

# Razvoj web platforme za analizu emocionalnih dnevnika korisnika

---

**Nikolić, Marija**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zadar / Sveučilište u Zadru**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:162:169079>

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

*Download date / Datum preuzimanja:* **2024-11-15**



**Sveučilište u Zadru**  
Universitas Studiorum  
Jadertina | 1396 | 2002 |

*Repository / Repozitorij:*

[University of Zadar Institutional Repository](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJ

# Sveučilište u Zadru

Odjel za informacijske znanosti  
Stručni prijediplomski studij  
Informacijske tehnologije



Zadar, 2024.

Sveučilište u Zadru

Odjel za informacijske znanosti  
Stručni prijediplomski studij  
Informacijske tehnologije

RAZVOJ WEB PLATFORME ZA ANALIZU EMOCIONALNIH DNEVNIKA  
KORISNIKA

Završni rad

Student/ica:

Marija Nikolić

Mentor/ica:

dr. sc. Ljiljana Zekanović-Korona

Komentor/ica:

mag. ing. El. Niko Vrdoljak

Zadar, 2024.



## **Izjava o akademskoj čestitosti**

Ja, Marija Nikolić, ovime izjavljujem da je moj završni rad pod naslovom Razvoj web platforme za analizu emocionalnih dnevnika korisnika. rezultat mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na izvore i radove navedene u bilješkama i popisu literature. Ni jedan dio mojega rada nije napisan na nedopušten način, odnosno nije prepisan iz necitiranih radova i ne krši bilo čija autorska prava.

Izjavljujem da ni jedan dio ovoga rada nije iskorišten u kojem drugom radu pri bilo kojoj drugoj visokoškolskoj, znanstvenoj, obrazovnoj ili inoj ustanovi.

Sadržaj mojega rada u potpunosti odgovara sadržaju obranjenoga i nakon obrane uređenoga rada.

**Zadar**, 1. rujna 2024.

## Sažetak

Živimo u doba ubrzanog razvoja umjetne inteligencije i njezine integracije u različite aspekte društva, koja značajno doprinosi unapređenju kvalitete života. Motivirani istraživanjem potencijala umjetne inteligencije za praćenje emocionalnog stanja korisnika, razvili smo full-stack web aplikaciju za analizu emocionalnih dnevnika uz pomoć Watson NLP-a. Aplikacija koristi moderne frontend i backend tehnologije za osiguravanje intuitivnog korisničkog sučelja i robusne obrade podataka. Integracija IBM Watson NLP-a omogućuje analizu tekstualnih unosa korisnika i pruža povratne informacije o emocionalnom stanju. Ovaj rad donosi pregled svih faza razvoja aplikacije: uvid u tehničke aspekte, evaluaciju konačnog proizvoda te predlaže daljnje smjerove razvoja.

Ključne riječi: Full-stack aplikacija, emocionalni dnevnik, IBM Watson procesor prirodnog jezika, analiza emocionalnog stanja, mentalno zdravlje, HTML, CSS, Angular, Django REST okvir, SQLite, prirodni jezik, umjetna inteligencija

## Sadržaj

1. Uvod.....	1
2. Ciljevi i svrha rada .....	3
2.1. Ciljevi istraživanja.....	3
2.2. Svrha istraživanja .....	4
3. Opis problema .....	6
4. Teorijski okvir uz problematiku .....	8
5. Razrada teme – rješenje problema.....	11
5.1. Arhitektura aplikacije .....	11
5.1.1. Svrha i ciljevi arhitekture .....	11
5.1.2. Frontend arhitektura .....	12
5.1.3. Backend arhitektura.....	17
5.1.4. Komunikacija između frontenda i backenda .....	19
5.2. Sigurnosni aspekti .....	20
5.3. Backend implementacija .....	21
5.3.1. Struktura backend sustava .....	22
5.3.2. Detalji o API-ju .....	24
5.3.3. Rad s bazom podataka.....	25
5.3.4. Upravljanje performansama i skalabilnost .....	25
5.3.5. Izazovi i naučene lekcije .....	27
5.4. Frontend implementacija.....	28
5.4.1. Struktura aplikacije.....	28
5.4.2. Dizajn korisničkog sučelja .....	29
5.4.3. Komunikacija s backendom .....	30
5.4.4. Izazovi i naučene lekcije .....	31
5.4.5. Rukovanje pogreškama .....	31
5.4.6. Asinkrona komunikacija i responzivnost .....	32
5.4.7. Sigurnosni aspekti komunikacije.....	32
5.5. Arhitektura i funkcionalnosti integracije.....	34
5.5.1. Upravljanje izazovima integracije.....	35
5.5.2. Optimizacija i performanse .....	36
6. Zaključak .....	37
Izvori: .....	39
Popis slika .....	41
Poveznica na GitHub repozitorij: .....	42

## 1. Uvod

U suvremenom društvu, obilježenom brzim tempom života i često prisutnim stresnim situacijama održavanje mentalnog zdravlja postaje prioritetno pitanje. Sve veća svijest o važnosti mentalnog zdravlja potaknula je razvoj različitih tehnoloških rješenja koja imaju za cilj pomoći pojedincima u praćenju, razumijevanju i reguliranju njihovog emocionalnog stanja. Jedno od takvih inovativnih rješenja je aplikacija "Dnevnik analize emocija", koja koristi napredne tehnike umjetne inteligencije, a osobito tehnologiju za obradu prirodnog jezika pod nazivom *Watson Natural Language Processing* (Watsonova obrada prirodnog jezika, u daljnjem tekstu IBM Watson NLP) koju je razvila tvrtka International Business Machines (IBM). Ova tehnologija omogućuje analizu emocionalnog stanja korisnika na temelju njihovih tekstualnih unosa, pružajući im tako uvid u vlastite emocionalne obrasce.

Cilj ovog završnog rada je detaljno prikazati proces razvoja cjelovite web aplikacije pod nazivom *Dnevnik analize emocija*. Aplikacija je osmišljena kako bi korisnicima omogućila praćenje i razumijevanje vlastitog emocionalnog stanja kroz korištenje umjetne inteligencije, te kao podrška u poboljšanju mentalnog zdravlja i opće dobrobiti. U radu će se obraditi teorijski okvir koji se odnosi na psihološke aspekte emocija i njihovu interpretaciju putem tehnologija obrade prirodnog jezika, kao i detaljan opis tehničke implementacije navedene aplikacije.

Korisničko sučelje aplikacije, to jest *frontend* dio, razvijeno je korištenjem standardnih web tehnologija poput jezika za izradu web stranica, *HyperText Markup Language* (HTML) i kaskadnih stilova (CSS), uz korištenje programskog okvira *Angular* (u daljnjem tekstu: Angular), čime se postiže interaktivno i intuitivno korisničko iskustvo. Pozadinski dio aplikacije (ili *backend*) je implementiran pomoću okvira za razvoj aplikacija *Django REST Framework* (u daljnjem tekstu: Django REST Framework), koji omogućuje izgradnju robusnih i skalabilnih aplikacijskih programskih sučelja (API) za obradu i pohranu podataka. Podaci se pohranjuju u lokalnu bazu podataka SQLite, a što osigurava jednostavno upravljanje informacijama i brz pristup pohranjenim podacima.

Integracija tehnologije *Watson Natural Language Processing* unutar pozadinskog sustava omogućuje analizu tekstualnih unosa korisnika i povratni rezultat analize njihovog emocionalnog stanja. Aplikacija pruža korisnicima neposrednost informacija o njihovom emocionalnom stanju te omogućuje praćenje promjena o istom kroz duže vremenske periode, poput tjedana ili mjeseci. Time se korisnicima pruža podrška u autorefleksiji, osvještavanju emocionalnih obrazaca i potencijalnom poboljšanju njihovog mentalnog zdravlja.

Ovaj rad će obuhvatiti cjelokupni proces razvoja aplikacije *Dnevnik analize emocija*, počevši od faze planiranja i dizajna, preko tehničke implementacije i integracije sustava do završne evaluacije funkcionalnosti i korisničkog iskustva. Detaljnom analizom svih ključnih komponenti i izazova s kojima smo se susreli tijekom razvoja cilj je pokazati na koji način napredne tehnologije, kao što je umjetna inteligencija, mogu biti primijenjene za poboljšanje mentalnog zdravlja i općeg blagostanja pojedinca.



## 2. Ciljevi i svrha rada

### 2.1. Ciljevi istraživanja

Glavni cilj ovog istraživanja je ispitati proces razvoja aplikacije *Dnevnik analize emocija*, koja koristi napredne tehnologije umjetne inteligencije za analizu i interpretaciju ljudskih emocija na temelju tekstualnih unosa korisnika. Primarni fokus istraživanja je na tehničkoj izradi aplikacije, s naglaskom na korištenje Djanga za backend, Angulara za frontend te IBM Watson NLP-a za analizu prirodnog jezika. Cilj je istražiti kako se ovi alati i tehnologije mogu integrirati kako bi se stvorio funkcionalan i pouzdan sustav koji omogućuje korisnicima praćenje vlastitih emocionalnih stanja.

Kroz razvoj aplikacije *Dnevnik analize emocija* istraživanje će se usmjeriti na tehničke aspekte izrade, uključujući projektiranje arhitekture aplikacije, integraciju API-ja, razvoj korisničkog sučelja te optimizaciju performansi i responzivnosti sustava. Analizirat će se izazovi povezani s razvojem modularne i skalabilne aplikacije koja koristi *SOLID* principe, s posebnim naglaskom na primjenu *principa jedinstvene odgovornosti* (eng. Single Responsibility Principle) kako bi se osigurala jasnoća i održivost koda. Razmatranjem ovih aspekata omogućit će bolje razumijevanje suradnje različitih komponenata unutar složenih aplikacijskih sustava.

Jedan od ključnih ciljeva istraživanja je i procjena izvedbe aplikacije u realnim uvjetima, uključujući ispitivanje točnosti i pouzdanosti NLP tehnologije u analizi emocionalnih obrazaca iz tekstualnih unosa. Istražit će se kako različiti oblici teksta – od kratkih bilješki do složenijih narativa – utječu na kvalitetu analize te kako aplikacija može dosljedno pružiti korisnicima relevantne povratne informacije. Ovaj dio istraživanja obuhvaća i evaluaciju integracije NLP servisa s backend sustavom, te optimizaciju procesa obrade podataka kako bi se osigurala brza i efikasna analiza emocija.

Daljnji cilj istraživanja uključuje i tehničku evaluaciju implementiranih rješenja u vezi s bazama podataka, rukovanjem podacima i sigurnošću aplikacije. Koristi se SQLite baza podataka, koja omogućuje jednostavno upravljanje i pohranu podataka na lokalnoj razini, što je u skladu s prototipnom prirodom aplikacije. Istražit će se učinkovitost ovakvog rješenja te mogućnosti njegovog proširenja ili zamjene u slučaju daljnjeg razvoja aplikacije na višoj razini. Također, analiza će uključiti procjenu sigurnosnih mjera, iako su one minimalne zbog lokalne uporabe aplikacije, razmatrat će se potencijalne nadogradnje sigurnosti poput autentifikacije i autorizacije korisnika u budućim verzijama aplikacije.

Osim toga, rad istražuje dizajnerske odluke povezane s korisničkim sučeljem aplikacije. Naglasak je stavljen na jednostavnost i intuitivnost sučelja: korištenje neutralnih boja kako bi se izbjegao utjecaj na emocije korisnika te implementaciju vizualizacija koje omogućuju lako praćenje i razumijevanje emocionalnih obrazaca. Evaluirat će se učinkovitost dizajna sučelja u kontekstu korisničkog iskustva i mogućnosti daljnjeg unapređenja.

Istraživanje također obuhvaća procjenu fleksibilnosti i moguće proširivosti aplikacije. Razmatrat će se kako arhitektura aplikacije i primijenjeni razvojni principi omogućuju lako proširivanje funkcionalnosti bez narušavanja postojećeg sustava. To uključuje proučavanje mogućnosti dodavanja funkcionalnosti: poput naprednih sigurnosnih značajki, podrške za više korisnika ili integracije dodatnih analitičkih alata bez potrebe za značajnim preinakama postojećeg koda.

