

Web aplikacija za deskriptivnu i inferencijalnu analizu podataka iz empirijskih HCI istraživanja

Grenko, Fran

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:371761>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-12-25**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni diplomski studij računarstva

Diplomski rad

**Web aplikacija za deskriptivnu i
inferencijalnu analizu podataka iz
empirijskih HCI istraživanja**

Rijeka, studeni 2024.

Fran Grenko
0069085585

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni diplomski studij računarstva

Diplomski rad

**Web aplikacija za deskriptivnu i
inferencijalnu analizu podataka iz
empirijskih HCI istraživanja**

Mentor: prof. dr. sc. Sandi Ljubić

Rijeka, studeni 2024.

Fran Grenko
0069085585

Rijeka, 06.03.2024.

Zavod: Zavod za računarstvo
Predmet: Objektno orijentirano programiranje

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Fran Grenko (0069085585)**
Studij: Sveučilišni diplomski studij računarstva (1400)
Modul: Programsko inženjerstvo (1441)

Zadatak: **Web aplikacija za deskriptivnu i inferencijalnu analizu podataka iz empirijskih HCI istraživanja / A Web Application for Descriptive and Inferential Analysis of Data from Empirical HCI Research**

Opis zadatka:

U području interakcije čovjeka i računala, različita sučelja i modaliteti interakcije se uobičajeno vrednuju provođenjem odgovarajućeg eksperimenta. Pri tome, izlazni podaci iz takvog eksperimenta predstavljaju osnovu za provođenje detaljne statističke analize. U radu je potrebno razviti web aplikaciju za obradu postojećih dnevnčkih zapisa iz eksperimentalnih HCI istraživanja. Aplikacija mora omogućiti: (1) deskriptivnu analizu podataka zasnovanu na interaktivnom konfiguratoru grafičkog prikaza, (2) inferencijalnu analizu podataka provođenjem specificiranih statističkih testova, te (3) višekorisnički kontekst - upravljanje korisnicima, arhivom/bazom spremljenih statističkih analiza te pristupom relevantnim podacima.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Mentor:
prof. dr. sc. Sandi Ljubić

Predsjednik povjerenstva za
diplomski ispit:
prof. dr. sc. Miroslav Joler

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, studeni 2024.

Fran Grenko

Zahvala

Zahvaljujem svojoj djevojci na neizmornoj podršci tijekom studiranja i pisanja ovog rada. Također posebna zahvala i mojoj obitelji koja me kroz cijeli život nesebično motivirala i bila uz mene u svakom koraku.

Sadržaj

Popis slika	viii
Popis tablica	ix
1 Uvod	1
2 Empirijska HCI istraživanja	3
2.1 Empirijski pristup u području HCI	3
2.1.1 Metoda opažanja	4
2.1.2 Korelacijska metoda	5
2.1.3 Eksperimentalna metoda	6
2.2 Deskriptivna i inferencijalna statistika	7
2.2.1 Deskriptivna statistika	7
2.2.2 Inferencijalna statistika	11
3 Razvoj Web aplikacije za deskriptivnu i inferencijalnu analizu po- dataka iz empirijskih HCI istraživanja	19
3.1 Funkcionalni zahtjevi i arhitektura aplikacije	19
3.2 Tehnološki stog	21
3.2.1 TypeScript	21
3.2.2 Next.js i React	22
3.2.3 Python i Flask okvir	23
3.2.4 MongoDB	24
3.2.5 D3.js	24
3.2.6 Clerk	25

Sadržaj

3.3	Implementacija poslužiteljske strane sustava	26
3.3.1	Kolekcija <i>user</i>	27
3.3.2	Kolekcija <i>project</i>	28
3.3.3	Kolekcija <i>graph-options</i>	29
3.3.4	Kolekcija <i>axis</i>	30
3.3.5	Implementacija statističkih testova	30
3.4	Implementacija klijentske strane sustava	33
3.4.1	Komponenta <i>FileUpload</i>	33
3.4.2	Komponenta <i>Parser</i>	34
3.4.3	Komponenta <i>GraphOptions</i>	34
3.4.4	Komponenta <i>Graph</i>	35
3.4.5	Komponenta <i>Statistics</i>	38
4	Prikaz karakterističnih slučajeva korištenja	39
4.1	Prijava u sustav i upravljanje korisničkim podacima	40
4.2	Prikaz postojećih projekata	43
4.3	Stvaranje novog projekta	44
4.3.1	Učitavanje eksperimentalnih podataka	44
4.3.2	Opcije grafičkog prikaza	46
4.4	Pregled grafova i deskriptivne statistike	50
4.5	Statistički testovi i post-hoc analize	53
5	Zaključak	59
	Sažetak	62

Popis slika

4.1	Početna stranica aplikacije	40
4.2	Prozor za registraciju u sustav	41
4.3	Postavke osnovnih informacija	42
4.4	Sigurnosne postavke	42
4.5	Prikaz stvorenih projekata	44
4.6	Prikaz praznog obrasca za unos općenitih informacija o projektu . .	45
4.7	Primjer ispunjenog obrasca za unos općenitih informacija o projektu	46
4.8	Primjer ispunjenog obrasca o opcijama vizualnog prikaza za stupčasti graf	48
4.9	Primjer obavijesti o pogrešci na x-osi	48
4.10	Primjer ispunjenog obrasca o opcijama vizualnog prikaza za multilinijski graf	49
4.11	Primjer generiranog stupčastog grafa	50
4.12	Primjer generiranog kutijastog grafa	51
4.13	Generirana deskriptivna statistika za usporedbu tipkovnica	52
4.14	Primjer generiranog multilinijskog grafa	53
4.15	Primjer generiranog histogram grafa	55
4.16	Rezultati Wilcoxonovog testa za usporedbu učinkovitosti unosa dvije tipkovnice	56
4.17	Prikaz stupčastog grafa za usporedbu 3 tipkovnice	57
4.18	Prikaz rezultata Friedmanovog testa	57
4.19	Prikaz rezultata post hoc analize (međusobne usporedbe Wilcoxonovim testom)	58

Popis tablica

2.1	Usporedba metode opažanja, korelacijske i eksperimentalne metode [3]	7
-----	--	---

Poglavlje 1

Uvod

S neprestanim razvojem tehnologije i njezinom sve većom prisutnošću u svakodnevnom životu, razumijevanje načina na koji ljudi koriste računalne sustave postalo je od ključne važnosti. To omogućuje daljnje unaprjeđenje tih sustava kako bi bili što učinkovitiji i intuitivniji. Interakcija čovjeka i računala (*engl. Human-Computer Interaction, HCI*) predstavlja disciplinu koja proučava dizajn, evaluaciju i implementaciju interaktivnih računalnih sustava s naglaskom na potrebe i iskustva krajnjih korisnika. Cilj ove discipline je stvaranje sustava koji su ne samo funkcionalni, već i lako razumljivi te prilagođeni korisnicima.

