

RAZVOJ VEB-APLIKACIJE ZA DRUŠTVENU INTERAKCIJU S INTEGRACIJOM EKSTERNOG SISTEMA ZA PLAĆANJE**DEVELOPMENT OF A WEB APPLICATION FOR SOCIAL INTERACTION WITH INTEGRATION OF AN EXTERNAL PAYMENT SYSTEM**

Nataša Ščekić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj –Ovaj rad istražuje razvoj veb-aplikacije koja poboljšava korisničko iskustvo kroz vizualizaciju podataka, društvenu interakciju i integraciju e-plaćanja. Fokus je na funkcionalnostima koje omogućavaju laku navigaciju i korišćenje različitih sekcija aplikacije, uključujući i obradu plaćanja.

Ključne reči: Veb-aplikacija, korisničko iskustvo, vizualizacija, društvena interakcija, e-plaćanje

Abstract – This paper explores the development of a web application that enhances user experience through data visualization, social interaction, and e-payment integration. The focus is on functionalities that allow easy navigation and use of different sections of the application, including payment processing.

Keywords: Web application, user experience, visualization, social interaction, e-payment

1. UVOD

Aplikacija razvijena u okviru ovog rada predstavlja inovativnu platformu koja omogućava korisnicima da se povežu, komuniciraju i izraze kroz virtualne prostore. Ova platforma omogućava jednostavnu registraciju i prijavu, čime se svaki korisnik može brzo uključiti u zajednicu. Nakon kreiranja profila, korisnici imaju mogućnost da ažuriraju svoje informacije i vizualni identitet, obogaćujući svoje iskustvo unutar aplikacije, uz dodatne opcije za personalizaciju koje doprinose jedinstvenom doživljaju svakog korisnika.

Pored toga, platforma nudi funkcionalnosti za pretragu i interakciju među korisnicima, što podstiče dublju socijalnu povezanost. Korisnici mogu ostavljati komentare na profilima drugih članova, čime se stvara aktivna i dinamična komunikacija unutar zajednice. Kako bi unapredili svoje iskustvo, korisnici imaju pristup virtualnoj prodavnici gde mogu kupovati pakete novčića i razne avatare, dodatno personalizujući svoje prisustvo u aplikaciji. Kroz sveobuhvatne funkcionalnosti, aplikacija pruža korisnicima mogućnost da istražuju, komuniciraju i učestvuju u zajednici, čime se stvara jedinstveno okruženje za interakciju i kreativno izražavanje.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Željko Vuković, docent

2. PREGLED KORIŠĆENIH SOFTVERSKIH TEHNOLOGIJA

Aplikacija je razvijena koristeći savremene alate i platforme kako bi se obezbedilo visokokvalitetno korisničko iskustvo i efikasno upravljanje podacima. Tehnologije korišćene za frontend razvoj uključuju React.js [1] i Vercel [2], dok su za backend deo aplikacije korišćeni Node.js [3], Express.js [4], MongoDB [5], i Render [6]. Svaka od ovih tehnologija doprinosi različitim aspektima funkcionisanja aplikacije, od izrade korisničkog interfejsa do upravljanja podacima i hostovanja.

2.1. Frontend tehnologije

Za izradu korisničkog interfejsa aplikacije korišćen je React.js, popularna otvorena frontend JavaScript biblioteka koja se koristi za izgradnju korisničkih interfejsa zasnovanih na komponentama. React.js je posebno pogodan za razvoj single-page aplikacija, kao i aplikacija koje se renderuju na serveru pomoću nekog framework-a. React.js se temelji na deklarativnoj programskoj paradigmi, gde programeri definišu prikaze za svako stanje aplikacije, a React ažurira i renderuje komponente kada dođe do promena u podacima. Korišćenje React.js omogućava efikasno kreiranje dinamičkih i interaktivnih veb-stranica uz ponovnu upotrebu koda i brzo ažuriranje korisničkog interfejsa bez potrebe za ponovnim učitavanjem stranica. Ovaj pristup značajno poboljšava performanse aplikacije i pruža korisnicima glatko i responsivno iskustvo.