Konačno, ovaj rad teži pružiti sveobuhvatan pregled svih faza razvoja aplikacije *Dnevnik analize emocija*; uključujući planiranje, dizajn, implementaciju, testiranje i evaluaciju. Cilj je ponuditi detaljne uvide u praktične aspekte izgradnje aplikacije koja koristi umjetnu inteligenciju za analizu emocija, s naglaskom na tehničke izazove, rješenja i naučene lekcije. Ovaj rad ne samo da dokumentira razvoj aplikacije, već i nudi smjernice za buduće projekte koji se bave sličnim tehnološkim izazovima u području obrade prirodnog jezika i analize emocionalnih stanja.

## **2.2. Svrha istraživanja**

Svrha ovog istraživanja je dublje razumjeti kako se umjetna inteligencija može koristiti za praćenje i analizu ljudskih emocija te istražiti načine na koje tehnologije za obradu prirodnog jezika mogu biti učinkovito primijenjene u kontekstu mentalnog zdravlja. Kroz razvoj aplikacije *Dnevnik analize emocija*, istraživanje ima za cilj pružiti uvid u proces kreiranja inovativnog alata koji omogućuje korisnicima bolji uvid u njihova emocionalna stanja i potiče ih na autorefleksiju i osobni razvoj.

Aplikacija *Dnevnik analize emocija* može igrati važnu ulogu u unapređenju mentalnog zdravlja i psihoterapijskih praksi na nekoliko načina:

- **Unapređenje mentalnog zdravlja:** aplikacija može pružiti korisnicima kontinuirani uvid u njihova emocionalna stanja, omogućujući im da bolje razumiju vlastite emocionalne reakcije i obrasce. Ovaj proces može dovesti do ranog prepoznavanja emocionalnih poteškoća, poput znakova anksioznosti, depresije ili drugih poremećaja raspoloženja, te pružiti mogućnost za pravovremene intervencije.

- **Dostupnost i praktičnost:** za razliku od tradicionalnih metoda, kao što su pisani dnevnici ili periodične konzultacije s terapeutima, aplikacija omogućuje neposredno svakodnevno praćenje emocija. To znači da korisnici mogu kontinuirano bilježiti svoje emocionalne reakcije u svakodnevnom životu, uz minimalan napor, te dobiti povratne informacije koje su prilagođene njihovom trenutnom emocionalnom stanju.
- **Podrška za stručnjake:** aplikacija može poslužiti kao dodatni alat za terapeute, psihologe i druge stručnjake za mentalno zdravlje, pružajući im uvid u emocionalne obrasce i promjene klijenata između terapijskih sastanaka. Ovaj dodatni izvor informacija može pomoći stručnjacima da bolje razumiju potrebe i izazove svojih klijenata te optimiziraju terapijske intervencije.
- **Etička i sigurnosna pitanja:** razmatranje etičkih i sigurnosnih implikacija primjene umjetne inteligencije za analizu emocija je ključno – posebice u kontekstu zaštite privatnosti korisnika i osiguravanja pravedne upotrebe tehnologije. Istraživanje se bavi pitanjima transparentnosti i sigurnosti u prikupljanju, obradi i pohrani podataka te istražuje kako osigurati da aplikacija bude korištena na etički prihvatljiv način.

Ovo istraživanje pruža dublji uvid u izazove i mogućnosti primjene umjetne inteligencije u kontekstu mentalnog zdravlja i psihoterapije, te nastoji prikazati kako tehnologija može podržati emocionalni razvoj i poboljšanje kvalitete života korisnika. Razvoj aplikacije *Dnevnik analize emocija* predstavlja značajan korak prema integraciji digitalnih alata u svakodnevnu praksu podrške mentalnom zdravlju, uz istovremeno bavljenje ključnim tehničkim, etičkim i sigurnosnim pitanjima.

### 3. Opis problema

U suvremenom društvu, praćenje i upravljanje emocionalnim stanjima korisnika predstavlja značajan izazov – posebice u kontekstu mentalnog zdravlja. Trenutni alati i metode koji su dostupni korisnicima i stručnjacima za mentalno zdravlje, poput pisanih dnevnika, periodičnih konzultacija i tradicionalnih testova za procjenu raspoloženja često su ograničeni u svojoj mogućnosti pružanja kontinuiranog i preciznog uvida u emocionalne promjene [1][2].

Jedan od glavnih problema je nepostojanje integriranih, digitalnih alata koji su prilagođeni potrebama korisnika a koji mogu pružiti detaljnu analizu i praćenje emocionalnih stanja na jednostavan i pristupačan način. Pisani dnevnici, iako korisni, zahtijevaju disciplinu i redovito vođenje, što mnogi korisnici ne uspijevaju dosljedno održavati [3]. Osim toga, pisani dnevnik ne omogućuje detaljnu analizu emocionalnih obrazaca niti pruža povratne informacije neposredno, što može biti ključno za ranu intervenciju kod emocionalnih problema [4].

Psihoterapijski procesi također pate zbog nedostatka alata koji omogućuju stručnjacima kontinuirano praćenje emocionalnog stanja klijenata između terapijskih sastanaka. Ova praznina u alatu često rezultira time da terapeuti nemaju potpunu sliku o promjenama u emocionalnom stanju klijenta koje bi se mogle dogoditi između sastanaka, što otežava procjenu napretka i prilagodbu terapijskog plana [5].

S obzirom na sve veći naglasak na personaliziranom pristupu mentalnom zdravlju, postoji očita potreba za moderniziranim, tehnološki naprednim alatima koji mogu osigurati precizno i kontinuirano praćenje emocija. Umjetna inteligencija, posebno tehnologije obrade prirodnog jezika (NLP), nudi potencijal za rješavanje ovih izazova. Integracija NLP-a u digitalne alate omogućuje automatiziranu analizu tekstualnih unosa korisnika, što može pružiti dubok uvid u njihove emocionalne obrasce i omogućiti neposredno pružanje povratnih informacija [6].

Prepoznat je značajan nedostatak u dostupnosti digitalnih rješenja koja bi mogla pružiti podršku u psihoterapiji i svakodnevnom praćenju emocionalnih stanja. Iako postoje aplikacije koje se bave općim praćenjem zdravlja, one često ne zadovoljavaju specifične potrebe za detaljnom analizom emocionalnog stanja ili su nedovoljno sofisticirane za pružanje uvida u složene emocionalne procese [3]. Ovaj nedostatak predstavlja ključnu prepreku u osiguravanju cjelovitije podrške za mentalno zdravlje, što dodatno naglašava potrebu za razvojem naprednijih alata.

Aplikacija *Dnevnik analize emocija* razvijena je upravo s ciljem rješavanja tih izazova. Korištenjem naprednih tehnologija, poput IBM Watson NLP-a, ova aplikacija omogućuje korisnicima praćenje njihovih emocionalnih stanja na temelju svakodnevnih tekstualnih unosa. Time se pruža kontinuirani uvid u emocionalne promjene, što može značajno poboljšati korisničku samosvijest i pružiti terapijski alat za stručnjake, omogućujući im precizniju i pravovremenu intervenciju [7].

#### 4. Teorijski okvir uz problematiku

Razumijevanje emocionalnih stanja i njihovo praćenje temeljni su aspekti psihološke znanosti i psihoterapijske prakse. Teorijski okvir ovog istraživanja temelji se na nekoliko ključnih koncepata u području psihologije emocija i primjeni tehnologije u mentalnom zdravlju, koji zajedno pomažu u definiranju potreba za inovativnim rješenjima kao što je aplikacija Dnevnik analize emocija.

Jedna od osnovnih teorija koja podupire ovo istraživanje jest teorija emocionalne inteligencije. Daniel Goleman, jedan od vodećih istraživača na ovom području, definira emocionalnu inteligenciju kao sposobnost prepoznavanja, razumijevanja, kontroliranja i izražavanja emocija na način koji poboljšava međuljudske odnose i opće mentalno zdravlje [8]. Prema Golemanu, emocionalna inteligencija igra ključnu ulogu u osobnom i društvenom funkcioniranju, utječući na sposobnost pojedinaca da se nose sa stresom, grade zdrave odnose i donose racionalne odluke [8]. Povećana emocionalna svijest može pomoći u sprječavanju i liječenju mentalnih poremećaja poput anksioznosti i depresije, jer korisnicima omogućuje bolje razumijevanje vlastitih emocionalnih reakcija i obrazaca [9].

Teorija emocionalne inteligencije također je relevantna u kontekstu tehnoloških inovacija koje podržavaju emocionalno praćenje. Kroz korištenje tehnologija za obradu prirodnog jezika (NLP), moguće je analizirati tekstualne unose korisnika kako bi se prepoznali neposredni emocionalni obrasci i promjene [10]. Benton i suradnici (2017) ističu da se NLP može koristiti za razvijanje digitalnih alata koji mogu pružiti povratne informacije o emocionalnom stanju korisnika, što omogućuje personalizirane intervencije i podršku [3]. Ovi alati mogu pomoći korisnicima u prepoznavanju svojih emocionalnih obrazaca i pridonijeti njihovoj emocionalnoj regulaciji, što je ključno za održavanje dobrog mentalnog zdravlja.

Dodatno, Paul Ekman, pionir u istraživanju emocija, identificirao je pet osnovnih emocija – sreću, tugu, strah, gađenje i ljutnju – koje su univerzalno prepoznatljive kroz izraze lica, bez obzira na kulturološke razlike. Ekmanova teorija osnovnih emocija pruža važno polazište za razvoj NLP alata jer omogućuje prepoznavanje ovih emocionalnih stanja u tekstualnim i neverbalnim zapisima korisnika, što je ključno za personaliziranu podršku u aplikacijama za mentalno zdravlje. Njegova istraživanja čine temelj za razumijevanje kako se emocije mogu

automatski analizirati putem umjetne inteligencije, što je posebno relevantno u razvoju digitalnih terapijskih alata.

Korištenje NLP-a u aplikacijama za mentalno zdravlje podržava teoriju kognitivne bihevioralne terapije (CBT), koja naglašava važnost prepoznavanja i analize misli i osjećaja kao temeljne strategije u procesu liječenja [11]. CBT se temelji na ideji da se emocije i ponašanje mogu mijenjati kroz svjesnu analizu i reinterpretaciju negativnih obrazaca mišljenja [12]. Digitalni alati koji koriste umjetnu inteligenciju, poput aplikacije Dnevnik analize emocija, mogu omogućiti korisnicima neposredno praćenje njihovih emocionalnih stanja, pružajući dodatnu podršku i omogućujući terapijske intervencije izvan tradicionalnih okvira terapijskih sastanaka [13]. Istraživanje Fitzpatricka i suradnika (2017) pokazalo je da su automatizirani digitalni alati korisni u poboljšanju učinkovitosti terapijskih postupaka, pružajući stalnu podršku korisnicima i omogućujući bolje razumijevanje njihovih emocionalnih stanja [4].

Osim emocionalne inteligencije i CBT-a, ovaj rad također uzima u obzir teoriju adaptivnog učenja i automatiziranih sustava koji koriste povratne informacije za kontinuirano poboljšanje [6]. Prema ovom pristupu, aplikacija koja koristi umjetnu inteligenciju može učiti iz korisničkih unosa i prilagođavati svoje odgovore i preporuke na temelju ponašanja korisnika i povratnih informacija. Ovo je osobito važno u kontekstu mentalnog zdravlja, gdje se zahtijeva visok stupanj prilagodljivosti kako bi se osigurale učinkovite intervencije koje su prilagođene individualnim potrebama korisnika [13].

Teorijski okvir također uključuje razmatranja etičkih implikacija korištenja umjetne inteligencije za praćenje i analizu emocija. Upotreba NLP-a i drugih tehnologija umjetne inteligencije u mentalnom zdravlju nosi sa sobom potencijalne rizike vezane uz privatnost i sigurnost korisničkih podataka [14]. Prema Mittelstadtu i suradnicima (2016), etička pitanja poput transparentnosti u prikupljanju podataka, pristupa podacima i pohrane, te informiranog pristanka korisnika su od ključne važnosti za osiguravanje odgovorne upotrebe ovih tehnologija [15]. Zaštita privatnosti i podataka korisnika postaje kritična u kontekstu aplikacija koje bilježe i analiziraju osjetljive emocionalne informacije, što zahtijeva postavljanje visokih standarda za etičku primjenu tehnologije.