HCI uključuje empirijska istraživanja usmjerena na optimizaciju korisničkog iskustva putem dizajnerskih i tehničkih poboljšanja. Takva istraživanja često generiraju veliku količinu podataka koji zahtijevaju detaljnu statističku analizu kako bi se otkrili obrasci, testirale hipoteze i potvrdili zaključci. Primjerice, istraživanje može ispitati kako različiti elementi web trgovine utječu na korisničko ponašanje ili kako brzina odziva aplikacije doprinosi korisničkom zadovoljstvu. Iako je moguće intuitivno prepoznati određene trendove, znanstveni pristup zahtijeva statističku obradu kako bi se utvrdilo jesu li opaženi rezultati ishod značajnih uzoraka ili samo slučajnosti. Inferencijalna statistika u ovom kontekstu pomaže nam donijeti pouzdane zaključke kroz testove koji kvantificiraju vjerojatnost da su razlike ili trendovi doista značajni.

Provođenje statističkih analiza u HCI istraživanjima može biti složen proces, koji uključuje prikupljanje i obradu podataka te korištenje raznih statističkih alata za interpretaciju rezultata. Osim toga, budući da istraživanja često uključuju više stručnjaka, suradnja među istraživačima postaje ključna, ali može biti otežana potrebom

Poglavlje 1. Uvod

za stalnim dijeljenjem podataka. Ovi izazovi ukazuju na potrebu za razvojem aplikacije koja omogućuje jednostavno izvođenje deskriptivne i inferencijalne statistike te učinkovitu suradnju među istraživačima, koji rade na zajedničkim projektima u kontekstu empirijskog pristupa HCI-u.

Cilj ovog diplomskog rada je razvoj aplikacije za provođenje deskriptivne i inferencijalne statistike prilagođene HCI empirijskim istraživanjima, kao i prikaz njezine praktične primjene u istraživačkom okruženju. U radu će biti detaljno opisani proces razvoja aplikacije, korištene tehnologije, teoretske osnove deskriptivne i inferencijalne statistike, kao i primjeri korištenja aplikacije.

Poglavlje 2

Empirijska HCI istraživanja

Empirijsko istraživanje temelji se na donošenju zaključaka isključivo na osnovu provjerljivih, objektivnih dokaza dobivenih iz opažanja, eksperimenata, ili mjerenja. Takvi dokazi zovu se empirijski (*grč. empirikos - iskustvo*) jer se temelje na stvarnim opažanjima, a ne na pretpostavki ili teoriji.

Empirijsko istraživanje može koristiti različite metodološke pristupe, pri čemu se najčešće razlikuju kvantitativne i kvalitativne metode. Kvantitativni pristupi u istraživanju temelje se na prikupljanju i analizi numeričkih podataka, što omogućuje mjerljivo i objektivno donošenje zaključaka u eksperimentu, koji se potom mogu statistički analizirati na širim populacijama [1].

U središtu empirijskog istraživanja uglavnom stoji postavljanje i testiranje hipoteza. Hipoteza je temeljna pretpostavka koja postavlja okvir istraživanja. Obično se testira nulta hipoteza, koja tvrdi da ne postoji razlika ili učinak među promatranim varijablama, nasuprot alternativnoj hipotezi koja predlaže suprotno i koju istraživač nastoji potvrditi. Statističko testiranje hipoteza omogućuje procjenu slučajnosti dobivenog rezultata ili je dovoljno značajan da se odbaci nulta hipoteza i podrži alternativna.

2.1 Empirijski pristup u području HCI

Interakcija čovjeka i računala je znanstvena disciplina čiji je cilj dizajnirati i evaluirati računalne sustave i uređaje, na način koji je intuitivan, učinkovit i prilagođen korisnicima, kako bi se omogućila što jednostavnija, ugodnija upotreba istih. Samo

Poglavlje 2. Empirijska HCI istraživanja

područje HCI-ja pojavilo se u ranim 1980-tima, potaknuto sve većom upotrebom računala u poslovne i privatne svrhe. S pojavom koncepata radne površine i grafičkog korisničkog sučelja, fokus se prebacio na poboljšanje upotrebljivosti računala za privatne korisnike, čineći ih intuitivnijima i pristupačnijima. Cilj je postao olakšati korisnicima interakciju s tehnologijom, uzimajući u obzir ne samo funkcionalnost sustava, već i cjelokupno korisničko iskustvo. Ovaj pristup omogućio je da se računala ne koriste samo za tehničke svrhe, već da postanu dio svakodnevnog života, prilagođena širem krugu korisnika s različitim potrebama i tehničkim znanjima.

Kako se područje HCI razvijalo, proširilo je svoje djelovanje i izvan same upotrebljivosti. Do 1990-ih, ovo područje počelo je naglašavati "ljudski" aspekt interakcije, uključujući pristupe dizajna usmjerenog na korisnika te se fokusirati na lako razumljiva radna okruženja. Ovo razdoblje obilježava rast participativnog dizajna, gdje su sustavi dizajnirani zajedno s korisnicima. Kasnije, u 2000-ima i nadalje, HCI se sve više usmjerava na dizajn raznolikih okruženja, uključujući emocije, kulturne razlike i nove tehnologije poput mobilnog računarstva i društvenih mreža [2].

HCI istraživanja trebaju biti praktična i relevantna, s jasnim fokusom na korisnike, organizacije ili dizajn sustava. Ključno je da ova istraživanja imaju stvaran utjecaj na oblikovanje korisničkih sučelja, optimizaciju razvojnih procesa ili obuku korisnika. Procjena korisničkog iskustva mora se temeljiti na rigoroznim istraživačkim postupcima, s jasno definiranim metodama evaluacije. To podrazumijeva ne samo analizu potreba korisnika, već i detaljno proučavanje interakcija sa sustavom, uključujući preciznost izvedbe, vrijeme potrebno za izvršenje zadataka i razinu zadovoljstva korisnika. Validnost istraživanja ključna je za osiguravanje vjerodostojnih rezultata [3].

Postoje tri glavna načina provođenja istraživanja u HCI-ju: metodom opažanja, eksperimentalnom metodom i korelacijskom metodom.

2.1.1 Metoda opažanja

Ova metoda se svodi na promatranje kako ljudi komuniciraju s računalima ili računalnim sustavima. Omogućuje analiziranje ponašanja korisnika u njihovom prirodnom okruženju, što doprinosi realnom razumijevanju događaja. Umjesto da se ponašanje ispitanika ispituje u umjetno stvorenim uvjetima, opažanje se provodi u

Poglavlje 2. Empirijska HCI istraživanja

kontekstu stvarnog svijeta, gdje se može zabilježiti autentično korisničko iskustvo. U okviru ove metode, najčešće se koriste različite istraživačke tehnike, uključujući fokus grupe, intervju, terenske studije i seminare, koji daju detaljniji uvid u sami događaj.

Iako rezultati ovih tehnika mogu biti iznimno korisni, oni često nisu lako mjerljivi. Opažanje se, kao kvalitativna metoda, fokusira više na značaj i kontekst događaja nego na precizna numerička mjerenja. No to ograničenje omogućuje znanstvenicima identificiranje ključnih obrazaca i trendove u ljudskom ponašanju.

Metoda opažanja posebno je korisna u proučavanju složenih ljudskih interakcija i emocija koje je teško kvantificirati. Kroz ovu metodu, istraživači mogu steći dublje razumijevanje korisničkih potreba, očekivanja i problema s kojima se susreću u svakodnevnoj upotrebi tehnologije. Također, opažanjem se mogu istražiti različiti aspekti korisničkog iskustva, kao što su frustracije, zadovoljstvo, motivacija i angažman [1].