Node.js library je skup biblioteka i modula koji se koriste za rad sa Node.js platformom. Iako nije direktno korišćena u frontend delu aplikacije, ova biblioteka omogućava efikasno upravljanje asinhronim operacijama, rad sa datotekama i mrežnim resursima. Node.js library je koristan za integraciju sa alatima i bibliotekama koje mogu poboljšati frontend razvoj, kao što su alati za build i testiranje aplikacija.

Vercel je platforma za hostovanje i implementaciju frontend aplikacija koja omogućava brzo i efikasno postavljanje veb-stranica. Korišćena je za hostovanje frontend dela aplikacije koji je razvijen u React.js tehnologiji. Vercel omogućava automatske i kontinuirane izgradnje i implementacije sa GitHub-a i GitLab-a, čime se olakšava proces implementacije i upravljanja verzijama. Platforma takođe pruža optimizaciju performansi i brzinu učitavanja stranica, što doprinosi kvalitetu korisničkog iskustva.

2.2. Backend tehnologije

Node.js je korišćena server-side JavaScript platforma koja omogućava razvoj visokoperformantnih aplikacija. Temeljena je na V8 JavaScript engine-u, koristi event-driven, non-blocking I/O (Input/Output) model koji omogućava upravljanje velikim brojem istovremenih konekcija uz minimalne resurse. Ovaj model omogućava brzu obradu zahteva i efikasno upravljanje resursima, što je posebno korisno za razvoj real-time aplikacija. Node.js je korišćen za izgradnju server-side logike aplikacije, omogućavajući efikasnu obradu korisničkih zahteva i interakciju sa bazom podataka.

Express.js je fleksibilan web framework za Node.js koji olakšava razvoj server-side aplikacija i API-ja. Pruža bogat set funkcionalnosti za upravljanje HTTP zahtevima, definisanje ruta i middleware. Express.js omogućava jednostavno postavljanje servera, upravljanje sesijama, autentifikaciju korisnika i rad sa različitim vrstama zahteva. Korišćen je u aplikaciji za izgradnju server-side logike, upravljanje podacima i pružanje API-ja koji omogućava komunikaciju između frontend i backend dela aplikacije.

MongoDB je dokument-orijentisana NoSQL (Not Only Structured Query Language) baza podataka koja koristi JSON dokumente (ili slične njima) za skladištenje podataka. Kao NoSQL rešenje, MongoDB omogućava fleksibilno modelovanje podataka i lako skaliranje u skladu sa potrebama aplikacije. Ovaj pristup omogućava rad sa kompleksnim strukturama podataka i brzo pretraživanje. MongoDB je korišćen za čuvanje i upravljanje podacima aplikacije, pružajući efikasnu platformu za rad sa velikim količinama podataka.

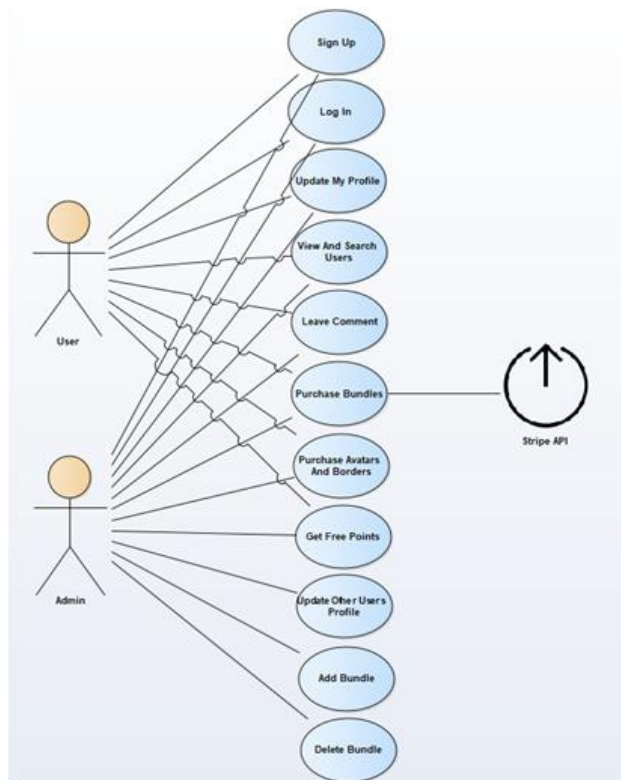
Render je sveobuhvatna cloud platforma koja omogućava hostovanje server-side aplikacija, baza podataka i drugih resursa na jednom mestu. Korišćen je za hostovanje backend dela aplikacije razvijenog u Node.js i Express.js. Render pruža jednostavno upravljanje resursima, automatske deploy-eve i skalabilnost za server-side aplikacije. Platforma omogućava jednostavnu konfiguraciju i upravljanje aplikacijama, čime se osigurava stabilan i efikasan rad aplikacije.