Nadalje, teorijski okvir istraživanja uključuje koncepte iz područja bihevioralne znanosti i tehnološke prihvatljivosti, koji su ključni za razvoj i implementaciju novih digitalnih alata u praksi [16]. Prema teoriji prihvatljivosti tehnologije, korisnici će usvojiti novu tehnologiju

samo ako je smatraju korisnom, jednostavnom za korištenje i sigurnom [17]. Ovo je posebno važno u kontekstu mentalnog zdravlja, gdje povjerenje korisnika u alat igra ključnu ulogu u njegovoj učinkovitosti i uspješnosti.

Kombinacija ovih teorija i koncepata pruža sveobuhvatan temelj za razvoj aplikacije Dnevnik analize emocija. Integracija emocionalne inteligencije, CBT-a, adaptivnog učenja, etičkih razmatranja i tehnološke prihvatljivosti osigurava da je aplikacija ne samo tehnološki napredna već i usklađena s potrebama korisnika i stručnjaka za mentalno zdravlje. Kroz ovaj teorijski okvir, istraživanje nastoji pokazati kako se napredne tehnologije mogu učinkovito koristiti za poboljšanje mentalnog zdravlja i emocionalnog blagostanja korisnika.



## 5. Razrada teme – rješenje problema

Razvoj aplikacije *Dnevnik analize emocija* predstavlja odgovor na brojne izazove u praćenju emocionalnih stanja korisnika, s naglaskom na poboljšanje mentalnog zdravlja. Ova aplikacija koristi naprednu frontend-backend arhitekturu koja omogućuje jednostavno, efikasno i kontinuirano praćenje emocija korisnika uz pomoć umjetne inteligencije, konkretno tehnologije obrade prirodnog jezika (NLP)[18]. Aplikacija je osmišljena s ciljem intuitivnosti za krajnje korisnike, dok istovremeno nudi sofisticirane analize koje mogu koristiti i stručnjaci za mentalno zdravlje.

### 5.1. Arhitektura aplikacije

Arhitektura aplikacije *Dnevnik analize emocija* osmišljena je s ciljem da osigura jednostavno i intuitivno korisničko iskustvo za praćenje emocionalnih stanja kroz primjenu umjetne inteligencije. Glavni pristup arhitekturi temelji se na podjeli sustava na frontend i backend komponente, čime se postiže razdvajanje odgovornosti i omogućuje lakše održavanje i potencijalno proširivanje aplikacije.

Ova podjela omogućuje frontend dijelu da se fokusira na interakciju s korisnikom i prezentaciju podataka dok backend dio obavlja teške zadatke obrade podataka, upravljanja bazom podataka i komunikacije s NLP servisima[19]. Kroz primjenu SOLID principa, s naglaskom na načelo jednostruke odgovornosti (Single Responsibility Principle), svaka komponenta sustava ima jasno definiranu ulogu, što značajno doprinosi čitljivosti koda, skalabilnosti i lakšem snalaženju tijekom razvoja i održavanja.

#### 5.1.1. Svrha i ciljevi arhitekture

Primarni cilj arhitekture aplikacije *Dnevnik analize emocija* bio je stvoriti sustav koji je jednostavan za korištenje, lako proširiv i prilagodljiv budućim zahtjevima. Razdvajanje na frontend i backend dio omogućilo je jasan pregled funkcionalnosti i pojednostavilo razvoj. Arhitektura je osmišljena tako da podržava osnovne funkcionalnosti potrebne za prototip aplikacije, s mogućnošću daljnjeg razvoja i proširenja prema potrebama.

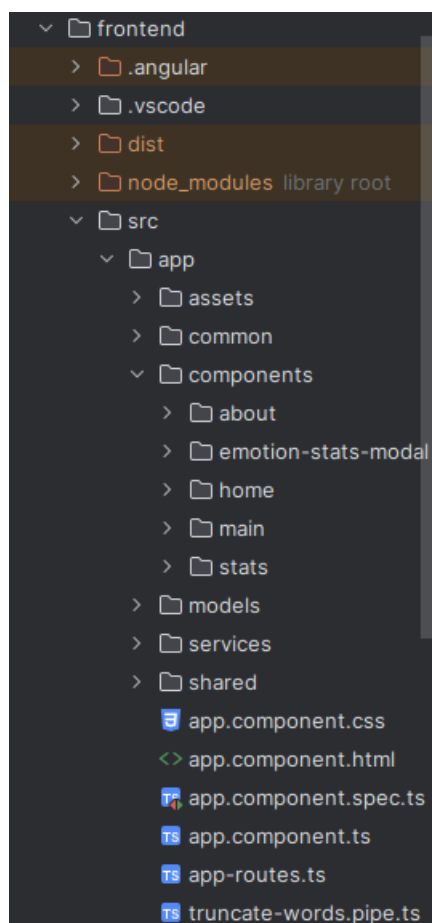
Razdvajanje na frontend i backend dijelove omogućuje modularni razvoj gdje svaki dio sustava može biti razvijan i održavan neovisno, bez utjecaja na drugi. Ovakva arhitektura omogućuje timu da se paralelno fokusira na različite aspekte aplikacije – frontend tim može raditi na poboljšanju korisničkog sučelja i korisničkog iskustva, dok backend tim može optimizirati performanse i upravljati složenim poslovnim logikama. Osim toga, primjena SOLID principa doprinosi održivosti koda. Načelo jednostruke odgovornosti (Single Responsibility Principle) nalaže da svaka klasa ili modul unutar sustava ima specifičnu ulogu, što smanjuje složenost i međuzavisnost koda, čineći ga lakšim za razumijevanje i održavanje.

Cilj arhitekture je također bio osigurati skalabilnost i fleksibilnost. Iako je aplikacija trenutno u fazi prototipa s osnovnim funkcionalnostima, arhitektura je postavljena tako da omogućuje dodavanje novih značajki bez potrebe za značajnim izmjenama postojećeg koda. Na primjer, jednostavno se mogu dodati nove API rute ili proširiti funkcionalnosti NLP analize bez da se pritom naruši osnovna struktura sustava.

### **5.1.2. Frontend arhitektura**

Frontend aplikacije *Dnevnik analize emocija* razvijen je koristeći Angular, moderni framework za razvoj web aplikacija koji omogućuje izradu složenih korisničkih sučelja s visokim stupnjem interaktivnosti i responzivnosti[20].

Angular je odabran zbog svoje modularnosti, bogatog ekosustava i jake podrške zajednice, što ga čini idealnim alatom za razvoj web aplikacija koje zahtijevaju visok stupanj fleksibilnosti i brzine. U aplikaciji *Dnevnik analize emocija* Angular upravlja cjelokupnim korisničkim sučeljem, od unosa podataka do prikaza rezultata emocionalnih analiza.

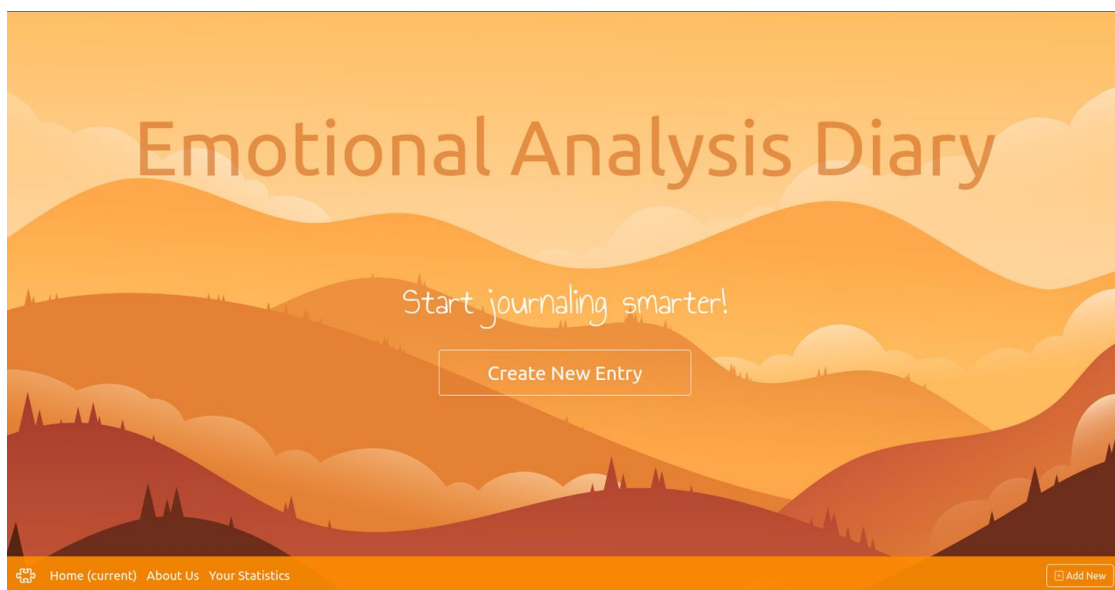


Slika 1 Vizualni prikaz strukture frontend dijela aplikacije

- **Komponente sučelja:** Angular arhitektura temelji se na komponentama koje predstavljaju osnovne gradivne blokove aplikacije. U *Dnevniku analize emocija*, glavne komponente uključuju *HomeComponent*, *MainComponent*, *AboutComponent*, i *StatsComponent*. Svaka od ovih komponenti ima specifičnu ulogu unutar aplikacije. Na primjer, *MainComponent* omogućuje unos korisničkih podataka, pružajući korisniku polje za unos tekstualnog sadržaja koji se šalje na analizu. Nakon što korisnik unese svoj dnevni unos, podaci se putem servisa šalju backendu, a rezultati analize prikazuju se neposredno. *StatsComponent* je s druge strane zadužen za vizualizaciju prikupljenih podataka o emocionalnim stanjima korisnika kroz različite grafičke prikaze, koristeći alate poput kružnih grafikona za prezentaciju udjela različitih emocija.
- **Upravljanje stanjima i komunikacija:** upravljanje stanjima u Angular aplikaciji omogućuje praćenje promjena unutar aplikacije i njihovo sinkroniziranje s korisničkim sučeljem. Angularov ugrađeni sustav za upravljanje stanjima omogućuje da svaka promjena

podataka automatski ažurira prikaz korisničkog sučelja, čime se osigurava da korisnik uvijek vidi najnovije informacije. Za komunikaciju s backendom koriste se HTTP zahtjevi putem Angularovog HttpClient servisa, omogućujući jednostavnu i efikasnu razmjenu podataka između frontenda i backenda.

- **Dizajn korisničkog sučelja:** dizajn sučelja u *Dnveniku analize emocija* naglašava intuitivnost i jednostavnost korištenja. Korisničko sučelje je dizajnirano s naglaskom na neutralnost, koristeći boje koje su odabrane kako bi se izbjeglo nepoželjno utjecanje na emocionalna stanja korisnika tijekom interakcije s aplikacijom. Osnovna filozofija dizajna je nenametljivost korisničkog sučelja čime se osigurava fokus korisnika na unos i analizu njegovih emocija bez dodatnih distrakcija. Sve komponente su postavljene tako da omogućavaju jednostavnu navigaciju i pristup ključnim funkcionalnostima aplikacije, poput unosa novih dnevnika, pregledavanja statistika i analize emocionalnih stanja.



*Slika 2 Odredišna stranica (eng. landing page) aplikacije*

B I U | Heading | ☰

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vel fringilla purus. Nunc quis quam vitae nisi hendrerit pulvinar. Aenean nisi ante, lacinia faucibus convallis vitae, hendrerit in lacus. Nullam accumsan viverra magna sed dapibus. Nullam efficitur vulputate dui suscipit tincidunt. Sed aliquet neque in erat pellentesque, eu volutpat tortor venenatis. Nunc ullamcorper urna nec arcu suscipit condimentum. Aliquam consequat nibh a felis commodo aliquam. Donec tincidunt, nibh eu accumsan ultrices, massa quam dapibus diam, sed condimentum ex velit id ipsum. Pellentesque arcu lorem, hendrerit vel lacinia a, vestibulum eu libero.