2.1.2 Korelacijska metoda

Korelacijska metoda koristi se za istraživanje odnosa između dvije ili više varijabli. Ova metoda ključna je za utvrđivanje postoji li statistički značajna veza između varijabli, ali ne nudi uvid u uzrok korelacije. Na primjer, validno istraživanje moglo bi ispitivati povezanost vremena reakcije pametnih asistenata s razinom korisničkog zadovoljstva. Ako rezultati pokažu značajnu korelaciju, to znači da postoji veza između ta dva faktora, no iz tih podataka ne možemo zaključiti koji faktor uzrokuje promjene u drugome [4].

Podaci za korelacijska istraživanja mogu se prikupljati na različite načine, uključujući opažanja, intervju, ankete, upitnike ili specifična mjerenja. Često se korelacijska metoda kombinira s eksperimentalnim pristupima, osobito kada se koriste upitnici, kako bi se dodatno istražile ciljane varijable.

Iako je korelacijska metoda korisna za prepoznavanje obrazaca i poveznica, ključna je ograničenost u tome što ne može dokazati uzročnost. To znači da, iako dvije varijable mogu biti povezane, to ne znači da jedna nužno uzrokuje promjene u drugoj – primjerice, veza između duljeg vremena korištenja pametnih asistenata i nižeg zadovoljstva korisnika ne znači automatski da dulje vrijeme korištenja uzrokuje niže

zadovoljstvo.

Korelacijska metoda je posebno korisna u istraživanjima u kojima nije moguće provesti kontrolirane eksperimente. Korelacija može ukazati na smjer u kojem se buduća istraživanja trebaju kretati, ali za čvrste zaključke o uzročnosti potrebno je koristiti eksperimentalnu metodu.

2.1.3 Eksperimentalna metoda

Eksperimentalna ili znanstvena metoda se koristi za proučavanje hipoteza u strogo kontroliranim uvjetima kako bi se osigurala točnost i preciznost rezultata. Ova metoda je posebno učinkovita u HCI istraživanjima jer minimizira utjecaj vanjskih faktora koji bi mogli omesti rezultate u prirodnom okruženju, čime omogućuje preciznije donošenje zaključaka.

U svakom kontroliranom eksperimentu postoje barem dvije ključne varijable: nezavisna varijabla i zavisna varijabla. Nezavisna varijabla je ona koja se sustavno mijenja tijekom istraživanja. U HCI kontekstu, to može biti svojstvo sučelja ili tehnika interakcije koja se prezentira ispitanicima u različitim konfiguracijama. Zavisna varijabla, s druge strane, predstavlja mjerljivu karakteristiku ljudskog ponašanja, poput vremena potrebnog za dovršavanje zadatka, točnosti ili broja grešaka [4].

Eksperimentalna metoda omogućuje sustavno testiranje različitih konfiguracija sustava, pružajući uvid u to kako promjene u nezavisnoj varijabli utječu na zavisnu varijablu. Za razliku od testiranja upotrebljivosti, gdje se obično ispituje jedno sučelje bez manipulacije varijablama, eksperimentalna istraživanja omogućuju donošenje zaključaka o uzročno-posljedičnim odnosima među varijablama.

Iako je HCI eksperimentalna metodologija u velikoj mjeri inspirirana psihološkim istraživanjima, njezina je usmjerenost na specifične interakcije s tehnologijom, čime je prilagođena izazovima modernih korisničkih sustava. Takav pristup omogućuje istraživačima ne samo da analiziraju performanse korisnika već i da prikupe kvalitativne podatke o subjektivnim iskustvima, poput frustracije ili umora, koji mogu utjecati na cjelokupno korisničko iskustvo.

Međutim, iako eksperimentalna metoda donosi visoku razinu preciznosti i omogućuje zaključke o uzročnosti, umjetno laboratorijsko okruženje može smanjiti rele-

Poglavlje 2. Empirijska HCI istraživanja

Tip istraživanja	Ideja	Tvrdnja	Metode
metoda opažanja	opisati situaciju na temelju događaja	X se događa	opažanja, terenske studije, fokus grupe, intervjui
korelacijska metoda	pronaći poveznicu između više varijabli	X korelira s Y	opažanja, terenske studije, ankete
eksperimentalna metoda	pronaći uzrok događaja	X je razlog za Y	kontrolirani eksperimenti

Tablica 2.1 Usporedba metode opažanja, korelacijske i eksperimentalne metode [3]

vantnost rezultata u stvarnom svijetu.

U tablici 2.1 prikazana je usporedba triju metoda empirijskih istraživanja, zajedno s njihovim osnovnim idejama i načinima provedbe.

2.2 Deskriptivna i inferencijalna statistika

U ovom potpoglavlju razmatrat će se osnovni elementi deskriptivne i inferencijalne statistike. Deskriptivna statistika pruža alate za sažimanje i opisivanje podataka, uključujući mjere centralne tendencije, disperzije i grafičke prikaze koji pomažu u interpretaciji rezultata. Inferencijalna statistika, s druge strane, omogućuje donošenje zaključaka o populacijama na temelju uzoraka, koristeći metode poput testiranja hipoteza i izračuna intervala povjerenja. U daljnjem tekstu detaljnije ćemo obraditi obje statističke metode, koje zajedno omogućuju bolje razumijevanje podataka i donošenje informiranih odluka.

2.2.1 Deskriptivna statistika

Deskriptivna statistika je grana statistike koja se fokusira na opisivanje i analiziranje značajki skupa podataka, omogućujući razumijevanje podataka kroz pregledne i mjerljive numeričke informacije. Nakon prikupljanja podataka u istraživanju, prvi korak analize jest sažeti te podatke pomoću deskriptivne statistike. Ovim postupkom se omogućuje stjecanje osnovnih uvida u obrasce i trendove unutar podataka, čime se olakšava donošenje informiranih zaključaka i donošenje odluka.

Deskriptivna statistika dijeli se na tri glavne kategorije:

- **Distribucija** – pokazuje kako su podaci raspoređeni unutar skupa, omogućujući pregled frekvencije svake vrijednosti. Frekvencijskom distribucijom prikazujemo koliko se puta pojavljuje određena vrijednost, bilo brojevima ili postocima, što omogućuje uvid u obrazac podataka. Na primjer, distribucija frekvencije može otkriti dobne skupine s većim brojem ispitanika, kao što su maloljetnici u odnosu na punoljetne osobe ili pomoći nam zaključiti jesu li podaci normalno distribuirani.
- **Mjere centralne tendencije** koriste se za određivanje uobičajenih ili tipičnih vrijednosti unutar skupa podataka. Tri glavne mjere centralne tendencije su aritmetička sredina, medijan i mod. Aritmetička sredina, koja se izračunava kao zbroj svih vrijednosti podijeljen s brojem vrijednosti, korisna je za skupove podataka bez značajnih odstupanja. Međutim, osjetljivost na ekstremne vrijednosti može dovesti do iskrivljenih rezultata. Medijan, kao središnja vrijednost u uređenom skupu podataka, često je prikladniji u situacijama kada su prisutni značajni ekstremi, jer pruža stabilniju mjeru centralne tendencije. Mod, koji predstavlja najčešće prisutnu vrijednost u skupu podataka, koristan je za identifikaciju dominantnih ili uobičajenih vrijednosti. Uz ove tri mjere, važno je razmotriti i njihovu međusobnu usporedbu kako bi se odabrala najprikladnija mjera za specifičan skup podataka i svrhu analize [5].
- **Mjere disperzije** opisuju koliko su podaci u skupu rašireni i koliko odstupaju od srednje vrijednosti, omogućujući bolje razumijevanje stabilnosti i pouzdanosti podataka. Jedna od osnovnih mjera disperzije je raspon, koji predstavlja razliku između najveće i najmanje vrijednosti u skupu, dajući jednostavan uvid u ukupni raspon podataka. Standardna devijacija je jedna od najvažnijih mjera disperzije. Definira prosječnu udaljenost vrijednosti u skupu od aritmetičke sredine i ukazuje na raspršenost podataka. Veća standardna devijacija sugerira veću nestabilnost i manju koncentraciju podataka oko srednje vrijednosti. U skupu podataka s normalnom distribucijom, standardna devijacija pokazuje udaljenost većine podataka od aritmetičke sredine, gdje se 68% podataka nalazi unutar jedne standardne devijacije od srednje vrijednosti, 95% unutar dvije, a 99,7% unutar tri standardne devijacije. Ova svojstva korisna su u procjeni očekivanih vrijednosti i vjerojatnosti pojave podataka unutar određenih intervala.

Poglavlje 2. Empirijska HCI istraživanja

Formula za standardnu devijaciju je:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \quad (2.1)$$

gdje σ označava standardnu devijaciju, x_i i -tu vrijednost iz skupa, μ aritmetičku sredinu, a N veličinu skupa.

Još jedna korisna mjera disperzije su kvartili, koji dijele skup podataka na četiri jednaka dijela, pružajući uvid u raspodjelu podataka oko medijana. Prvi kvartil (Q1) označava vrijednost ispod koje se nalazi 25% podataka, medijan ili drugi kvartil (Q2) određuje središnju vrijednost skupa, a treći kvartil (Q3) pak označava vrijednost ispod koje se nalazi 75% podataka. Razlika između prvog i trećeg kvartila, poznata kao interkvartilni raspon (*engl. interquartile range - IQR*), predstavlja područje u kojem se nalazi srednjih 50% skupa podataka. Ova mjera je posebno korisna u grafičkoj deskriptivnoj statistici, poput kutijastog grafa, koji će u nastavku biti detaljnije objašnjen.

IQR se koristi za filtriranje odstupanja u skupovima podataka, smanjujući utjecaj ekstremnih vrijednosti zahvaljujući svom fokusu na sredinu distribucije. Međutim, usredotočenost na ovaj srednji dio može rezultirati gubitkom važnih informacija o varijanci podataka. Osim toga, IQR može biti manje pouzdan kod manjih uzoraka, a njegova primjena nije idealna za normalne raspodjele, gdje su druge mjere disperzije, poput standardne devijacije, često prikladnije. Zbog tih ograničenja, preporučljivo je koristiti IQR zajedno s drugim statističkim mjerama kako bi se dobio sveobuhvatan uvid u varijancu podataka [5].

Asimetrija (*engl. skewness*) opisuje stupanj u kojem distribucija podataka odstupa od simetrije. Asimetrija može biti:

- pozitivno asimetrična (desno nagnuta) - većina vrijednosti je koncentrirana na lijevom dijelu distribucije, što znači da je prosjek obično veći od medijana.
- negativno asimetrična (lijevo nagnuta) - većina vrijednosti je koncentrirana na desnom dijelu distribucije, što znači da je prosjek obično manji od medijana.

Asimetrija je korisna u identifikaciji prisutnosti ekstremnih vrijednosti, koje

Poglavlje 2. Empirijska HCI istraživanja

uvelike utječu na disperziju podataka.

Kako bi se potkrijepile varijable deskriptivne statistike, često se u kombinaciji s njima koriste i grafovi. Za grafički prikaz distribucije frekvencije koristi se **histogram**, koji vizualizira opažanja na x-osi u obliku stupaca. Visina svakog stupca predstavlja broj podataka unutar određenog raspona vrijednosti. Na primjer, histogram može prikazivati broj godina sudionika u istraživanju, pružajući uvid u raspodjelu podataka. Histogram je koristan alat za analizu distribucije, omogućuje uočavanje simetričnosti i asimetričnosti podataka te identifikaciju potencijalnih odstupanja ili anomalija. Odabir veličine intervala značajno utječe na oblik histograma. Premali intervali mogu uzrokovati "šum" u grafu, otežavajući prepoznavanje stvarne distribucije podataka, dok preveliki intervali mogu prikriti važne detalje. Stoga je ključno pažljivo odabrati veličinu intervala kako bi se postiglo bolje razumijevanje strukture i obrazaca u skupu podataka.

Linijski graf se koristi za prikaz podataka koji se mijenjaju kroz kontinuirane intervale, najčešće u vremenskom okviru. Vrijednosti su prikazane na y-osi, dok je na x-osi obično prikazan vremenski interval. Svaka točka na grafu predstavlja vrijednost u određenom trenutku, a povezivanjem tih točaka linijama prikazuje se trend promjene. Ovaj graf je vrlo koristan u financijskim analizama, poput praćenja promjena cijena ili kretanja dionica.

Jedna od ključnih prednosti linijskog grafa je mogućnost prikazivanja višestrukih serija podataka unutar istog grafikona, što se naziva multilinijski graf. Na taj se način omogućava usporedba različitih trendova u istom vremenskom okviru. Ovo je naročito korisno u analizi podataka koji pripadaju različitim kategorijama, ali imaju zajedničke vremenske varijable. Linijski i multilinijski grafovi pružaju jednostavan uvid u obrasce poput rasta, pada ili stabilnosti kroz vrijeme, čime olakšavaju donošenje zaključaka na temelju promjena u podacima [6].

Stupčasti grafovi su korisni za usporedbu aritmetičkih sredina različitih kategorija, ističući razlike među njima u odnosu na određenu numeričku vrijednost. Visina svakog stupca predstavlja vrijednost koju kategorija postiže, omogućujući usporedbu kvalitativnih (kategorijskih) podataka s kvantitativnima. Y-os najčešće prikazuje aritmetičku sredinu vrijednosti unutar svake kategorije, čime se postiže jasna vizualizacija relativnih razlika među skupinama. Varijanca u podacima može se prikazati dodavanjem intervala odstupanja, odnosno prikazom standardne devi-

Poglavlje 2. Empirijska HCI istraživanja

jacije za svaku kategoriju. Ove linije, koje se protežu iznad i ispod vrha stupca, predstavljaju vrijednosti standardne devijacije, dajući dodatni uvid u raspon očekivanih varijacija unutar podataka. Takav prikaz omogućava lakše razumijevanje stabilnosti i pouzdanosti podataka te pomaže u ocjeni razlika među skupinama [6].