3. SPECIFIKACIJA ZAHTEVA

3.1. Funkcionalni zahtevi

Funkcionalni zahtevi sistema (Slika 1) obuhvataju osnovne i napredne funkcionalnosti koje omogućavaju korisnicima i administratorima efikasno korišćenje aplikacije. Predstavljaju se dijagramom slučajeva korišćenja [7]. U osnovne funkcionalnosti spadaju registracija korisnika, prijava, ažuriranje profila, pregled i pretraga korisničkih profila, ostavljanje komentara, kao i pregled i kupovina virtualnih dobara, kao što su novčići, avatari i bordure. Korisnici takođe mogu dobiti besplatne novčiće kroz razne aktivnosti, kao što su dnevni logini.

Na administrativnom nivou, administrator ima dodatna ovlašćenja za upravljanje korisnicima i paketima novčića, što uključuje nadgledanje aktivnosti korisnika i kreiranje promocija. Ove napredne funkcionalnosti pomažu u održavanju ravnoteže u sistemu i omogućavaju prilagođavanje potrebama korisnika, čime se poboljšava ukupno iskustvo na platformi.



Slika 1. Dijagram slučajeva korišćenja

3.2. Nefunkcionalni zahtevi

U ovoj veb-aplikaciji zadovoljavaju se značajni nefunkcionalni zahtevi, čime se osigurava kvalitetno korisničko iskustvo i efikasnost sistema.

Performanse sistema optimizovane su za brzu obradu zahteva korisnika. Testiranja pokazuju da aplikacija obrađuje zahteve za učitavanje profila i komentare u vrlo kratkom roku, čak i pod povećanim opterećenjem.

Korisničko iskustvo postavljeno je kao prioritet. Intuitivan dizajn interfejsa omogućava lako kretanje kroz funkcije aplikacije. Uputstva i obaveštenja dostupna su korisnicima tokom svih aktivnosti, čime se smanjuje potreba za tehničkom podrškom.

Korisnički interfejs estetski je privlačan i prilagođen različitim uređajima. Aplikacija poseduje responzivan dizajn, što znači da se prilagođava veličini ekrana, obezbeđujući optimalno korisničko iskustvo na računarima, tabletima i mobilnim uređajima.

Prenosivost sistema obezbeđena je, što znači da aplikacija funkcioniše na različitim operativnim sistemima i veb pregledačima, omogućavajući korisnicima da bez problema pristupaju aplikaciji bez obzira na platformu.

Bezbednost sistema je na visokom nivou. Integracija Stripe API-ja omogućava bezbedne virtualne finansijske transakcije, dok sistem koristi kriptovanje podataka i mehanizme za zaštitu korisničkih naloga.

Sistem je projektovan da bude otporan na greške, uz mehanizme za otkrivanje i izveštavanje o greškama. Ova funkcionalnost osigurava da sistem može nastaviti sa radom u slučaju neočekivanih problema, što je ključno za održavanje poverenja korisnika.

Sve ove karakteristike i funkcionalnosti doprinose stvaranju kvalitetne i bezbedne veb-aplikacije spremne da zadovolji potrebe korisnika i u budućnosti.

4. SPECIFIKACIJA DIZAJNA

U ovom poglavlju opisane su šeme podataka i klasni dijagram [8], što ima za cilj da pruži detaljan uvid u arhitektonske aspekte aplikacije, naglašavajući ključne elemente koji čine osnovu funkcionalnosti i efikasnosti razvijenog rešenja.