Aliquam eu facilisis mi. Donec condimentum, elit vel porttitor vulputate, felis ex porta velit, id laoreet leo urna sed nibh. Pellentesque tristique sit amet turpis ut auctor. Curabitur ultrices pretium ex, sed mattis quam efficitur et. Mauris eget felis ullamcorper, pharetra sem eu, aliquet massa. Donec est nulla, tempus a ante id, tristique auctor nisi. Phasellus scelerisque porta vehicula. In sed pellentesque tortor. Etiam euismod, mauris in ullamcorper posuere, tellus nisi sagittis erat, interdum pharetra leo est volutpat nulla. Donec et diam condimentum, venenatis eros eget, pretium enim. Mauris pulvinar arcu ac egestas dapibus. Nam lacinia egestas ante. Proin faucibus nisi ut purus convallis commodo.

Nulla sagittis libero eget viverra fermentum. Vivamus in pretium lacus. In interdum, erat nec ornare porta, odio ex consequat neque, nec pellentesque justo magna id diam. Fusce at mauris libero. Sed vitae ex sollicitudin, malesuada quam vel, placerat eros. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam efficitur mi at nibh porttitor, a fermentum erat porttitor. Nulla eget lectus vitae ex hendrerit sollicitudin ut vitae tortor. Sed sagittis mauris eleifend, tincidunt felis iaculis, rutrum tortor. In at justo vel mauris venenatis ornare non et odio. Proin rhoncus, lectus sed suscipit condimentum, felis est mollis risus, id tristique nisi augue in odio. Praesent quam mauris, interdum nec metus et, congue ornare diam. In a congue enim, quis pulvinar augue. Sed at tempor tortor, posuere pulvinar elit.

Morbi viverra congue quam, et ultrices odio pharetra eget. Nulla quis nulla eu justo vestibulum faucibus sed vel massa. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Integer bibendum id nibh vitae bibendum. In tristique egestas commodo. Integer euismod, metus eget faucibus vulputate, erat dui congue neque, convallis bibendum metus neque eu diam. Proin tincidunt at nisi et rutrum. Nunc at ligula vulputate, sagittis ligula sed, venenatis tellus. Nunc eu magna tellus. Aliquam id ex sed eros fermentum luctus. Donec in convallis odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Praesent est eros, efficitur porttitor diam sit amet, pretium dignissim sapien.

Submit

^- Sentiment Label: neutral

☰ Sentiment Score: 0

😊 Joy Score: 0

☹ Sadness Score: 0

😡 Anger Score: 0

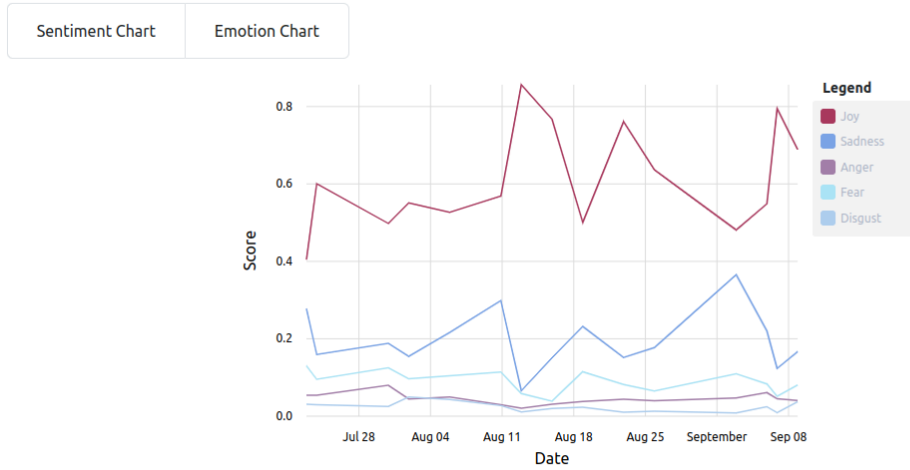
😱 Fear Score: 0

🤢 Disgust Score: 0

### *Slika 3 Vizualni prikaz novog korisničkog unosa*

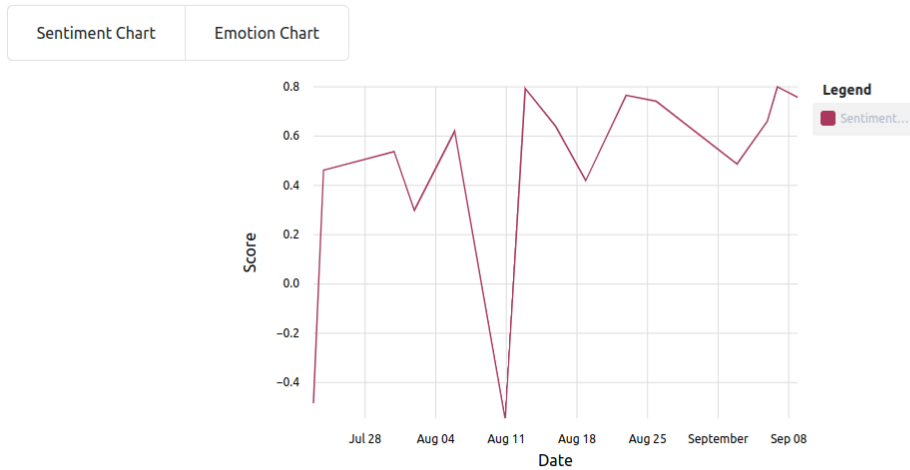
- Vizualizacija podataka:** vizualni prikaz podataka ima ključnu ulogu u razumijevanju emocionalnih obrazaca korisnika. Za vizualizaciju se koristi biblioteka *ngx-charts*, koja omogućuje kreiranje različitih tipova grafikona unutar Angular aplikacije. Korištenjem kružnih i stupčastih grafikona te drugih vizualnih prikaza korisnici mogu brzo i intuitivno razumjeti rezultate svojih emocionalnih analiza. Ova vizualizacija pruža ne samo estetski privlačan prikaz već i funkcionalnu vrijednost – jer olakšava korisnicima prepoznavanje obrazaca i trendova u svojim emocionalnim stanjima tijekom vremena.

## Sentiment and Emotion Over Time



Slika 4 Grafikon emocionalnih unosa korisnika

## Sentiment and Emotion Over Time



Slika 5 Grafikon sentimenta korisnika

### 5.1.3. Backend arhitektura

Backend aplikacije *Dnevnik analize emocija* izgrađen je pomoću Django REST Frameworka, što omogućuje brzo i efikasno kreiranje API-ja za manipulaciju podacima. Django REST Framework je odabran zbog svoje robusnosti, fleksibilnosti i jednostavnosti integracije s postojećim Django projektima. Ovaj framework omogućuje definiranje složenih poslovnih logika i pravila unutar API-ja, uz minimalan napor u pogledu pisanja koda, čime se značajno ubrzava razvojni proces.

- **Modeli podataka:** backend aplikacije temelji se na dva glavna modela - *UserInput* i *SentimentEmotion*. Model *UserInput* služi za pohranu tekstualnih unosa korisnika, dok model *SentimentEmotion* pohranjuje rezultate emocionalnih analiza koje se generiraju nakon obrade unosa putem IBM Watson NLP servisa. Ovi modeli su definirani unutar *Django ORM* sustava, što omogućuje jednostavno upravljanje podacima kroz Python kod, bez potrebe za direktnim pisanjem SQL upita. Django ORM automatski upravlja odnosima između različitih modela, pružajući intuitivno sučelje za rad s podacima. Na primjer, kada se kreira novi korisnički unos, povezani emocionalni rezultati se automatski stvaraju i povezuju s tim unosom, čime se osigurava konzistentnost podataka.

```
class SentimentEmotion(models.Model): 11 usages  👤 pakel
    user_input = models.ForeignKey(UserInput, on_delete=models.CASCADE)
    sentiment_score = models.FloatField()
    sentiment_label = models.CharField(max_length=20)
    joy_score = models.FloatField()
    sadness_score = models.FloatField()
    anger_score = models.FloatField()
    fear_score = models.FloatField()
    disgust_score = models.FloatField()
```

Slika 6 Postavka *SentimentEmotion* modela

- **Viewsets i API rute:** backend aplikacije koristi *viewsets* za definiranje API ruta koje omogućuju pristup i upravljanje podacima o korisničkim unosima i rezultatima emocionalne analize. Korištenje viewsetova omogućuje razvoj API-ja na deklarativan način, gdje se specificira samo osnovna poslovna logika, dok se sve ostalo automatski upravlja unutar

Django REST Frameworka. *UserInputViewSet* i *SentimentEmotionViewSet* pružaju osnovne funkcionalnosti za CRUD (Stvoriti, pročitati, ažurirati, izbrisati; eng. Create, Read, Update, Delete) operacije nad korisničkim unosima i njihovim analizama. Dodatni viewsetovi, poput *AverageWeekScoresViewSet* i *AverageMonthScoresViewSet*, omogućuju proračune prosječnih emocionalnih rezultata na tjednoj i mjesečnoj razini, što korisnicima pruža uvid u dugoročne promjene njihovih emocionalnih stanja. Ovi viewsetovi su ključno sredstvo za agregaciju podataka, omogućujući korisnicima da vide širu sliku svojih emocionalnih obrazaca kroz vrijeme.

- **Komunikacija s NLP servisom:** jedan od ključnih zadataka backenda je komunikacija s IBM Watson NLP servisom, koji se koristi za analizu tekstualnih unosa korisnika. Funkcija `analyze_text` šalje podatke unosa NLP servisu, koji zatim vraća rezultate analize. Ovi rezultati uključuju emocionalne ocjene za različite emocije kao što su radost, tuga, bijes, strah i gađenje. Proces integracije s IBM Watson NLP-om je jednostavan i direktan, omogućujući backendu da efikasno koristi napredne algoritme obrade prirodnog jezika bez potrebe za izradom vlastitih modela ili infrastrukture. Rezultati se zatim pohranjuju u *SentimentEmotion* modelu, a sustav je dizajniran tako da automatski povezuje rezultate s odgovarajućim korisničkim unosima. Ova integracija osigurava da su svi podaci sinkronizirani i lako dostupni za daljnju analizu i vizualizaciju.

```
def analyze_text(text, features):  # Marija
    response = natural_language_understanding.analyze(
        text=text,
        features=features
    ).get_result()

    return response
```

Slika 7 Funkcija `analyze_text` koja komunicira sa Watson NLP-om

- **Baza podataka:** za pohranu podataka koristi se SQLite baza podataka, koja je integrirana unutar Django sustava. SQLite je odabran zbog svoje jednostavnosti i kompatibilnosti s Django ORM-om, što omogućuje brzo postavljanje i jednostavno upravljanje podacima. Budući da je aplikacija zamišljena kao prototip i koristi se lokalno, SQLite pruža dovoljne performanse i kapacitete za pohranu podataka. Baza podataka je



strukturirana tako da omogućí učinkovito upravljanje unosima i emocionalnim analizama, uz minimiziranje potreba za održavanjem. Svi podaci se pohranjuju lokalno na korisnikovom računalu, što osigurava privatnost i sigurnost, jer su podaci dostupni samo korisniku aplikacije. Django ORM upravlja svim aspektima komunikacije s bazom, uključujući kreiranje, ažuriranje i brisanje zapisa, čime se znatno smanjuje složenost koda i olakšava održavanje.

#### **5.1.4. Komunikacija između frontenda i backenda**

Komunikacija između frontenda i backenda odvija se putem REST API-ja, što se ostvaruje korištenjem HTTP zahtjeva. U ovom procesu, Angular aplikacija koristi HttpClient servis, koji je specijalizirani alat unutar Angular okvira za slanje zahtjeva prema backendu. Nakon što frontend pošalje određeni zahtjev, backend ima zadatak obraditi te zahtjeve – a što obuhvaća različite operacije nad podacima te vraćanje odgovarajućih odgovora. Ti odgovori se najčešće dostavljaju u obliku JSON podataka, jer je JSON format standardiziran i vrlo pogodan za razmjenu podataka između različitih sustava. Takav mehanizam omogućuje asinkronu komunikaciju između frontenda i backenda. Asinkronost znači da se zahtjevi mogu slati, a odgovori primati bez da korisničko sučelje postane neupotrebljivo ili zamrznuto dok backend izvršava određene, ponekad vrlo složene operacije. Čak i kada backend provodi kompleksnu obradu podataka, poput komunikacije s vanjskim NLP servisima, aplikacija ostaje responzivna i funkcionalna. Asinkrona priroda ovog mehanizma ima ključnu ulogu u održavanju fluidnog i ugodnog korisničkog iskustva, jer omogućuje korisnicima da nastave koristiti aplikaciju bez prekida, bez obzira na trajanje ili složenost obrade podataka na backendu.

