U njemu su deklarirane funkcije sačuvaj(engl. *store*) i uništi(engl. *destroy*). Funkcija „sačuvaj“ se brine za spremanje podataka koje je korisnik uveo, te ima razna ograničenja kako bi sam unos bio bolje optimiziran. Pomoću funkcije „uništi“ korisnik može obrisati svoju objavu. Uvedena su ograničenja da samo korisnik koji je kreirao objavu ju ujedno i može obrisati. To je postignuto pomoću *Policies*, gdje se mogu definirati razne funkcionalnosti i ograničenja koje se poslije mogu koristiti. Formu koja je zaslužena za kreiranje objava, obložena je u ugrađenoj *blade* metodi *@can* koja će provjeriti da li je korisnik koji je prijavljen dozvoljen da obriše određenu objavu, Slika 4.8.

The image shows a web form for creating a post. At the top, there is a text input field with the label 'Title: Enter Title'. Below this is a large, empty text area with the placeholder text 'Write your story!'. At the bottom left of the form is a blue button with the text 'Post'.

Sl. 4.7 Prikaz *post* forme

### ***Linija Kod***

```
16: @can('delete', $story)
17: <form action="{{ route('posts.destroy', $story) }}"
18: method="post">
19: @csrf
20: @method('DELETE')
21: <button type="submit" class="text-blue-500">Delete</button>
22: </form>
23: @endcan
```

Sl. 4.8 Kod za *delete button*

Kao i kod registracije, potrebno je kreirati tablicu i model za spremanje objava. Važno je dodati „Sadrži mnogo“ (engl. *hasMany*) odnos između *User* i *Story(Post)* modela. Taj odnos omogućuje da svaki korisnik može imati više objava. Pomoću ugrađene funkcije za paginaciju određeno je da se samo jedna objava vidi po stranici, što omogućava bolju preglednost, Slika 4.9.

### ***Linija Kod***

```
16:     $stories = Story::with('user', 'likes')->paginate(1);
```

Sl. 4.9 Kod za paginaciju.

Na kraju, dodano je ograničenje da korisnik koji nije prijavljen ne može pristupiti opcijama da kreira novu priču, ili da pregleda priče, nego je preusmjeren prema *Login* formi. To je ostvareno pomoću ugrađenog posredničkog *softvera* (engl. *Middleware*) koji filtrira i pregledava HTTP zahtjeve koji dolaze na stranicu.

## **4.5. Povratne informacije**

Povratne informacije dobit će se tako da se uvede „sviđa mi se“ (engl. *like*) gumb. Koncept je vrlo jednostavan, što više “sviđanja” neka priča ima to je ona bolja. Ponovno je potrebno kreirati upravljač, model i tablicu u kojoj će se spremati koliko koji korisnik ima oznaka „sviđa mi se“. Važno je opet postaviti „*hasMany*“ odnos između *Story(Post)* i *Like* modela. (Slika 4.10) Svaki korisnik je ograničen da samo jednom može označiti objavu sa „sviđa mi se“. Ako je određena objava označena kao „sviđa mi se“, korisnik ima mogućnost da makne oznaku.

### ***Linija Kod***

```
25:     public function likes() {  
26:         return $this->hasMany(Like::class);  
27:     }
```

Sl. 4.10 Prikaz koda za „*hasMany*“ odnos između *Story* i *Like* modela



## 4.6. Urednik za tekst

Kako bi korisnik imao mogućnost uređivanja teksta, dodavanja tablica, linkova itd. implementiran je urednik za tekst (engl. *Text editor*). Pomoću njega moguće je dodavati pojedine HTML oznake, koje će se poslije iscrtati na prikazu objava. Urednik se implementira tako da se postavi skripta u kojem je definiran i inicijaliziran, te pomoću njegove vrijednosti (*id*), moguće ga je koristiti na mjestima gdje je potreban. (Slika 4.11) Kod prikaza svih objava, potrebno ga je i tamo implementirati, ali tako da mu se vrijednost „*isReadOnly*“ postavi na istina (engl. *true*). To će omogućiti da se HTML oznake prikažu, a da se pri tom onemoguće daljnje promjene na samom sadržaju.