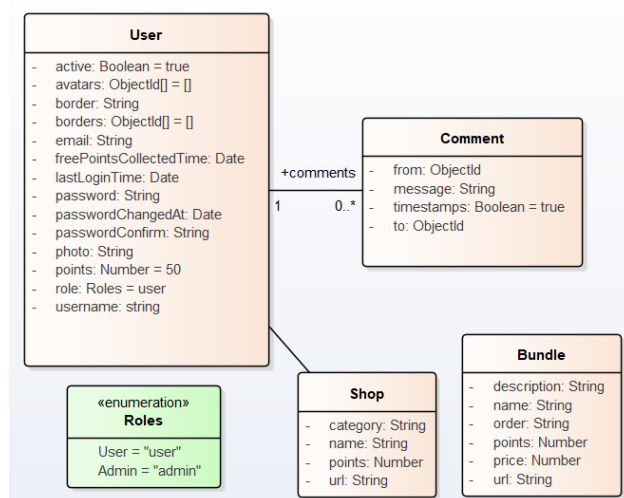
4.1. Šeme podataka

Aplikacija koristi MongoDB bazu podataka za upravljanje podacima, a definisane su četiri glavne šeme koje predstavljaju osnovne entitete – User (korisnik), Shop (prodavnica), Comment (komentar) i Bundle (paket). Šeme podataka predstavljaju logičku strukturu informacija koje se koriste u aplikaciji. Ove šeme omogućavaju strukturirano upravljanje korisničkim nalogima, interakcijama i kupovinom virtualnih dobara unutar aplikacije.

4.2. Klasni dijagram

Klasni dijagram predstavlja strukturu sistema u smislu objekata i njihovih interakcija. U okviru ovog sistema ilustruje klasu, atribut i metode, kao i odnose između klasa, pružajući detaljan uvid u logičku organizaciju aplikacije (Slika 2).

Glavne komponente klasnog dijagrama uključuju klase User (Korisnik), Shop (Prodavnica), Comment (Komentar) i Bundle (Paket novčića), zajedno sa njihovim atributima i vezama. Svaka klasa u dijagramu definiše određeni entitet u sistemu.



Slika 2. Klasni dijagram sistema

5. DEMONSTRACIJA SISTEMA

U ovom poglavlju je predstavljena demonstracija svih osnovnih funkcionalnosti aplikacije kroz ključne segmente koji su predstavljeni na stranici za prijavu korisnika, registraciju korisnika, početnoj stranici, stranici regularnog korisnika i stranici ulogovanog korisnika.

Svaka stranica aplikacije je u potpunosti responzivna, što omogućava prilagođavanje izgleda i funkcionalnosti različitim uređajima – od desktopa do mobilnih telefona. Bez obzira na veličinu ekrana, korisnici mogu lako i intuitivno pristupiti aplikaciji, uz očuvanu vidljivost i funkcionalnost elemenata, čime se obezbeđuje šira pristupačnost.

Logo aplikacije (Slika 3), koji je istaknut u dizajnu, doprinosi prepoznatljivosti brenda. Ova funkcionalnost doprinosi estetskoj privlačnosti stranice i čini korisnički interfejs vizuelno interesantnijim.



Slika 3. Logo aplikacije

5.1. Prijava na aplikaciju

Prijava na aplikaciju predstavlja prvi korak u korišćenju aplikacije i omogućava korisnicima siguran i jednostavan način pristupa njihovim personalizovanim podacima. Ovaj proces obuhvata unos korisničkih podataka koji se proveravaju radi autentifikacije i daljeg pristupa funkcionalnostima aplikacije.

Obuhvaćena je validacija unosa na stranici za prijavu – omogućena je trenutna povratna informacija o greškama u unosu, kao što su nepravilno unesena adresa e-pošte ili lozinka. Nakon prijavljivanja korisnika, prikazuje se poruka o uspešnosti prijave.

Na stranici za prijavu se nalazi link koji omogućava prelazak na stranicu za registraciju, pružajući korisnicima jednostavan način za kreiranje novog naloga ukoliko još uvek nemaju jedan.