Kada frontend šalje zahtjeve prema backendu, ti zahtjevi mogu biti različitih tipova, ovisno o funkcionalnostima aplikacije. Među najčešćim zahtjevima su oni koji se koriste za stvaranje novih unosa u bazu podataka, dohvaćanje rezultata analize emocija iz NLP servisa kao i prikupljanje već spremljenih podataka o prosječnim emocionalnim stanjima korisnika tijekom određenog vremenskog razdoblja, poput tjedna ili mjeseca. Svaki od ovih zahtjeva backend obrađuje unutar svojih viewsetova. Viewsetovi su posebne komponente na backendu koje se koriste za definiranje načina obrade i odgovaranja na različite HTTP zahtjeve. Oni koriste *serijalizatore*, koji imaju ključnu ulogu u pretvaranju podataka između modela koji se koriste unutar aplikacije i JSON formata, koji je primjenjiv za razmjenu podataka. Ovaj proces

omogućuje dosljednu, točnu i sigurnu razmjenu podataka između različitih dijelova aplikacije, što je ključno za njezinu pouzdanost i sigurnost.

Primjenom REST API arhitekture, aplikacija je osposobljena za jednostavno proširenje svojih mogućnosti komunikacije i integracije s drugim sustavima ili vanjskim servisima. Na primjer, aplikacija može u budućnosti implementirati dodatne funkcionalnosti kao što su korisnička autentifikacija ili dublja integracija s naprednim NLP servisima koji nude specifične vrste analiza ili funkcionalnosti. Jedna od glavnih prednosti ove arhitekture jest to što omogućuje da se takva proširenja mogu implementirati bez potrebe za opsežnim promjenama u postojećem kodu. To je omogućeno modularnim dizajnom aplikacije, u kojem su svi dijelovi jasno odvojeni i definirani, te zahvaljujući API rutama koje su strukturirane na način da podržavaju jednostavno dodavanje novih funkcionalnosti ili promjenu postojećih.

## 5.2. Sigurnosni aspekti

Upravljanje sigurnošću aplikacije *Dnevnik analize emocija* je ključno, čak i ako se radi o prototipu s osnovnim funkcionalnostima. Sigurnosni aspekti aplikacije osmišljeni su u vidu lokalnog korištenja, što znači da podaci korisnika nisu pohranjeni na vanjskim poslužiteljima ili oblacima, nego u SQLite bazi podataka na korisnikovom računalu. Ovaj pristup znatno smanjuje rizik od neovlaštenog pristupa ili curenja podataka jer svi podaci ostaju na korisnikovom uređaju. Lokalna pohrana pruža osnovnu razinu sigurnosti jer se podaci ne šalju na udaljene poslužitelje, ali također dolazi s određenim rizicima. Ako bi došlo do neovlaštenog pristupa korisnikovom računalu, netko bi mogao potencijalno pristupiti i podacima unutar SQLite baze jer podaci nisu kriptirani. U slučaju daljnjeg razvoja aplikacije, preporučuje se implementiranje enkripcije podataka u bazi kako bi se dodatno zaštitili korisnički podaci.

Uz to, aplikacija koristi IBM Watson NLP servis za analizu tekstualnih unosa korisnika, što uključuje slanje podataka s aplikacije na Watsonov API. Kako bi se osigurala sigurnost tijekom prijenosa podataka između backenda i Watson NLP API-ja, koristi se *Secure Sockets Layer* (SSL) certifikacija. SSL certifikati omogućuju enkripciju podataka tijekom prijenosa, čime se sprječava prisluškivanje, presretanje ili manipulacija podacima od strane neovlaštenih subjekata. Ova mjera osigurava da su korisnički podaci, uključujući osjetljive informacije vezane uz emocionalna stanja korisnika, zaštićeni dok su u tranzitu između aplikacije i NLP servisa.

Iako su osnovne sigurnosne mjere implementirane, aplikacija trenutačno ne uključuje napredne sigurnosne značajke kao što su autentifikacija i autorizacija korisnika. Budući da je aplikacija zamišljena kao prototip za jednog korisnika i koristi se isključivo lokalno, ove značajke nisu bile prioritet tijekom razvoja. Međutim, arhitektura aplikacije omogućuje jednostavno proširenje i dodavanje ovih sigurnosnih funkcionalnosti u budućnosti. Ako bi aplikacija bila skalirana ili korištena u okruženju s više korisnika, bilo bi ključno implementirati sustave autentifikacije koji bi osigurali da samo ovlašteni korisnici mogu pristupiti svojim podacima, kao i sustave autorizacije koji bi regulirali pristup specifičnim funkcionalnostima unutar aplikacije.

Osim toga, aplikacija je razvijena uz primjenu SOLID principa dizajna, što dodatno doprinosi sigurnosti koda i smanjuje mogućnost uvođenja sigurnosnih propusta. Korištenje principa jedinstvene odgovornosti, koji je osnova SOLID principa, omogućuje da svaki dio aplikacije ima jasno definiranu ulogu, što olakšava prepoznavanje i rješavanje sigurnosnih problema. Primjerice, komunikacija s Watson NLP API-jem je izolirana u poseban modul, što omogućuje da se svi sigurnosni zahtjevi i prilagodbe vezane uz tu funkcionalnost lako primijene i kontroliraju na jednom mjestu.

Iako su osnovne sigurnosne mjere primijenjene, budući razvoj aplikacije mogao bi uključivati dodatne sigurnosne mjere, kao što su kriptiranje podataka unutar baze, uvođenje autentifikacijskih protokola te redovito ažuriranje softverskih komponenti radi zaštite od novih sigurnosnih prijetnji. U konačnici, ciljevi sigurnosti u aplikaciji *Dnevnik analize emocija* usmjereni su na zaštitu privatnosti korisnika i integritet njihovih podataka, čime se stvara sigurno okruženje za korištenje aplikacije, čak i u njezinoj početnoj fazi kao prototipa. Dodatno, fleksibilna arhitektura omogućuje buduće proširenje sigurnosnih funkcionalnosti, što aplikaciji otvara vrata za širu primjenu, uključujući i profesionalna okruženja gdje je sigurnost podataka od presudne važnosti.

### **5.3. Backend implementacija**

Implementacija backenda aplikacije *Dnevnik analize emocija* temelji se na korištenju Django REST Frameworka, modernog alata za izradu API-ja u Pythonu, koji omogućuje učinkovitu obradu podataka i interakciju s frontend dijelom aplikacije razvijenim u Angularu.

Backend arhitektura je osmišljena prema SOLID principima razvoja softvera, s posebnim naglaskom na načelo jednostruke odgovornosti, što osigurava čitljivost koda, lako održavanje i skalabilnost sustava. Ovaj odjeljak detaljno će opisati sve ključne komponente, funkcionalnosti i tehničke odluke koje su dovele do izgradnje učinkovitog i modularnog backenda.

### 5.3.1. Struktura backend sustava

Backend aplikacije *Dnevnik analize emocija* osmišljen je na način da bude strukturiran u nekoliko glavnih modula, pri čemu svaki modul obavlja specifične funkcije koje su nužne za ispravan rad i funkcioniranje sustava. Ova jasno definirana struktura omogućuje bolju organizaciju koda, što rezultira većom efikasnošću i lakšim održavanjem aplikacije. Svaki modul odgovoran je za određeni aspekt rada sustava, a ta podjela zadataka doprinosi preglednosti koda te olakšava buduća proširenja i nadogradnje aplikacije. Ključne komponente koje čine okosnicu ove modularne arhitekture uključuju modele podataka, viewsetove, serijalizatore, administracijsko sučelje, kao i komunikaciju s IBM Watson NLP servisom, koja se odvija putem posebnog modula pod nazivom *nlp\_utils*. Ova modularnost omogućuje jasan i precizan raspored odgovornosti unutar aplikacije, gdje svaka pojedina komponenta ima točno definiranu ulogu, što doprinosi većoj koherentnosti cijelog sustava i omogućuje lakšu prilagodbu promjenama ili dodavanje novih funkcionalnosti. Glavni moduli backenda, koji čine ovu strukturu, uključuju sljedeće:

- **Modeli podataka (models.py):** ovaj modul odgovoran je za definiranje strukture podataka koji se pohranjuju u bazi podataka aplikacije. Drugim riječima, modeli predstavljaju kostur podataka, pri čemu se koriste za definiranje što sve pojedini zapisi u bazi podataka moraju sadržavati. U okviru ovog modula nalaze se ključni modeli kao što su *UserInput* i *SentimentEmotion*, koji pohranjuju informacije o unosima koje korisnici šalju putem aplikacije, kao i rezultate emocionalne analize dobivene od NLP servisa. Ovi modeli imaju centralnu ulogu u cijelom sustavu, jer bez njih ne bi bilo moguće pravilno pohraniti ni obraditi podatke unutar baze podataka.

- **ViewSetovi (viewsets/router.py):** Viewsetovi su modul koji definira API rute, što znači da oni određuju kako aplikacija pristupa podacima i kako njima upravlja. Korištenjem viewsetova, backend aplikacije omogućuje niz operacija poput kreiranja novih zapisa, dohvaćanja postojećih podataka, ažuriranja već unesenih informacija, kao i brisanja podataka. Ovaj modul je od iznimne važnosti jer omogućuje komunikaciju između korisnika i sustava, te pruža backendu mogućnost pružanja API ruta za različite tipove podataka i funkcionalnosti.
- **Serijalizatori (serializers.py):** Serijalizatori imaju ključnu ulogu u pretvorbi podataka između formata koji koriste modeli i formata koji koristi frontend aplikacije. Konkretnije, oni omogućuju da se podaci iz modela pretvore u JSON format, koji je standardiziran format za razmjenu podataka u web aplikacijama, te obrnuto – da se podaci u JSON formatu pravilno pretvore natrag u modele kako bi se mogli pohraniti u bazu podataka. Ovaj proces je ključan za pravilnu komunikaciju između različitih dijelova aplikacije, a serijalizatori omogućuju da taj proces bude efikasan i pouzdan.
- **Komunikacija s NLP-om (nlp\_utils.py):** Ovaj specifični modul omogućuje da aplikacija šalje korisničke podatke IBM Watson NLP servisu te obrađuje rezultate koje dobiva natrag. Proces se sastoji od slanja korisničkih unosa prema NLP servisu koji zatim vrši analizu emocija prisutnih u tekstu, te vraća rezultate emocionalne analize natrag aplikaciji, gdje se ti rezultati mogu koristiti za daljnju obradu ili prikaz korisnicima. Komunikacija s NLP servisom od suštinske je važnosti za funkcionalnost aplikacije, jer upravo ona omogućuje analizu emocionalnog stanja na temelju korisničkih unosa.
- **Administracijsko sučelje (admin.py):** Administracijsko sučelje koristi ugrađenu funkcionalnost Django okvira kako bi omogućilo grafički prikaz za upravljanje podacima. Ova funkcionalnost omogućuje administratorima sustava da na jednostavan i pregledan način upravljaju svim podacima unutar aplikacije, bez potrebe za direktnim pristupom bazi podataka ili pisanjem posebnih skripti. Korištenje administracijskog sučelja olakšava održavanje aplikacije i omogućuje administratorima da na jednostavan način vrše promjene ili nadgledaju rad aplikacije kroz intuitivno grafičko sučelje.