### Linija Kod

```
39: <script>
40:   ClassicEditor
41:     .create( document.querySelector( '#editor' ) )
42:     .catch( error => {
43:       console.error( error );
44:     } );
45: </script>
```

Slika 4.11 Kod za implementaciju urednika

Title: Lorem2 Stjepan

The screenshot shows a rich text editor interface. At the top, there is a title field containing "Lorem2 Stjepan". Below the title is a toolbar with various icons for text formatting (bold, italic, link), alignment, and other functions. The main text area contains several paragraphs of placeholder text (Lorem ipsum) with various HTML tags applied, such as bold, italic, and underlined text. A table is also visible in the text area, containing the URL "https://www.ferit.unios.hr/2021/". The table has 2 columns and 3 rows. At the bottom left of the editor, there is a blue "Post" button.

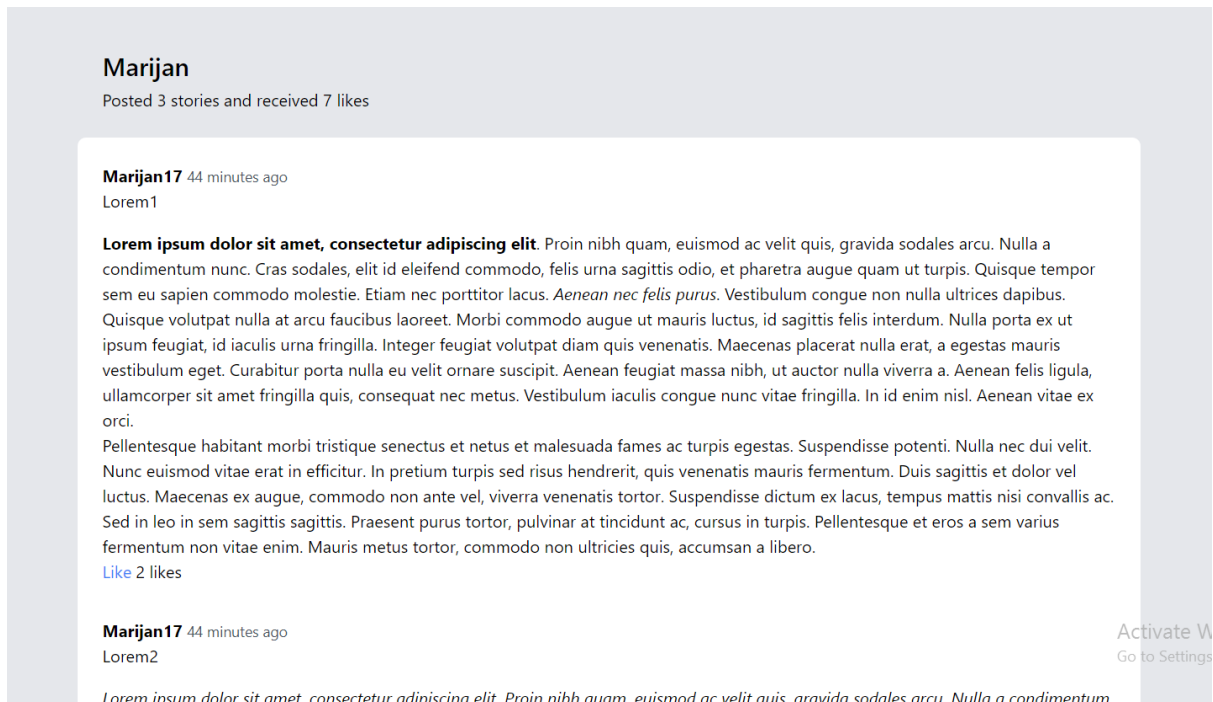
F	https://www.ferit.unios.hr/2021/	rd
4	https://www.ferit.unios.hr/2021/	4
1	4	5

Post

Sl. 4.12 Prikaz urednika

## 4.7. Pregled korisnika

Još jedna dodana mogućnost je pregled korisnikove stranice. Pritisak na korisnikovo ime u svim objavama, preusmjerit će korisnika na stranicu s detaljima o traženom korisniku. Na toj stranici prikazana je količina priči koju je određeni korisnik napravio, količina oznaka „svidi mi se“, te prikaz svih priči koje je korisnik objavio. (Slika 4.13)



Sl. 4.13 Prikaz korisnikove stranice

Kako bi se ovo ostvarilo, potrebno je u rutama predati id korisnika na kojeg je pritisnuto. Kreiran je novi upravljač koji ima funkcionalnost prema slici 4.14. Kako bi pristupili broju oznaka „svidi mi se“ za svakog korisnika, potrebno je u modelu *User* definirati odnos „ima mnogo kroz“ (engl. *hasManyThrough*), nakon toga je moguće dobiti broj oznaka „svidi mi se“ za svakog korisnika u pogledu. (Slika 4.15)