Kutijasti graf (*engl. boxplot, box-and-whiskers*) omogućava grafički prikaz raspodjele i raspršenja podataka te olakšava prepoznavanje odstupajućih vrijednosti. Kutijasti dio grafa prikazuje interkvartilni raspon, odnosno razmak između prvog i trećeg kvartila, gdje se nalazi središnjih 50% podataka. Medijan unutar kutijastog dijela grafa označava središnju vrijednost skupa podataka, a njegova pozicija može otkriti asimetriju u raspodjeli podataka. Ako je medijan bliže prvom kvartilu, podaci su više koncentrirani u nižim vrijednostima, dok je suprotno ako se nalazi bliže trećem kvartilu. Od kutijastog dijela grafa se protežu linije nazvane "brkovima" (*engl. whiskers*), koje prikazuju najbliže vrijednosti koje nisu dalje od 1.5 puta interkvartilnog raspona od kvartila. Uz pomoć ovih elemenata dobivaju se dodatne informacije o širini distribucije podataka i pomažu u pregledu simetrije, raspršenja i prisutnosti ekstremnih vrijednosti. Kutijasti graf često se koristi za usporedbu skupova podataka, primjerice rezultata testova između grupa ili prihoda u različitim regijama, čime omogućava brzi pregled simetrije i raspršenja. Prednost kutijastog grafa je jasno prikazivanje varijabilnosti unutar podataka, dok ograničenja uključuju njegovu smanjenu učinkovitost kod manjih uzoraka i nemogućnost prikazivanja detalja o gustoći podataka unutar kutije. U normalno raspoređenim podacima učinkovito prikazuje simetriju, dok kod asimetrične raspodjele to naglašava razmakom medijana i rubova interkvartilnog raspona [5].

2.2.2 Inferencijalna statistika

Inferencijalna statistika predstavlja granu statistike usmjerenu na donošenje zaključaka o populacijama na temelju njihovih uzoraka. Korištenje uzoraka omogućuje procjenu karakteristika populacije, testiranje hipoteza te generiranje pretpostavki. Za razliku od deskriptivne statistike, koja se bavi opisivanjem i sažimanjem podataka, inferencijalna statistika koristi uzorke za procjenu obilježja populacije ili za ispitivanje hipoteza o njoj, čime se dolazi do uvida o populaciji bez obrade svih podataka. Osnovne metode inferencijalne statistike uključuju intervale pouzdanosti i

Poglavlje 2. Empirijska HCI istraživanja

testove značajnosti, čime se analizira pouzdanost procjene i provjera hipoteza [7].

Premda inferencijalna statistika nikada ne može sa sigurnošću potvrditi alternativnu hipotezu, omogućuje procjenu vjerojatnosti odbacivanja nulte hipoteze. Ključni koncept u ovom procesu je **p-vrijednost**, koja predstavlja vjerojatnost dobivanja rezultata jednakih ili ekstremnijih od opaženih pod pretpostavkom da je nulta hipoteza istinita. Ako je p-vrijednost niža od postavljenog praga značajnosti (obično 0.05), podaci sugeriraju značajno odstupanje od nulte hipoteze, pa se ona odbacuje u korist alternativne. P-vrijednost ovisi o odabranom statističkom testu i predstavlja ključnu varijablu za procjenu razlike među skupinama.

Kako bi se dodatno razumjela preciznost statističkog modela, analizira se i njegova sposobnost prikazivanja varijacija u podacima. Varijacije se dijele na sistematske i nesistematske. Sistematska varijacija odnosi se na objašnjive razlike koje proizlaze iz istraživanog efekta, dok nesistematska varijacija označava slučajne promjene koje model ne može objasniti. Omjer između ove dvije varijacije naziva se **testna statistika** i koristi se za mjerenje uspješnosti modela u objašnjavanju podataka:

$$\text{testna statistika} = \frac{\text{sistematska varijanca}}{\text{nesistematska varijanca}} = \frac{\text{efekt}}{\text{pogreška}} \quad (2.2)$$

Testna statistika pruža uvid u pouzdanost modela, omogućujući istraživačima procjenu koliko dobro podaci podržavaju postavljenu hipotezu i koliko su rezultati značajni za donesene zaključke.

Da bi se odabrao odgovarajući statistički test za analizu podataka iz promatrane populacije, važno je razumjeti njene specifične karakteristike. Osnovna podjela populacija može se vršiti na nezavisne i povezane skupine. Nezavisne skupine odnose se na situacije gdje se uspoređuju različite grupe ispitanika koje nisu međusobno povezane. Ovaj tip analize često se koristi u istraživanjima koja uključuju različite intervencije ili tretmane. Na primjer, ako se istražuje učinkovitost dvaju različitih lijekova, dva odvojena uzorka pacijenata mogu se koristiti za testiranje svakog lijeka, čime se eliminira potencijalni utjecaj jedne grupe na drugu. Ova vrsta studija omogućuje istraživačima da dobiju jasnu sliku o učincima tretmana bez konfuzije uzrokovane interakcijama između sudionika.

S druge strane, povezane skupine uključuju situacije u kojima su ispitanici na neki način povezani ili su mjereni u različitim uvjetima. Na primjer, u kliničkim

Poglavlje 2. Empirijska HCI istraživanja

ispitivanjima često se prikupljaju podaci o istim pacijentima prije i nakon terapije. Time se omogućuje analiziranje promjene unutar iste grupe, što može pružiti snažnije dokaze o učincima tretmana. Korištenje povezanih skupina omogućava kontrolu varijance između ispitanika, što može rezultirati većom snagom statističkog testa.

Osim razlike između nezavisnih i povezanih skupina, statistički testovi mogu se klasificirati kao parametarski i neparametarski. Parametarski testovi temelje se na određenim pretpostavkama o distribuciji podataka, poput normalne distribucije, te o homogenosti varijanci među grupama. Kada su ti uvjeti ispunjeni, parametarski testovi obično pružaju preciznije i pouzdanije rezultate u usporedbi s neparametarskim testovima.

Neparametarski testovi ne zahtijevaju stroge pretpostavke o distribuciji podataka. Oni su posebno korisni u situacijama kada je veličina uzorka mala, kada je potrebna otpornost na ekstremne vrijednosti ili kada podaci nisu normalno distribuirani. Zbog svoje fleksibilnosti, neparametarski testovi se često koriste u analizi podataka koja ne ispunjava uvjete za primjenu parametarskih testova, omogućujući donošenje validnih zaključaka čak i u složenijim situacijama. Statistički testovi dijele se i na temelju koliko skupina uspoređuju, te tako mogu uspoređivati dvije skupine, ali i više njih.

U nastavku će biti objašnjeni neki od najčešće korištenih statističkih testova.

T-test

T-test je parametarski statistički test koji služi za usporedbu aritmetičkih sredina dvaju skupova podataka. Pretpostavlja normalnu distribuciju i homogenost varijanci podataka. Postoje dvije glavne vrste t-testa:

- T-test za nezavisne uzorke koristi se za usporedbu srednjih vrijednosti dvije različite populacije, npr. uspoređujući rezultate dvije grupe ispitanika.
- T-test za zavisne uzorke (ili upareni t-test) primjenjuje se na iste ispitanike u različitim uvjetima ili na parove podataka unutar jedne grupe, primjerice prije i nakon intervencije.