5.2. Registracija korisnika

Stranica za registraciju sadrži formu sa nekoliko obaveznih polja, uključujući ime, prezime, adresu e-pošte, lozinku i potvrdu lozinke.

Validacija unosa podataka predstavlja ključni aspekt procesa registracije. Odmah se proveravaju uneti podaci i korisnicima se pruža povratna informacija o eventualnim greškama, kao što su neispravan format adrese e-pošte ili lozinka koja ne ispunjava sigurnosne kriterijume. Na ovaj način, korisnici mogu odmah da isprave svoje unose bez potrebe za slanjem podataka na server. Osigurano je da se samo ispravni i sigurni podaci unose u sistem, čime se smanjuje rizik od neovlašćenog pristupa ili pokušaja zlonamernih aktivnosti.

5.3. Početna stranica

Pregled početne stranice sadrži sve ključne funkcionalnosti i vizualne elemente koji čine osnovu korisničkog iskustva u aplikaciji.

Korisnicima je omogućen prikaz svih postojećih korisnika u aplikaciji. Ova funkcionalnost obrađuje zahtev za pristup bazi podataka i povlači sve korisničke naloge, prikazujući ih u vidu liste ili tabele u korisničkom interfejsu. Ovim postupkom omogućava se lako i brzo pregledavanje svih registrovanih korisnika, čime se poboljšava efikasnost upravljanja korisničkim podacima u aplikaciji.

U zaglavlju aplikacije nalaze se logo aplikacije koji vodi na početnu stranu i korisničko ime trenutno prijavljenog korisnika. Avatar korisnika otvara padajući meni sa opcijama kao što su profil, puni ekran, promena teme i odjava. Prikazuje se i broj novčića koje korisnik poseduje, a klikom na njih se otvara prodavnica podeljena u tri sekcije: Coin Shop, Border Shop i Avatar Shop.

U Coin Shop sekciji, korisnik može kupiti dodatne novčiće putem Stripe-a. Stripe trenutno funkcionira u testnom okruženju, što omogućava korisnicima da koriste test karticu za sve kupovine. Ova funkcionalnost omogućava korisnicima da upravljaju svojim profilima i unaprede ih pomoću virtualne valute.

U Avatar i Border Shop sekciji, prikazuju se avatari i bordure koje korisnik može kupiti i koristiti za sliku profila. Ukoliko korisnik pokuša da kupi avatar ili borduru koju već poseduje, sistem će ga o tome obavestiti kroz jasno prikazano upozorenje.

Korisnicima je omogućeno da dobiju besplatne novčiće na svaka 3 sata, u nasumičnom iznosu između 150 i 350.

Polje za pretragu omogućava korisnicima da pretražuju druge korisnike na osnovu korisničkog imena. Integrirano je sa sistemom validacije koji osigurava da su uneti podaci pravilno obrađeni. Istorija pretrage se automatski ažurira sa svakim uspešnim pretraživanjem i omogućava korisnicima da lako pristupe prethodnim pretragama, kao i da uklone stavke iz istorije pretrage.

5.4. Profil korisnika

Stranica profila korisnika pruža sveobuhvatan pregled ključnih informacija o određenom korisniku. Na stranici su prikazani komentari koje su ostavili drugi korisnici, omogućavajući uvid u interakcije korisnika sa zajednicom. Svaki komentar prikazuje informacije o korisniku koji ga je ostavio, uključujući vreme objavljivanja.

Administratori aplikacije imaju proširene funkcionalnosti na stranici profila korisnika. Pored mogućnosti izmene korisničkog imena i adrese e-pošte, administratori mogu menjati broj novčića koje korisnici poseduju, resetovati tajmer za besplatne novčiće za bilo kog korisnika u sistemu, kao i deaktivirati korisnički nalog.

5.5. Profil ulogovanog korisnika

Stranica profila ulogovanog korisnika služi kao centralno mesto za upravljanje ličnim podacima i vizuelnu personalizaciju unutar aplikacije, omogućavajući korisnicima potpunu kontrolu nad svojim profilom. Slična je stranici profila bilo kog korisnika, ali sadrži dodatne funkcionalnosti za upravljanje sopstvenim podacima.