Linija	Kod
9:	<code>public function index(User \$user){</code>
10:	<code>    \$stories = \$user-&gt;stories()-&gt;with(['user', 'likes'])-</code>
	<code>&gt;paginate(20);</code>
11:	<code>    return view('users.posts.index', [</code>
12:	<code>        'user' =&gt; \$user,</code>
13:	<code>        'stories' =&gt; \$stories</code>
14:	<code>    ]); }</code>

Sl. 4.14 Kod upravljača za korisnikovu stranicu

### **Linija Kod**

```
55: public function receivedLikes() {
56:     return $this->hasManyThrough(Like::class, Story::class);
57: }
```

Sl. 4.15 Odnos *User* i *Like* modela

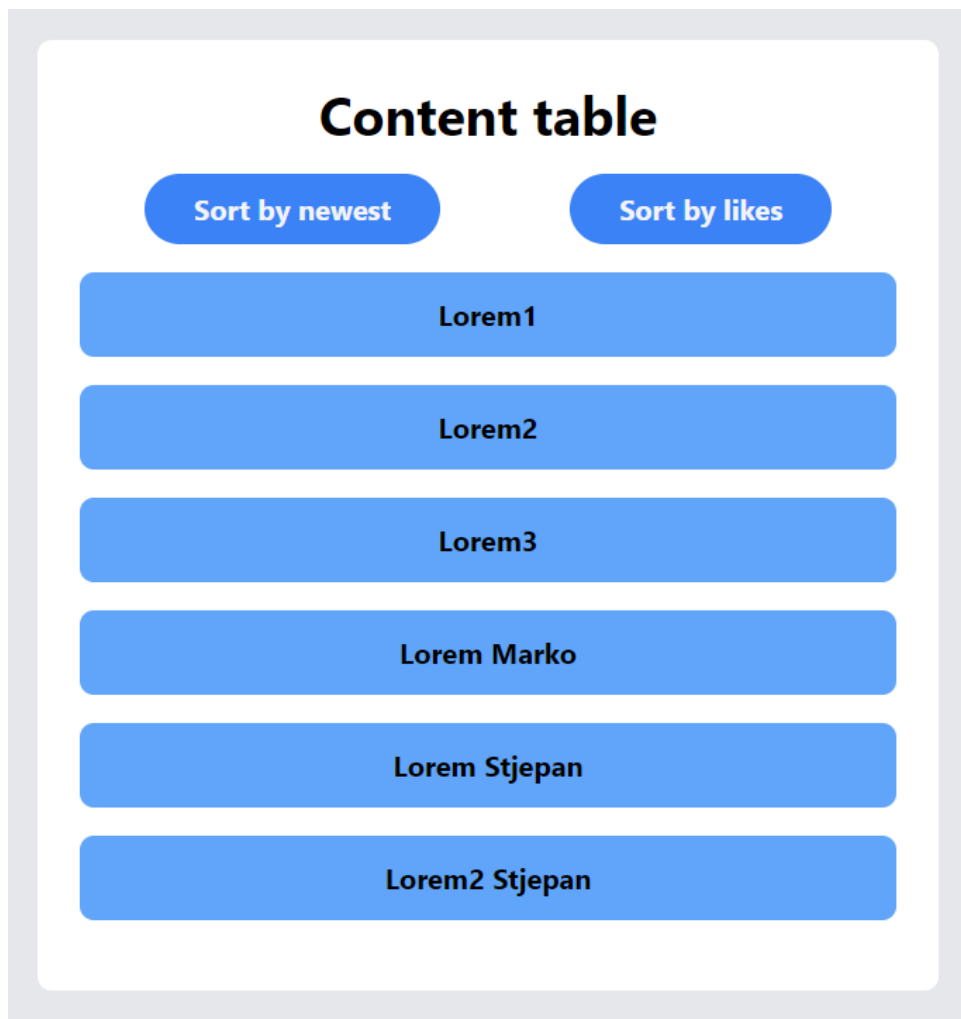
## **4.8. Sadržaj i sortiranja**

Sadržaj sadrži poredane priče prema najstarije kreiranim, te ima mogućnosti sortiranja priči prema najnovije kreiranim, te sortiranje po pričama koje sadrže najviše oznaka „sviđa mi se“. Sadržaj je postignut tako da je iskorišten već prethodno definiran odnos između različitih tablica u bazi. Samo sortiranje postignuto je pomoću ugrađenih Laravel funkcija *latest()*, gdje se u pogled vraća lista objekata, koja je sortirana prema najnovije kreiranim pričama. U pogledu se kreira tablica naslova priči, gdje je dodana mogućnost pritiska na naslov, što će odvesti korisnika na sam sadržaj priče. Za sortiranje priči prema broju oznaka „sviđa mi se“, potrebno je koristiti Laravel funkciju *withCount()*, gdje se kao argument predaje prethodno definirana funkcija za odnos *likes()*, koja se nalazi u modelu *Story*. (Slika 4.16)