T-test je fleksibilan alat jer nudi različite verzije prilagođene raznim istraživačkim situacijama. Rezultat t-testa uključuje dvije ključne vrijednosti: testnu statistiku i p-vrijednost. U nezavisnom t-testu, testna statistika se dobiva dijeljenjem razlike

Poglavlje 2. Empirijska HCI istraživanja

između srednjih vrijednosti dviju skupina s procijenjenom standardnom pogreškom razlike. U zavisnom t-testu, testna statistika računa se dijeljenjem prosječne razlike između mjerenja s odgovarajućom standardnom devijacijom razlike [6].

Wilcoxonov test

Wilcoxonov test (Wilcoxon signed-rank test) je neparametarski test koji uspoređuje razlike dva povezana skupa podataka. Posebno je koristan kada podaci ne zadovoljavaju uvjete normalne distribucije, što ga čini odličnom alternativom zavisnom t-testu.

Za razliku od parametarskih testova, Wilcoxonov test koristi rangove apsolutnih razlika između parova mjerenja, bilježeći znak (pozitivan ili negativan) svakog ranga ovisno o smjeru razlike. Time se test usredotočuje na smjer i veličinu razlika, a ne na same numeričke vrijednosti. Izračun testne statistike temelji se na zbroju rangova za pozitivne i negativne razlike, čime se procjenjuje je li medijan razlika značajno različit od nule. Dodatno, test generira p-vrijednost koja omogućuje zaključak o statističkoj značajnosti dobivenih rezultata.

Mann-Whitneyjev test

Mann-Whitneyjev test je neparametarski test koji uspoređuje dva nezavisna skupa podataka. Često se koristi kao alternativa t-testu za nezavisne uzorke kada podaci ne zadovoljavaju pretpostavke normalne distribucije. Umjesto usporedbe aritmetičkih sredina, ovaj test se temelji na rangovima vrijednosti.

Za izvođenje testa najprije se rangiraju sve vrijednosti iz oba skupa podataka zajedno, od najmanje do najveće. U slučaju ponavljanja vrijednosti, prosječni rang koristi se za sve duplikate. Nakon rangiranja, rangovi se zbrajaju za svaki od uzoraka i na temelju dobivenih suma računa se vrijednost testa U_i čija formula glasi:

$$U_i = n_1 * n_2 + \frac{n_i * (n_i + 1)}{2} - T_i \quad (2.3)$$

gdje n_i označava broj elemenata u odgovarajućoj skupini, a T_i označava sumu rangova skupine. Manja vrijednost između U_1 i U_2 koristi se kao konačna vrijednost U-statistike. Na temelju ove U-statistike, zajedno s veličinama uzoraka, određuje

Poglavlje 2. Empirijska HCI istraživanja

se p-vrijednost koja pokazuje statističku značajnost razlike između skupina. Također, mnogi softverski alati izračunavaju z-vrijednost (normaliziranu vrijednost) koja omogućava lakšu interpretaciju rezultata, osobito kod većih uzoraka.

Analiza varijance

Analiza varijance (ANOVA, *engl. Analysis of Variance*) je parametarska statistička metoda za ispitivanje značajnosti razlika između aritmetičkih sredina triju ili više skupina podataka. Temelj analize varijance je usporedba varijanci unutar pojedinih grupa i između njih kako bi se utvrdilo je li varijacija između srednjih vrijednosti statistički značajna. Postoji nekoliko vrsta analize varijance, uključujući jednosmjernu analizu varijance (*one-way ANOVA*), analizu varijance ponovljenih mjerenja (*ANOVA repeated measures*), višesmjernu analizu varijance (*multi-way ANOVA*), multivarijantnu analizu varijance (*MANOVA*) i analizu varijance s kontrolom kovarijanci (*MANCOVA*). Ovaj tekst će se fokusirati na jednosmjernu analizu varijance i analizu varijance ponovljenih mjerenja.

Jednosmjerna analiza varijance uspoređuje aritmetičke sredine više nezavisnih grupa u odnosu na jedan faktor. Glavna pretpostavka je da varijance između grupa i unutar grupa održavaju uzorak populacije. Nulta hipoteza navodi da nema značajnih razlika među srednjim vrijednostima grupa, dok alternativna hipoteza sugerira postojanje značajnih razlika. Jednosmjerna analiza varijance analizira varijancu unutar svake grupe te varijancu između grupa kako bi se izračunala F-statistika. Ako je F-statistika značajna, to ukazuje na to da se barem jedna aritmetička sredina razlikuje od ostalih, što sugerira potrebu za daljnjim *post hoc* analizama kako bi se identificirale specifične razlike među grupama [4]. Za provođenje analize varijance moraju biti ispunjene određene pretpostavke:

- Nezavisnost mjerenja: Grupe moraju biti odvojene i nezavisne
- Homogenost varijanci: Varijance između grupa trebaju biti približno jednake
- Normalna distribucija: Podaci unutar svake grupe trebaju biti normalno distribuirani.

Primjer jednosmjerne analize varijance bi bilo istraživanje tri različite terapije kod pacijenata i praćenje postoje li značajne razlike u promjeni nekog parametra.

Poglavlje 2. Empirijska HCI istraživanja

Analiza varijance ponovljenih mjerenja je verzija jednosmjerne analize varijance kada su podaci prikupljeni od istih ispitanika pod različitim uvjetima ili u različitom vremenu, što znači da su zavisni. Ova metoda analizira varijancu unutar pojedinaca kako bi se utvrdilo postoje li značajne promjene u srednjim vrijednostima kroz različite uvjete. Slično jednosmjernoj analizi varijance, analiza varijance ponovljenih mjerenja koristi F-statistiku za testiranje hipoteze, a također može uključivati i analizu interakcija između faktora ako se ispituje više od jednog faktora. Pretpostavke kod testa analize varijance ponovljenih mjerenja su sljedeći:

- Normalna distribucija: Podaci unutar svake grupe trebaju slijediti normalnu distribuciju.
- Homogenost varijanci: Varijance između grupa trebaju biti približno jednake
- Homogenost kovarijanci¹: Kovarijance između grupa trebaju biti približno jednake
- Podaci ne smiju sadržavati značajne ekstreme

Primjer analize varijance ponovljenih mjerenja je utjecaj dijetalnog režima na kilažu kod istih ispitanika tokom nekog vremena.

Friedmanov test

Friedmanov test je neparametarski test koji se koristi za ispitivanje razlika između više povezanih skupina podataka. Često se primjenjuje kao alternativa analizi varijance ponovljenih mjerenja kada podaci ne zadovoljavaju pretpostavke normalne distribucije ili homogenost varijance. Ovaj test je posebno prikladan za istraživanja u kojima se mjerenja provode na istim ispitanicima kroz različite uvjete, kao što su različiti vremenski intervali ili različiti tretmani. Primjeri uključuju istraživanja koja uspoređuju učinke nekoliko vrsta terapija na istim sudionicima ili utjecaj različitih dijeta tijekom vremena.