Svaki korisnik može izmeniti svoje korisničko ime i adresu e-pošte. Pored toga, ulogovani korisnik može pristupiti kolekciji svojih kupljenih avatara i bordura i prilagoditi izgled svog profila primenom istih.

Administratori aplikacije imaju mogućnost upravljanja paketima novčića u okviru sistema kroz dve ključne funkcionalnosti – dodavanje i brisanje paketa novčića. Na ovaj način mogu aktivno upravljati paketima novčića u sistemu, prilagođavajući i optimizujući prodajnu ponudu u skladu sa potrebama korisnika.

6. ZAKLJUČAK

U ovom radu je predstavljeno softversko rešenje kojim se razvija veb-aplikacija za društvenu interakciju korisnika. U funkcionalnosti sistema spadaju kreiranje i ažuriranje profila, pregled i pretragu drugih korisnika, postavljanje komentara, kupovinu paketa novčića i drugih virtualnih dobara (avatara i bordura), kao i dobijanje besplatnih novčića u određenim vremenskim intervalima.

Implementacija je realizovana uz korišćenje Stripe API-ja, što omogućava sigurne virtualne finansijske transakcije. Trenutno sistem funkcionira u testnom režimu, bez uključivanja pravog novca, što pruža bezbedno okruženje za korisnike tokom kupovine digitalnih sadržaja. U razvoju aplikacije korišćene su moderne tehnologije i okviri, kako bi se obezbedila efikasnost, skalabilnost i bezbednost sistema.

Prednosti rešenja uključuju intuitivno korisničko iskustvo, sigurne finansijske transakcije, modularan dizajn i fleksibilnost u upravljanju korisnicima. Posebna pažnja je posvećena korisničkom interfejsu, što olakšava navigaciju i korišćenje aplikacije, čime se povećava zadovoljstvo korisnika. Integracija Stripe API-ja garantuje bezbednost prilikom virtualnih kupovina, što je ključno za stvaranje poverenja kod korisnika. Ova struktura omogućava laku nadogradnju i implementaciju novih funkcionalnosti u skladu sa potrebama korisnika i tržišta.

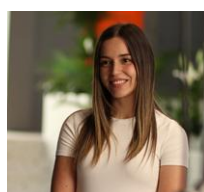
Međutim, rešenje ima i neke nedostatke, poput pravih finansijskih transakcija, ograničenih društvenih interakcija i potrebe za naprednijim sistemima obaveštavanja. Trenutno funkcionisanje u testnom režimu ograničava pravu primenu aplikacije, što može uticati na korisničko iskustvo i angažovanje.

U budućnosti se mogu otkloniti mane rešenja kroz razvoj pravog okruženja za finansijske transakcije, unapređenje funkcionalnosti foruma dodavanjem novih društvenih mehanizama kao što su rangiranje korisnika, sistemi za ocenjivanje sadržaja i napredne notifikacije, kao i integraciju više sistema plaćanja. Ove promene će doprineti boljem korisničkom iskustvu i monetizaciji aplikacije, čime će se stvoriti solidna osnova za dalji rast i unapređenja.

7. LITERATURA

- [1] S. M. Smith, React.js Development: Mastering Modern Frontend, 1st edition, 2022
- [2] J. L. Harris, Building Modern Web Applications with Vercel and React, 1st edition, 2023
- [3] T. A. Brown, Mastering Node.js: Expert Techniques for Building Scalable Network Applications, 2nd edition, 2022
- [4] J. W. Kelly, Express.js Guide: The Comprehensive Book on Express.js Framework, 1st edition, 2021
- [5] MongoDB, MongoDB Documentation, <https://www.mongodb.com/docs/>
- [6] R. P. Turner, Deploying Applications with Render and MongoDB Atlas, 2023
- [7] Lucidchart, UML Use Case Diagram Tutorial, <https://www.lucidchart.com/pages/uml-use-case-diagram/>
- [8] Lucidchart, Class Diagram Tutorial, <https://www.lucidchart.com/pages/uml-class-diagram/>

Kratka biografija:



Nataša Ščekić rođena je u Novom Sadu 1999. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Elektronsko poslovanje odbranila je 2024.godine.
kontakt: scekicnataša@gmail.com