### **Linija Kod**

```
24: public function sortedByLikes(User $user) {
25:     $stories = Story::withCount('likes')-
    >orderByDesc('likes_count')->get();
26:     return view('posts.contentTable', [
27:         'stories' => $stories
28:     ]);
29: }
```

Sl. 4.16 Kod za sortiranje priči prema broju oznaka „sviđa mi se“



Sl. 4.17 Prikaz tablice za sadržaj

#### 4.9. Najbolje priče i autori

Za kraj korisniku je omogućen pregled prema najboljim pričama i autorima. Za sortiranje najboljih priči iskorišten je isti kod prema slici 4.16, ali je vraćen drugi pogled. Kako bi se dobili najbolji autori potrebno je sortirati korisnike prema broju dobivenih oznaka „sviđa mi se“. (Slika 4.18) Ponovno je iskorištena Laravel funkcija *withCount()*, kojoj se kao argument predaje prethodno definirana funkcija za odnos *receivedLikes()*, te pomoću *orderByDesc* autori su sortirani silazno od najviše dobivenih oznaka „sviđa mi se“ prema najmanje.

### **Linija Kod**

```
40: public function displayTopAuthors(User $user){
41:     $user = User::withCount('receivedLikes')-
>orderByDesc('received_likes_count')->get();
44:     return view('posts.hallOfFame, [
45:         'users' => $users
46:     ]);
47: }
```

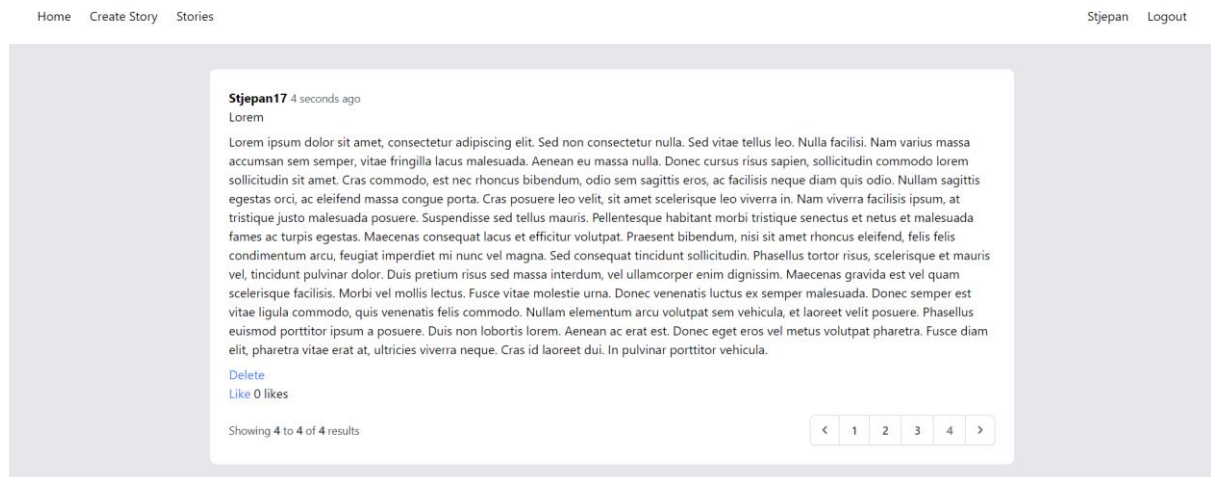
Sl. 4.18 Kod za sortiranje autora prema najviše dobivenih „svida mi se“ oznaka



Sl. 4.19 Prikaz tablice za najbolje autore

## 4.10. Krajnji rezultat

Krajnji rezultat je jednostavna web aplikacija koja nudi mogućnosti da se korisnik registrira, objavi svoj dio priče i da primi povratne informacije od drugih korisnika. Kroz razne ugrađene funkcionalnosti koje Laravel nudi, optimizirana je stranica kako bi bila sigurna i funkcionalna. (Slika 4.20)