Friedmanov test analizira rangove umjesto stvarnih vrijednosti mjerenja. Podaci unutar svakog uvjeta se rangiraju, a razlike u rangovima između uvjeta analiziraju kako bi se utvrdilo postoji li statistički značajna razlika između medijana. Nulta hipoteza Friedmanovog testa pretpostavlja da nema razlika između grupa, tj. da su

¹Promjena jedne varijable treba pratiti proporcionalnu promjenu druge varijable.

Poglavlje 2. Empirijska HCI istraživanja

mjerenja pod različitim uvjetima slična. Alternativna hipoteza predlaže postojanje barem jedne razlike u rangovima između uvjeta.

Ako test pokazuje značajne razlike, potrebno je provesti *post hoc* testove (kao što je Wilcoxonov test) kako bi se identificiralo koji su specifični parovi uvjeta različiti. Friedmanov test se zbog svoje fleksibilnosti u analizama ponovljenih mjerenja često koristi u raznim područjima znanosti.

***Post hoc* testovi**

Post hoc testovi se primjenjuju nakon utvrđivanja značajne razlike među grupama putem statističkog testa koji analizira tri ili više skupina. Njihov cilj je identificirati specifične parove grupa koje se međusobno značajno razlikuju u smislu mjerene varijable.

- Tukeyev test uspoređuje sve moguće parove srednjih vrijednosti među grupama. Ključna značajka Tukeyevog testa je očuvanje ukupne stope pogreške prvog tipa² na unaprijed definiranoj razini značajnosti. Proces počinje izračunavanjem kritične razlike (HSD), koja se temelji na broju grupa, veličini uzorka i standardnoj devijaciji. HSD se zatim koristi za usporedbu svih parova grupa, a ako razlika u srednjim vrijednostima između bilo koje dvije grupe premašuje HSD, ta razlika se smatra statistički značajnom. Tukeyev test je posebno prilagođen za jednosmjernu analizu varijance, ali se uz određene modifikacije može primijeniti i nakon drugih statističkih testova [4].
- Bonferronijeva korekcija smanjuje rizik od lažno pozitivnih rezultata prilikom višestrukih usporedbi. U situacijama kada se uspoređuje više elemenata, postoji mogućnost da se slučajno dobiju statistički značajne razlike. Bonferronijeva korekcija to rješava podešavanjem praga značajnosti. Postupak započinje dijeljenjem razine značajnosti s brojem usporedbi; na primjer, ako je razina značajnosti 0,05, a provodi se pet međusobnih usporedbi, prag značajnosti za svaku usporedbu iznosi $0,05 / 5 = 0,01$. Prednost Bonferronijeve korekcije je u tome što se može koristiti nakon bilo koje analize višestrukih grupa, osobito kada je važno smanjiti rizik od lažno pozitivnih rezultata [6].

²Pogreška prvog tipa označava lažno pozitivne rezultate, odnosno odbacivanje nulte hipoteze koja je zapravo istinita.

Poglavlje 2. Empirijska HCI istraživanja

- Wilcoxonov test je neparametarska metoda za post hoc analizu koja omogućuje usporedbu parova grupa kada podaci ne zadovoljavaju pretpostavke parametarskih testova, poput normalne distribucije. Wilcoxonov test prikladan je kao post hoc test nakon pozitivnog ishoda Friedmanovog testa.

Poglavlje 3

Razvoj Web aplikacije za deskriptivnu i inferencijalnu analizu podataka iz empirijskih HCI istraživanja

U ovom poglavlju razmotrit će se ključni aspekti razvoja web aplikacije koja omogućuje deskriptivnu i inferencijalnu analizu podataka iz empirijskih HCI istraživanja. Aplikacija je dizajnirana kako bi se omogućilo jednostavno prikupljanje, analiza i vizualizacija podataka, kao i izvođenje statističkih testova. U nastavku će se detaljno opisati funkcionalni zahtjevi i arhitektura aplikacije, odabrani tehnološki stog, te implementacija poslužiteljske i klijentske strane sustava.

3.1 Funkcionalni zahtjevi i arhitektura aplikacije

U ovom potpoglavlju navest će se glavni funkcionalni zahtjevi i arhitektura web aplikacije za deskriptivnu i inferencijalnu analizu podataka iz empirijskih istraživanja u području HCI. Glavni funkcionalni zahtjevi su:

- **Unos eksperimentalnih podataka** - korisnicima je omogućen unos podataka u obliku CSV (*engl. comma-separated values*) datoteka. Svaka datoteka mora sadržavati zaglavlje u prvom redu, koje sadrži nazive stupaca, dok svaki sljedeći redak predstavlja pojedinačan unos podataka. Prilikom učitavanja datoteke, korisniku će biti omogućeno specificirati znak koji se koristi kao graničnik (*engl. delimiter*) kako bi aplikacija mogla pravilno interpretirati i obraditi podatke iz datoteke.
- **Deskriptivna statistika** - aplikacija omogućuje automatsku generaciju os-

Poglavlje 3. Razvoj Web aplikacije za deskriptivnu i inferencijalnu analizu podataka

novnih deskriptivnih pokazatelja kao što su: srednja vrijednost, medijan, standardna devijacija, varijanca, raspon, kvartili, minimum i maksimum.

Korisnici mogu odabrati varijable za analizu i pregledati rezultate u realnom vremenu, uz vizualizaciju u obliku histograma, kutijastih, stupčastih i linijskih grafova.

- **Inferencijalna statistika** - aplikacija pruža mogućnost primjene različitih statističkih testova nad odabranim varijablama, uključujući: t-test za nezavisne uzorke, t-test za zavisne uzorke, Wilcoxonov test, Mann-Whitneyjev test, jednosmjernu analizu varijance, analizu varijance s ponovljenim mjerenjima, Friedmanov test i post hoc testove
- **Upravljanje korisničkim podacima i projektima** - Aplikacija podržava višekorisnički sustav u kojem korisnici mogu stvarati projekte i dodavati sudionike s kojima dijele pristup podacima i analizama.

Aplikacija je razvijena korištenjem tehnologija koje pokrivaju klijentsku i poslužiteljsku stranu. Na klijentskoj i poslužiteljskoj strani korišten je Next.js, napredni radni okvir temeljen na Reactu, popularnoj biblioteci za izradu korisničkih sučelja, a podržava programske jezike TypeScript i JavaScript. Pomoću ovog okvira razvijena je većina aplikacije, a povezana je s MongoDB bazom podataka. MongoDB je NoSQL baza podataka, što znači da koristi nerelacijski model za pohranu podataka, a podaci se pohranjuju u obliku dokumenata sličnim JSON-u (*engl. JavaScript Object Notation*). Upravljanje bazom podataka i pregled podataka mogući su putem platforme u oblaku MongoDB Atlas. Također, aplikacija koristi Mongoose, popularnu ORM (*engl. Object-Relational Mapping*) knjižnicu za MongoDB, koja olakšava rad s podacima i bazama podataka kroz definiranje shema i modela. Za dizajn korisničkog sučelja korišten je Tailwind CSS, uslužni okvir za stiliziranje koji omogućava brz i efikasan proces stvaranja dizajna korištenjem unaprijed definiranih klasa. Na poslužiteljskoj strani kao aplikacijsko programsko sučelje (*engl. API*) korišten je Flask, okvir baziran na Python programskom jeziku, čija je glavna zadaća izračun inferencijalne statistike u obliku statističkih testova. Aplikacija koristi knjižnicu D3.js, JavaScript alat za stvaranje dinamičkih i interaktivnih vizualizacija podataka. Ova knjižnica je korištena za vizualizaciju podataka u obliku grafova. Za potrebe autentifikacije korišteno je aplikacijsko programsko sučelje Clerk, koje podržava razne metode pri-

jave, uključujući prijavu putem lozinke i e-maila, kao i autentifikaciju preko vanjskih platformi, čime se pojednostavljuje proces prijave u aplikaciju.