### 5.3.2. Detalji o API-ju

Django REST Framework odabran je kao temelj za izgradnju API-ja u ovoj aplikaciji zbog svoje velike fleksibilnosti, jednostavnosti korištenja i mogućnosti brze implementacije čak i složenih poslovnih logika. Ovaj okvir omogućuje izgradnju API-ja na način koji olakšava razvoj i održavanje koda, dok istovremeno omogućuje modularnost i proširivost. API je dizajniran prema REST principima, što znači da koristi standardne HTTP metode (GET, POST, PUT, DELETE) za interakciju s podacima, omogućujući efikasno upravljanje resursima u aplikaciji. Ove metode su jednostavne za implementaciju i omogućuju dosljednu komunikaciju između frontenda i backenda.

Ključne rute API-ja uključuju sljedeće:

- **/user-input**: ova ruta koristi se za upravljanje korisničkim unosima. Pruža funkcionalnosti za kreiranje novih unosa, dohvaćanje postojećih podataka, kao i mogućnost filtriranja unosa na temelju specifičnih parametara, što omogućuje preciznije pretraživanje i manipulaciju podacima.
- **/sentimentemotion**: ova ruta omogućuje upravljanje rezultatima emocionalne analize. Korisnicima omogućuje dohvaćanje rezultata emocionalnih analiza povezanih s njihovim unosima. To im pomaže u praćenju i razumijevanju njihovih emocionalnih obrazaca kroz vrijeme.
- **/averageweek** i **/averagemonth**: ove rute omogućuju proračun prosječnih emocionalnih stanja na tjednoj i mjesečnoj razini. Omogućavaju korisnicima praćenje emocionalnih promjena u dužem vremenskom razdoblju, pružajući uvid u šire emocionalne trendove i obrasce.

Svaka od navedenih ruta koristi specifične serijalizatore kako bi podaci iz modela bili ispravno pretvoreni u JSON format, koji se zatim šalje frontendu aplikacije. Ovaj pristup osigurava dosljednost podataka, omogućujući da se manipulacije nad podacima obavljaju u skladu s pravilima i sigurnosnim protokolima definiranim unutar aplikacije. Zahvaljujući tome, podaci se razmjenjuju na siguran i kontroliran način, osiguravajući da sve operacije koje korisnici izvršavaju budu pouzdane i precizne.

### 5.3.3. Rad s bazom podataka

Za pohranu podataka, aplikacija koristi SQLite bazu podataka, koja je integrirana unutar Django okvira. SQLite je odabran zbog svoje jednostavnosti i pogodnosti za prototip aplikacije koji ne zahtijeva visoku razinu skalabilnosti. Baza podataka pohranjuje informacije o korisničkim unosima i rezultatima emocionalne analize, pri čemu Django ORM upravlja svim aspektima interakcije s bazom podataka, uključujući kreiranje, dohvaćanje, ažuriranje i brisanje podataka.

- **Struktura podataka:** model `UserInput` pohranjuje tekstualne unose korisnika zajedno s datumom unosa, dok model `SentimentEmotion` pohranjuje rezultate analize, uključujući ocjene za različite emocije kao što su radost, tuga, bijes, strah i gađenje. Svaki zapis u `SentimentEmotion` modelu povezan je s odgovarajućim unosom u `UserInput` modelu putem vanjskog ključa, čime se osigurava povezanost podataka.
- **Upravljanje podacima:** podaci se pohranjuju i dohvaćaju korištenjem Django ORM-a, što omogućuje jednostavnu interakciju s bazom podataka kroz Python kod. Na primjer, kada korisnik unese novi tekstualni unos, aplikacija automatski stvara odgovarajuće zapise u bazi podataka, uključujući rezultate analize emocija. Funkcije `average_by_week` i `average_by_month` koriste se za izračun prosjeka emocionalnih ocjena tijekom određenih vremenskih razdoblja, pružajući korisnicima dugoročne analitičke uvide.
- **Optimizacija upita:** kako bi se poboljšale performanse pri dohvaćanju podataka, korištene su funkcije `TruncWeek` i `TruncMonth` iz Django ORM-a, koje omogućuju grupiranje podataka prema tjednima ili mjesecima. Ovo grupiranje omogućuje aplikaciji da učinkovito izračuna prosječne emocionalne ocjene za svako vremensko razdoblje, što je ključno za pružanje korisnicima preglednih i korisnih informacija.

### 5.3.4. Upravljanje performansama i skalabilnost

Upravljanje performansama i skalabilnošću aplikacije *Dnevnik analize emocija* nadilazi samo tehničku arhitekturu i obuhvaća pristupe optimizaciji aplikacije u stvarnim uvjetima uporabe. Iako je trenutna verzija aplikacije dizajnirana kao prototip, razmotrene su strategije koje bi

mogle omogućiti skaliranje prema većem broju korisnika ili složenijim scenarijima analize podataka.

Jedan od potencijalnih pristupa koji se može razmotriti u budućnosti je horizontalno skaliranje aplikacije kroz korištenje tehnologija poput *kontejnerizacije s Dockerom* ili *orkestracije* pomoću *Kubernetes* sustava. To bi omogućilo aplikaciji da se lako razmješta na više servera ili čvorova, povećavajući tako kapacitet obrade podataka i smanjujući vrijeme odziva korisnicima. Ovaj pristup bi bio osobito koristan u situacijama kada bi se aplikacija proširila na veće populacije korisnika ili kada bi se obujam podataka značajno povećao.

Osim toga, optimizacija performansi može uključivati implementaciju naprednijih tehnika keširanja podataka. Na primjer, podaci koji se često dohvaćaju, poput prosjeka emocija za određena vremenska razdoblja, mogu se privremeno pohraniti u memoriji pomoću *Redis* ili *Memcached* servisa. Ova tehnika smanjuje broj upita prema bazi podataka i ubrzava vrijeme odgovora aplikacije, što je posebno važno za aplikacije koje zahtijevaju brze i česte pristupe podacima.

Za dugoročno upravljanje performansama, može se razmotriti uvođenje sustava za praćenje performansi i metrike aplikacije. Alati poput *Prometheusa* i *Grafane* mogu se koristiti za praćenje resursa sustava, vremena odziva, učestalosti pogrešaka i drugih ključnih metrika koje pružaju uvid u stvarne performanse aplikacije tijekom vremena. Ovi podaci mogu se koristiti za kontinuirano poboljšanje aplikacije, omogućujući brzu identifikaciju i rješavanje uskih grla ili problema u sustavu.

Uz to, uvođenje sofisticiranih algoritama za ravnotežu opterećenja može značajno poboljšati performanse aplikacije, osobito u okruženjima s velikim brojem korisnika. Implementacija *load balancera* može ravnomjerno rasporediti promet između različitih instanci aplikacije, sprječavajući preopterećenje pojedinih komponenti i osiguravajući stabilnost sustava. Ovo je važan korak u slučaju daljnjeg razvoja aplikacije u distribuiranom okruženju ili u oblaku.

U pogledu optimizacije koda, fokus na kontinuirano poboljšanje efikasnosti funkcija, refaktoriranje postojećih metoda i uklanjanje suvišnih procesa može značajno doprinijeti ukupnim performansama aplikacije. Automatizirano testiranje performansi, poput stres testova i testova opterećenja se mogu koristiti za simulaciju stvarnih uvjeta uporabe i za procjenu kako aplikacija reagira pod različitim uvjetima. Na kraju, važan aspekt upravljanja performansama i skalabilnosti je i organizacijski pristup DevOps kulturi, koja promiče blisku suradnju između razvojnih i operativnih timova. Uvođenjem *CI/CD (Continuous Integration/Continuous*



*Deployment*) *pipeline* se aplikacija može brzo i sigurno ažurirati, s minimalnim prekidima u radu i smanjenim rizikom od regresija. Ovaj pristup omogućuje brze iteracije i poboljšanja, čineći aplikaciju *Dnevnik analize emocija* prilagodljivom i spremnom za sve buduće izazove i zahtjeve korisnika.

Ukratko, upravljanje performansama i skalabilnost u kontekstu *Dnevnika analize emocija* uključuje ne samo trenutačne mjere optimizacije, već i strateške pristupe koji omogućuju da se aplikacija razvija i prilagođava budućim potrebama. Ovi aspekti su ključni za dugoročnu održivost aplikacije i njezinu sposobnost da pruži korisnicima stabilno i kvalitetno iskustvo, bez obzira na promjene u opsegu uporabe ili tehničke zahtjeve.

### 5.3.5. Izazovi i naučene lekcije

Tijekom razvoja backenda aplikacije *Dnevnik analize emocija*, jedan od najvećih izazova bio je pravilno razumijevanje i upravljanje rezultatima dobivenim od IBM Watson NLP servisa. Rezultati NLP-a često uključuju složene strukture podataka koje je bilo potrebno ispravno interpretirati i pohraniti na način koji omogućuje lako dohvaćanje i vizualizaciju na frontendu aplikacije. Ovaj izazov je riješen kroz pažljivo dizajnirane serijalizatore i funkcije unutar *nlp\_utils* modula, koje obrađuju podatke prije nego što se pohrane u bazu.

Još jedan značajan izazov bio je upravljanje performansama aplikacije, osobito s obzirom na ograničenu skalabilnost SQLite baze podataka. Kako bi se osiguralo da aplikacija može podnijeti povećanje broja korisnika i unosa, primijenjene su tehnike grupiranja podataka prema vremenskim razdobljima i asinkroni pristupi za analizu podataka, što je značajno poboljšalo performanse bez potrebe za prelaskom na složenije baze podataka.

Izgradnja backenda također je pokazala važnost modularnosti i jasnog definiranja odgovornosti unutar sustava. Primjenom SOLID principa i razdvajanjem funkcionalnosti na specifične module, aplikacija je postala lako proširiva i prilagodljiva budućim zahtjevima. Iako je trenutna verzija aplikacije osnovna, arhitektura je postavljena tako da omogućuje dodavanje novih značajki, sigurnosnih mjera ili poboljšanja performansi s minimalnim izmjenama u postojećem kodu.

## 5.4. Frontend implementacija

Frontend implementacija aplikacije *Dnevnik analize emocija* izrađena je pomoću Angular frameworka, što omogućuje razvoj složenih i responzivnih web aplikacija kroz komponentno usmjeren pristup.

Glavni cilj ove implementacije bio je osigurati jednostavno i intuitivno korisničko sučelje koje omogućuje korisnicima praćenje i analizu njihovih emocionalnih stanja kroz interaktivne vizualizacije i pristupačne funkcionalnosti. Kroz primjenu SOLID principa, frontend je dizajniran na modularan način, što osigurava čitljivost, održivost i proširivost koda.

### 5.4.1. Struktura aplikacije

Frontend aplikacije je strukturiran u skladu s Angular arhitekturom, koristeći komponentno usmjeren dizajn koji omogućuje razdvajanje funkcionalnosti u manje, specifične dijelove. Glavni dijelovi aplikacije uključuju module, komponente, servise i zajedničke resurse, čime se postiže modularnost i jasnoća u kodu.

- **Komponente:** Angular komponente su osnovni građevni blokovi aplikacije. U aplikaciji *Dnevnik analize emocija*, komponente su organizirane unutar direktorija `components`, koji uključuje:
  - **HomeComponent:** prikazuje početnu stranicu aplikacije s osnovnim informacijama i pozivom na čin ručnog unošenja emocija.
  - **MainComponent:** centralna komponenta za unos korisničkih podataka. Korisnici putem ove komponente unose svoje dnevne unose, a zatim se podaci šalju backendu na analizu.
  - **AboutComponent:** pruža korisnicima informacije o aplikaciji, njenom autoru i svrsi korištenja.
  - **StatsComponent:** prikazuje rezultate analize emocija kroz različite vizualizacije poput grafikona, čime se omogućuje korisnicima praćenje njihovih emocionalnih stanja tijekom vremena.