Sl. 4.20 Krajnji rezultat

## 5. ZAKLJUČAK

U ovom završnom radu proučene su razne mane i prednosti alata za izradu interaktivnih knjiga. Interaktivna knjiga je koncept koji pokušava unaprijediti svijet knjiženstva, na način da privuče što veći broj ljudi. Više ljudi ujedno znači i više umova, te ideja za novim pričama. Prednost ovog koncepta je da svaki čovjek može sudjelovati i podijeliti svoj dio priče, koji možda prije ne bi, a da pri tom ostane anonimna. Pomoću Laravel razvojnog okvira ostvarena je ideja, tako što je izgrađen siguran sustav za registraciju korisnika, sustav za postavljanje novih priča, te sustav za interakciju s pričama. Krajnji rezultat može se primjeniti kao *online* način pisanja priči/djela. Kroz rad su spomenuta razna druga razvojna okruženja pomoću kojeg je moguće ostvariti ovaj rad, prednosti koji ovakvi alati donose, te mane koje bi se mogle u skorijoj budućnosti riješiti. Na kraju, treba spomenuti da ovo nije jedini način izrade ovakve web aplikacije, te da postoji mnogo drugih razvojnih okruženja pomoću kojih bi se web aplikacija mogla ostvariti možda na lakši način, no svako razvojno okruženje dolazi sa svojim prednostima i manama.

## LITERATURA

- [1] „Adobe InDesign“, dostupno na: <https://helpx.adobe.com/menar/in-design/how-to/new-document-in-in-design.html> [31.8.2021.]
- [2] „Framework“, dostupno na: <https://techterms.com/definition/framework> [31.8.2021.]
- [3] „Software Framework“, dostupno na: [https://en.wikipedia.org/wiki/Software\\_framework](https://en.wikipedia.org/wiki/Software_framework) [31.8.2022.]
- [4] „Laravel ActiveRecord“, dostupno na: <http://www.laravelinterviewquestions.com/trick/active-record-implementation-in-laravel-mjg/> [31.8.2022.]
- [5] L. Welling, L. Thomson, PHP and MySQL Web Development, Developer's library



## **ŽIVOTOPIS**

Autor ovog završnog rada, Stjepan Klarić, student je treće godine Sveučilišnog preddiplomskog studija Računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek (FERIT). Prije upisivanja na FERIT, bio je učenik Tehničke škole Ruđera Boškovića u Vinkovcima smjera Tehnička gimnazija.

---

Potpis autora

## SAŽETAK

**Naslov:** Interaktivna knjiga primjenom web tehnologija

U prvom dijelu opisani su alati za izradu interaktivnih knjiga i web tehnologije koje se koriste za ostvarivanje rada. Ideja interaktivne knjige je da privuče veći broj ljudi u svijet pisanja djela. To se ostvaruje s činjenicom da je internet danas pristupačan skoro svakoj osobi, te brzina kojom informacije teku znatno posjpešuju šanse da svaka priča dođe do više čitatelja. Pomoću Laravel razvojnog okruženja osigurano je da se svaki korisnik može registrirati, kreirati svoj dio priče, te dati povratne informacije drugim djelima. U zadnjem poglavlju opisan je postupak izrade interaktivne knjige. Krajnji rezultat je jednostavna web aplikacija koja nudi piscima nov način stvaranja djela.

**Ključne riječi:** knjiga, Laravel, PHP, razvojni okvir, web

## **ABSTRACT**

**Title:** An interactive applied with web technology

The first part describes the tools for creating interactive books and web technologies that are used to achieve the work. The idea of the interactive book is to attract more people to the world of writing works. This is achieved with the fact that the Internet is now accessible to almost everyone, and the speed with which information flows significantly enhances the chances that each story reaches more readers. The Laravel development environment ensures that each user can register, create their own part of the story, and give feedback on other works. The last chapter describes the process of creating an interactive book. The end result is a simple web application that offers writers a new way to create works.

**Keywords:** book, framework, Laravel, PHP, web

## **PRILOZI**

Na CD-u:

1. Završni rad “ Interaktivna knjiga primjenom web tehnologija ” u *.docx* formatu
2. Završni rad “ Interaktivna knjiga primjenom web tehnologija ” u *.pdf* formatu
3. Izvorni kod praktičnog dijela rada