- **Navigacija:** navigacija unutar aplikacije je implementirana pomoću Angular Routera. Definirane su rute koje omogućuju korisnicima jednostavno prebacivanje između različitih dijelova aplikacije. Ruta *appRoutes* uključuje putanje za glavne komponente aplikacije (*HomeComponent*, *MainComponent*, *AboutComponent*, i *StatsComponent*), a navigacija se upravlja putem *NavbarComponent* koji pruža intuitivnu navigaciju korisnicima.
- **Servisi:** Angular servisi, kao što je *ApiService*, ključni su za komunikaciju između frontenda i backenda. *ApiService* upravlja svim API pozivima prema backendu, uključujući kreiranje novih korisničkih unosa, dohvaćanje rezultata emocionalne analize i skladištenje podataka za tjedne i mjesečne prosjeke. Koristeći *HttpClient*, servis omogućuje asinkronu komunikaciju s backendom, osiguravajući da korisničko sučelje ostaje responzivno.
- **Zajednički resursi:** zajednički direktorij uključuje zajedničke komponente poput *NavbarComponent*, koji omogućuje dosljedan izgled i funkcionalnost kroz cijelu aplikaciju. Dodatno, direktorij *common* sadrži funkcije za rukovanje pogreškama (npr. *api\_errors.ts*), što omogućuje centralizirano upravljanje greškama u aplikaciji.

#### 5.4.2. Dizajn korisničkog sučelja

Dizajn korisničkog sučelja temelji se na jednostavnosti i funkcionalnosti, koristeći Bootstrap kao osnovnu CSS biblioteku za stilizaciju elemenata. Korištenje Bootstrapa omogućuje brzu izradu responzivnih i estetski privlačnih komponenti, uz minimalan napor u prilagodbi stilova.

- **Stilizacija i responzivnost:** iako je Bootstrap primarni alat za stilizaciju, aplikacija koristi i prilagođene CSS stilove za dodatne prilagodbe i jedinstven izgled. Fokus je na neutralnim bojama i jednostavnom dizajnu kako bi se izbjeglo neželjeno utjecanje na emocionalna stanja korisnika prilikom korištenja aplikacije. Iako Bootstrap podržava responsivnost, aplikacija je specifično dizajnirana za desktop korištenje, što je uzeto u obzir prilikom odabira elemenata i njihovog pozicioniranja na ekranu.
- **Vizualizacija podataka:** prikaz emocionalnih stanja korisnika realiziran je pomoću biblioteke *ngx-charts*[21], koja omogućuje jednostavno integriranje različitih tipova grafikona u Angular aplikacije. Vizualizacija podataka je ključna za korisničko iskustvo, jer pruža jasan i intuitivan pregled emocionalnih obrazaca kroz vrijeme. Aplikacija koristi

kružne grafikone za prikaz udjela različitih emocija, što korisnicima omogućuje brzo i lako razumijevanje rezultata analize.

- **Korisničko sučelje i intuitivnost:** posebna pažnja posvećena je izradi intuitivnog korisničkog sučelja, s jasno definiranim akcijama i minimalnim brojem koraka potrebnih za postizanje željenog cilja. Na primjer, unos novih emocija je direktan i jasan, a svi rezultati analize su prikazani na pregledan i lako shvatljiv način. Korisnici se lako mogu prebacivati između različitih dijelova aplikacije putem navigacijske trake, što dodatno doprinosi pozitivnom korisničkom iskustvu.

### 5.4.3. Komunikacija s backendom

Komunikacija između frontenda i backenda odvija se putem HTTP zahtjeva, pri čemu frontend šalje zahtjeve backendu koji zatim obrađuje podatke i vraća odgovore u obliku JSON objekata. Ovaj asinkroni proces omogućuje aplikaciji da ostane responsivna, čak i kada backend obrađuje složene operacije poput analize teksta pomoću IBM Watson NLP servisa.

Upravljanje komunikacijom ostvaruje se putem servisa ApiService, koji koristi Angularov HttpClient za slanje i primanje podataka te upravlja svim API pozivima s frontenda. ApiService definira različite metode za rad s podacima, uključujući metode kao što su: *getUserInputs*, *getUserInput*, *getSentimentEmotionByUserInputId*, *getAllSentimentEmotionData*, *createUserInput*, *getAverageWeekScores* i *getAverageMonthScores*. Ove metode koriste odgovarajuće URL rute za pristup specifičnim resursima na backendu, osiguravajući da svi podaci budu ispravno obrađeni i pohranjeni. Na primjer, kada korisnik unosi novi emocionalni dnevnik putem MainComponent, tekstualni zapis se šalje backendu putem ApiService, koristeći metodu *createUserInput*, koja šalje HTTP POST zahtjev na odgovarajuću rutu (*/api/user-input/*). Backend zatim obrađuje ovaj zahtjev, pokreće analizu teksta koristeći IBM Watson NLP, te pohranjuje rezultate u bazi podataka. Nakon uspješne obrade, backend vraća odgovor frontend aplikaciji s potvrdom o uspješnom unosu ili detaljima o analiziranim emocijama. Ova integrirana struktura osigurava učinkovitu i sigurnu komunikaciju između frontenda i backenda, podržavajući sve funkcionalnosti aplikacije, uključujući prikupljanje podataka, analizu emocija i prikaz statistika, uz zadržavanje responsivnosti i pouzdanosti korisničkog sučelja.

#### 5.4.4. Izazovi i naučene lekcije

Jedan od najvećih izazova tijekom razvoja frontenda aplikacije bio je osigurati da korisničko sučelje ostane jednostavno i intuitivno, unatoč složenosti podataka koji se obrađuju i prikazuju. Kroz ponavljajući pristup dizajnu sučelja, testiranje i prilagodbu postignuto je sučelje koje korisnicima omogućuje lako razumijevanje emocionalnih rezultata i praćenje njihovih promjena kroz vrijeme.

Dodatni izazov predstavljala je integracija s vanjskim servisima, poput IBM Watson NLP-a, koji zahtijevaju asinkronu komunikaciju i pravilno rukovanje odgovorima. Korištenjem asinkronih servisa i centraliziranog upravljanja pogreškama je osigurana pravilna reakcija aplikacije na sve situacije, pružajući korisnicima dosljedno iskustvo.

Kroz rad na frontendu je naučeno da jednostavnost dizajna često donosi najbolja rješenja, osobito kada je cilj pružiti intuitivan alat koji korisnicima pomaže u praćenju emocionalnog zdravlja. Modularnost koda, korištenje provjerenih alata i frameworkova te fokus na korisničko iskustvo ključni su elementi uspješnog razvoja aplikacije koja zadovoljava potrebe korisnika na jasan i učinkovit način.

#### 5.4.5. Rukovanje pogreškama

Rukovanje pogreškama je implementirano kroz centraliziranu funkciju *handleErrors*, koja koristi *SweetAlert* za prikazivanje vizualno privlačnih i korisnički razumljivih poruka o greškama. Funkcija *handleErrors* analizira HTTP odgovore i u slučaju grešaka, poput problema s mrežom (statusi 500-504) ili grešaka u podacima, prikazuje poruke koje pomažu korisniku da razumije i ispravi problem. Ovaj pristup osigurava da su svi potencijalni problemi jasno komunikacijski obrađeni, smanjujući frustraciju korisnika i omogućujući lakše upravljanje greškama unutar aplikacije.

U slučajevima kada dolazi do grešaka sa serverom ili mrežom, funkcija prikazuje opće obavijesti o problemima s mrežom, pružajući korisniku informacije o mogućim uzrocima, poput prekida Internet veze ili problema s dostupnošću servera. To uključuje savjete poput provjere Internet veze ili ponovnog unosa podataka, što korisnicima olakšava dijagnosticiranje i potencijalno rješavanje problema.

Ako greška nije uzrokovana mrežnim problemima, već je specifična za podatke poslane na server, *handleErrors* koristi dodatnu funkciju *prettyPrintJSON* za obradu i formatiranje grešaka iz odgovora servera. Funkcija *prettyPrintJSON* analizira sadržaj grešaka, pretvarajući ih u HTML format koji je jednostavan za pregled i razumijevanje. Obrada podataka iz različitih formata, poput JSON-a ili binarnih podataka, dodatno doprinosi svestranosti ove funkcije.

#### **5.4.6. Asinkrona komunikacija i responzivnost**

Jedan od glavnih ciljeva tijekom izrade aplikacije bio je osigurati da korisničko sučelje ostane responzivno, čak i kada backend obrađuje zahtjeve koji mogu potrajati, poput slanja teksta na analizu NLP servisu. Za postizanje ovog cilja, svi API pozivi realizirani su na asinkroni način koristeći RxJS observables, što omogućuje da korisničko sučelje nastavi s radom dok se čeka na odgovor s backenda. Na primjer, kada korisnik unese novi tekstualni zapis i pritisne gumb za slanje, *ApiService* inicira asinkroni poziv prema backendu, dok korisničko sučelje ostaje funkcionalno i omogućuje korisniku da nastavi s drugim aktivnostima unutar aplikacije.

Ovaj asinkroni pristup također omogućuje elegantno rukovanje dugotrajnijim procesima, poput učitavanja statističkih podataka ili ažuriranja emocionalnih rezultata. Primjerice, prikaz tjednih ili mjesečnih prosjeka emocija može zahtijevati složene upite prema bazi podataka, ali korištenjem asinkronih poziva, aplikacija može učinkovito prikazati učitavanje ili druge indikatore kako bi korisnik bio svjestan da se podaci obrađuju.

#### **5.4.7. Sigurnosni aspekti komunikacije**

Iako aplikacija nije dizajnirana s naglaskom na složene sigurnosne mjere, osnovni sigurnosni principi su primijenjeni kako bi se osigurala zaštita podataka. Svi podaci prenose se putem HTTPS protokola, osiguravajući enkripciju komunikacije između frontenda i backenda. Također, Angularov *HttpClient* podržava CORS (Cross-Origin Resource Sharing) postavke, koje su konfigurirane na backendu kako bi se omogućio siguran pristup resursima aplikacije.

Ovaj osnovni sigurnosni pristup osigurava da su svi podaci korisnika zaštićeni tijekom prijenosa, iako se aplikacija koristi lokalno i namijenjena je za osobnu upotrebu na

pojedinačnim računalima. Ovaj dizajn omogućuje dodatne sigurnosne funkcionalnosti u budućnosti, poput autentifikacije i autorizacije u slučaju proširenja upotrebe aplikacije.

## 5.5. Arhitektura i funkcionalnosti integracije

```
import os
from ibm_watson import NaturalLanguageUnderstandingV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

# Mock API key and URL - please add your own values
api_key = os.getenv('IBM_API_KEY', 'your-ibm-api-key')
service_url = os.getenv('IBM_SERVICE_URL', 'your-service-url')

authenticator = IAMAuthenticator(api_key)
natural_language_understanding = NaturalLanguageUnderstandingV1(
    version='2022-04-07',
    authenticator=authenticator
)
natural_language_understanding.set_service_url(service_url)
natural_language_understanding.set_disable_ssl_verification(True)
```

*Slika 8 Postavljanje IBM Watson NLP-a u projekt*

IBM Watson NLP servis koristi se za analizu emocija i sentimenta u tekstualnim unosima korisnika. Funkcionalnost je implementirana unutar posebnog modula `nlp_utils.py`, koji se brine za sve aspekte komunikacije s Watson NLP API-jem. Ovaj modul sadrži ključnu funkciju `analyze_text`, koja prima tekst korisnika i definirane značajke za analizu, kao što su emocije (radost, tuga, bijes, strah i gađenje) te opći sentiment (pozitivan, neutralan ili negativan). Ova funkcija šalje zahtjev Watson NLP API-ju, koji zatim analizira tekst koristeći svoje napredne algoritme za obradu prirodnog jezika, te vraća rezultate analize u obliku JSON odgovora.

Proces integracije s NLP-om započinje kada korisnik unese tekstualni zapis putem MainComponent sučelja. Taj se unos zatim šalje backendu putem `ApiService` servisa u Angular aplikaciji, koristeći HTTP POST zahtjev na rutu `/api/user-input/`. Backend Django aplikacije, kroz `UserInputCreateSerializer`, prima ovaj tekst, obrađuje ga i poziva funkciju `analyze_text` iz `nlp_utils.py`. Ovaj pristup omogućuje modularnost i čistoću koda, jer je logika za analizu teksta jasno odvojena od ostalih funkcionalnosti aplikacije.

Funkcija `analyze_text` koristi IBM Watson NLP API ključ za autentifikaciju i povezivanje s Watsonovim NLP servisom. Za slanje zahtjeva, koristi se REST API, gdje su svi podaci enkriptirani tijekom prijenosa kako bi se osigurala sigurnost i privatnost korisničkih podataka.



Watson NLP zatim provodi analizu koristeći svoje unaprijed trenirane modele za prepoznavanje emocija i sentimenta. Povratni odgovor uključuje sentiment score (numeričku ocjenu sentimenta) i sentiment label (oznaku kategorije poput "pozitivan", "neutralan" ili "negativan"), kao i ocjene za pojedinačne emocije (npr. radost, tuga). Ovi rezultati se pohranjuju u SentimentEmotion modelu, koji je povezan s odgovarajućim unosom u UserInput modelu putem vanjskog ključa.

### 5.5.1. Upravljanje izazovima integracije

Tijekom integracije s IBM Watson NLP servisom, jedan od glavnih izazova bio je ispravno upravljanje složenim strukturama podataka koje vraća Watsonov API. Rezultati analize često uključuju višestruke slojeve podataka, kao što su detaljne ocjene za svaku emociju i opći sentiment. Kako bi se osiguralo pravilno rukovanje ovim podacima, razvijene su prilagođene funkcije unutar serijalizatora koje obrađuju i strukturiraju podatke prije nego što se pohrane u bazu podataka. Na primjer, UserInputCreateSerializer ne samo da stvara novi zapis za korisnički unos, već također pokreće funkciju analize teksta i stvara odgovarajući zapis u SentimentEmotion modelu.

Dodatni izazov predstavljao je rad s API ključevima i zaštita pristupa NLP servisu. Kako bi se osiguralo da su svi API ključevi sigurni, koristi se *IAMAuthenticator* iz IBM Cloud SDK-a za autentifikaciju svih zahtjeva prema Watson NLP API-ju. Ovaj pristup omogućuje siguran i pouzdan način za upravljanje osjetljivim podacima, osiguravajući da samo ovlašteni zahtjevi imaju pristup analitičkim funkcijama NLP servisa.

Još jedan izazov bio je upravljanje performansama tijekom obrade velikih količina podataka. Asinkroni pristupi korišteni su za slanje zahtjeva NLP servisu, što omogućuje paralelnu obradu zahtjeva bez utjecaja na responzivnost korisničkog sučelja. Angular aplikacija koristi RxJS observables za upravljanje asinkronim pozivima, omogućavajući da su svi zahtjevi prema backendu asinkroni, čime se sprečava blokiranje sučelja dok se čeka na odgovor Watson NLP servisa.

### 5.5.2. Optimizacija i performanse

Kako bi se osigurale optimalne performanse, poseban naglasak stavljen je na optimizaciju procesa slanja i obrade podataka između Angular aplikacije i IBM Watson NLP servisa. Angularov HttpClient omogućuje lako upravljanje asinkronim HTTP zahtjevima, dok backend aplikacija koristi asinkronu obradu podataka kako bi se minimiziralo vrijeme čekanja na odgovore NLP servisa. Ovaj pristup ne samo da poboljšava ukupne performanse aplikacije, već također osigurava bolje korisničko iskustvo jer su svi zahtjevi obrađeni brzo i učinkovito.

Optimizacija performansi uključivala je i smanjenje veličine prijenosa podataka između frontenda i backenda. Na primjer, prilikom slanja podataka Watson NLP servisu, prijenos se ograničava samo na nužne informacije potrebne za analizu, čime se smanjuje opterećenje mreže i poboljšava brzina obrade. Rezultati analize su strukturirani na način koji omogućuje lako dohvaćanje i vizualizaciju na frontendu, koristeći jednostavne i jasne podatkovne strukture.

## 6. Zaključak

U ovom radu opisan je kompletan proces izrade aplikacije *Dnevnik analize emocija*, koja koristi napredne tehnologije obrade prirodnog jezika za praćenje i analizu emocionalnih stanja korisnika. Prvi korak u razvoju bio je definiranje problema i prepoznavanje potrebe za alatima koji mogu pomoći korisnicima u autorefleksiji i praćenju vlastitih emocija. Na temelju toga su razrađeni funkcionalni zahtjevi aplikacije, koji uključuju prikupljanje korisničkih unosa, analizu emocija putem *IBM Watson NLP-a*, te prikaz rezultata kroz intuitivne grafičke prikaze, čime je korisnicima omogućeno praćenje promjena emocionalnih stanja tijekom vremena.

Aplikacija je izgrađena na modularnoj arhitekturi koja razdvaja frontend i backend komponente, što omogućuje lakše održavanje, fleksibilnost i skalabilnost sustava. Frontend aplikacije razvijen je u *Angularu*, s ciljem pružanja responzivnog i korisnički pristupačnog sučelja, koristeći *Bootstrap* za stilizaciju i *ngx-charts* za vizualizaciju podataka. Ova kombinacija tehnologija omogućila je stvaranje sučelja koje je vizualno privlačno i jednostavno za navigaciju, a istovremeno dovoljno robusno da podrži složene zahtjeve analize emocija. Backend je izrađen pomoću *Django REST Frameworka*, koji omogućuje obradu podataka, integraciju s NLP servisom i pohranu podataka u SQLite bazu. Komunikacija između frontenda i backenda implementirana je pomoću asinkronih HTTP zahtjeva, što osigurava brzu i efikasnu razmjenu podataka, a time i poboljšava korisničko iskustvo.

Jedan od najvećih izazova tijekom razvoja aplikacije bilo je rukovanje kompleksnim strukturama podataka vraćenim iz NLP servisa. Implementirane su prilagođene funkcije za obradu i pohranu rezultata emocionalne analize, čime je osigurana točnost i konzistentnost prikaza podataka korisnicima. Kroz taj proces stečeno je vrijedno iskustvo rada s vanjskim API-jem i prilagodbe složenih podataka potrebama aplikacije. Dodatno, posebna pažnja posvećena je sigurnosti podataka kroz enkripciju prijenosa informacija i zaštitu osjetljivih korisničkih podataka, što je ključno za aplikacije koje rade s osobnim i emocionalnim informacijama. Ovi sigurnosni aspekti osigurali su visoku razinu povjerenja i zaštite korisnika, što je posebno važno u kontekstu aplikacija koje se bave mentalnim zdravljem.

Kao završni korak, analizirani su potencijali za daljnji razvoj i nadogradnju aplikacije. Buduće verzije mogle bi uključivati dodatne funkcionalnosti kao što su napredna autentifikacija korisnika, podrška za veće baze podataka, složenije analize emocionalnih obrazaca, te implementacija strojnog učenja za poboljšanje točnosti analize emocija i personalizaciju korisničkog iskustva. Nadalje, moguće su nadogradnje koje bi uključivale bolju integraciju s

vanjskim servisima za mentalno zdravlje, čime bi aplikacija mogla pružiti dodatnu vrijednost korisnicima i stručnjacima u području psihoterapije.

Ovaj završni rad pokazao je da je moguće uspješno implementirati umjetnu inteligenciju u aplikacije koje prate ljudske emocije, ističući važnost modularnosti, fleksibilnosti i sigurnosti u razvoju softverskih rješenja. Izrada aplikacije *Dnevnik analize emocija* pružila je praktično iskustvo u razvoju kompleksnih softverskih sustava i primjeni teorijskih znanja stečenih tijekom studija. Kroz rad na ovom projektu usvojene su ključne vještine u programiranju, integraciji vanjskih servisa i upravljanju projektima, čime je postavljen temelj za daljnji profesionalni razvoj u području informacijske tehnologije. Osim tehničkih vještina, stečeno je i bolje razumijevanje važnosti korisničkog iskustva i kako tehnologija može biti prilagođena specifičnim potrebama korisnika, što je ključno za izradu korisnički orijentiranih softverskih rješenja koja mogu imati pozitivan utjecaj na svakodnevni život korisnika.

## **Izvori:**

- [1] Goleman, D. (1996). *Emotional Intelligence: Why It Can Matter More Than IQ*. New York: Bantam Books.
- [2] Scherer, K. R. (2005). What are emotions? And how can they be measured? *Social Science Information*, 44(4), 695-729.
- [3] Benton, A., Mitchell, M., & Hovy, D. (2017). Multi-task learning for mental health using social media text. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 165-174).
- [4] Fitzpatrick, K. K., Darcy, A., & Vierhile, M. (2017). Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): A randomized controlled trial. *JMIR Mental Health*, 4(2), e19.
- [5] Rachuri, K. K., Musolesi, M., Mascolo, C., Rentfrow, P. J., Longworth, C., & Aucinas, A. (2010). EmotionSense: A mobile phones based adaptive platform for experimental social psychology research. In *Proceedings of the 12th ACM international conference on Ubiquitous computing* (pp. 281-290).
- [6] Mohr, D. C., Zhang, M., & Schueller, S. M. (2017). Personal sensing: Understanding mental health using ubiquitous sensors and machine learning. *Annual Review of Clinical Psychology*, 13, 23-47.
- [7] Fitzpatrick, K. K., Darcy, A., & Vierhile, M. (2017). Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): A randomized controlled trial. *JMIR Mental Health*, 4(2), e19.
- [8] Goleman, D. (1996). *Emotional Intelligence: Why It Can Matter More Than IQ*. New York: Bantam Books.
- [9] Salovey, P., & Mayer, J. D. (1990). Emotional intelligence. *Imagination, Cognition and Personality*, 9(3), 185-211.
- [10] Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining Text Data* (pp. 415-463). Springer, Boston, MA.
- [11] Beck, A. T. (1976). *Cognitive Therapy and the Emotional Disorders*. New York: International Universities Press.

- [12] Kazantzis, N., Whittington, C., & Dattilio, F. M. (2010). Meta-analysis of homework effects in cognitive and behavioral therapy: A replication and extension. *Clinical Psychology Review*, 30(2), 193-202.
- [13] Koole, S. L. (2009). The psychology of emotion regulation: An integrative review. *Cognition and Emotion*, 23(1), 4-41.
- [14] Floridi, L., & Taddeo, M. (2016). What is data ethics? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2083), 20160118.
- [15] Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2).
- [16] Venkatesh, V., Thong, J. Y. L., & Xu, X. (2012). Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. *MIS Quarterly*, 36(1), 157-178.
- [17] Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319-340.
- [18] IBM Watson Natural Language Understanding. (n.d.). Retrieved from <https://www.ibm.com/watson/services/natural-language-understanding/>
- [19] Django REST Framework documentation. (n.d.). Retrieved from <https://www.django-rest-framework.org/>
- [20] Angular documentation. (n.d.). Retrieved from <https://angular.io/docs>
- [21] Ngx-charts: Declarative Charting Framework for Angular. (n.d.). Retrieved from <https://swimlane.github.io/ngx-charts/>

## Popis slika

Slika 1 Vizualni prikaz strukture frontend dijela aplikacije.....	13
Slika 2 Odredišna stranica (eng. landing page) aplikacije .....	14
Slika 3 Vizualni prikaz novog korisničkog unosa.....	15
Slika 4 Grafikon emocionalnih unosa korisnika .....	16
Slika 5 Grafikon sentimenta korisnika .....	16
Slika 6 Postavka SentimentEmotion modela .....	17
Slika 7 Funkcija analyze_text koja komunicira sa Watson NLP-om.....	18
Slika 8 Postavljanje IBM Watson NLP-a u projekt .....	34

**Poveznica na GitHub repozitorij:**

<https://github.com/mnikolic91/emotional-analysis-diary>



# DEVELOPMENT OF A WEB PLATFORM FOR THE ANALYSIS OF USERS' EMOTIONAL JOURNALS

## Summary

We live in an era of rapid development of artificial intelligence and its integration into various aspects of society, significantly contributing to the improvement of the quality of life. Motivated by exploring the potential of artificial intelligence for monitoring users' emotional states, we have developed a full-stack web application for analyzing emotional journals with the help of *Watson NLP*. The application utilizes modern frontend and backend technologies to ensure an intuitive user interface and robust data processing. The integration of *IBM Watson NLP* enables the analysis of users' textual inputs and provides feedback on their emotional states. This paper provides an overview of all phases of application development, from technical aspects to the evaluation of the final product, and suggests directions for further development.

Keywords: Full stack application, emotional journal, IBM Watson Natural Language Processor, emotional state analysis, mental health, HTML, CSS, Angular, Django REST Framework, SQLite, natural language, artificial intelligence