3.2 Tehnološki stog

U ovom dijelu detaljno će se prikazati korišteni tehnološki stog, čiji je odabir bio ključan za postizanje stabilnosti, skalabilnosti i jednostavnosti aplikacije. Objasnit će se alati i okviri korišteni za razvoj klijentske i poslužiteljske strane, kao i izbor baze podataka i rješenja za autentifikaciju. Svaka od ovih komponenti ima specifičnu ulogu u osiguravanju učinkovite obrade podataka i provođenju statističkih analiza, što je temelj za postizanje visoke razine funkcionalnosti i korisničkog iskustva unutar sustava.

3.2.1 TypeScript

TypeScript je napredan alat koji se koristi u razvoju modernih web aplikacija. Razvijen je kao nadogradnja s JavaScripta s ciljem unaprijeđenja skalabilnosti i održavanja koda. Glavna prednost TypeScripta leži u korištenju statičkih tipova, čime se smanjuje broj pogrešaka tijekom razvoja. Zahvaljujući ovoj funkcionalnosti, pogreške se mogu otkriti već tijekom faze prevođenja koda, prije nego što se aplikacija pokrene, što značajno poboljšava pouzdanost i stabilnost aplikacije.

Statička tipizacija osim smanjenju pogrešaka, doprinosi i većoj produktivnosti i lakšem održavanju koda. Programerima omogućuje bolju preglednost i predvidljivost ponašanja programa. Istraživanja pokazuju da programski jezici sa statičkom tipizacijom mogu smanjiti broj pogrešaka i ubrzati identifikaciju problema u čak 59,5% slučajeva u usporedbi s dinamički tipiziranim jezicima [8].

TypeScript je razvijen od strane Microsofta 2012. godine, a njegova popularnost brzo je porasla među web programerima zbog jednostavne integracije s JavaScriptom i široke primjene u projektima svih veličina. Posebno je koristan u većim timovima i projektima gdje je važno imati dobro strukturiran, skalabilan kod s jasnim definiranim tipovima podataka.

3.2.2 Next.js i React

React je JavaScript knjižnica za izradu korisničkog sučelja, kojeg su 2013. razvili inženjeri Facebook-a. Jedna je od najpopularnijih knjižnica za JavaScript. Glavni koncept Reacta je komponentni pristup, gdje je korisničko sučelje podijeljeno u male, ponovno iskoristive komponente koje mogu imati vlastito stanje i logiku. Glavne značajke React-a su:

- Virtualni model objekta dokumenta (*engl. Document Object Model - DOM*) - umjesto direktnog rada s DOM-om, React koristi virtualni, što omogućava brže prikazivanje promjena na korisničkom sučelju. React prvo ažurira virtualni DOM te ga uspoređuje s prethodnim stanjem i potom minimalno ažurira stvarni DOM gdje je to potrebno
- Jednosmjerni tok podataka - podaci se prenose od roditeljske komponente prema dolje, čime se omogućuje precizna kontrola nad komponentama i poboljšava preglednost koda
- Hook - React uvodi "udice", posebne funkcije koje omogućuju rad sa stanjem komponentata bez potrebe za klasama. Najpopularniji su `useState`, koji se koristi za upravljanjem stanja varijabli koje se pojavljuju u više komponentata, i `useEffect`, funkcije koja se poziva svaki puta kada se promijeni stanje nekog od argumenata [9].

React omogućava fleksibilnost u razvoju aplikacija jer ga je moguće koristiti samostalno ili u kombinaciji s drugim alatima i bibliotekama. Ipak, sam po sebi, React nije "full-stack" alat, jer pokriva isključivo klijentsku stranu aplikacije. Tu dolazi do izražaja Next.js, koji nadograđuje React i proširuje njegovu funkcionalnost.

Next.js je napredni radni okvir temeljen na Reactu, koji je stvorila tvrtka Vercel s ciljem pojednostavljanja razvoja React web aplikacija. Danas se ubraja među vodeće okvire za izradu modernih web platformi. Pruža cjelovito rješenje za izradu dinamičkih i statičkih web stranica, spajajući fleksibilnost Reacta s dodatnim mogućnostima generiranja prikaza, optimizacije performansi i upravljanje logikom s poslužiteljske strane. Jedna od glavnih prednosti Next.js-a je podrška za generiranje stranica na strani poslužitelja (*engl. Server-Side Rendering - SSR*), što rezultira bržim vremenima učitavanjima i boljom optimizacijom za tražilice (*engl. Search En-*

gine *Optimization - SEO*). Budući da se sadržaj generira na poslužitelju u obliku statičkog HTML-a, tražilice ga lakše indeksiraju u usporedbi s tradicionalnim React aplikacijama. SSR je posebno koristan za aplikacije s dinamičkim sadržajem, gdje je važno da podaci budu ažurirani i dostupni u stvarnom vremenu. Pored SSR-a, Next.js podržava i statičko generiranje stranica tijekom izgradnje aplikacije, što omogućuje vrlo brzo učitavanje za stranice s minimalnim dinamičkim sadržajem, poput dokumentacija ili blogova. Također nudi i klijentsko generiranje prikaza, slično kao React, pružajući programerima slobodu odabira tehnologije koja najbolje odgovara projektu. Kao okvir koji pokriva klijentsku i poslužiteljsku stranu, Next.js omogućuje definiranje API krajnjih točaka (*engl. endpoints*) bez potrebe za dodatnim poslužiteljskim sustavima. Dodatno, automatsko generiranje ruta prema strukturi datoteka i direktorija čini razvoj i održavanje aplikacija jednostavnijim i bržim [10].

3.2.3 Python i Flask okvir

Python je jedan od najpopularnijih programskih jezika, poznat po jednostavnoj sintaksi i širokoj primjeni u različitim područjima, od razvoja web aplikacija do znanstvenih istraživanja i analize podataka. Zahvaljujući bogatom ekosustavu biblioteka kao što su NumPy, SciPy, Pandas i Statsmodels, Python omogućuje jednostavno generiranje i analizu različitih vrsta statističkih podataka, uključujući statistike, distribucije i statističke testove. NumPy i Pandas služe kao temelj za rad s podacima, omogućujući manipulaciju i analizu velikih skupova podataka, dok SciPy i Statsmodels pružaju napredne funkcionalnosti za provedbu statističkih testova. Ove biblioteke omogućuju izvođenje raznih testova, poput T-testova, testova analize varijance, Friedmanovog testa, Wilcoxonovog testa, itd.

Flask je web okvir zasnovan na Pythonu, dizajniran s ciljem brzog i jednostavnog stvaranja web aplikacija. Za razliku od složenijih okvira poput Django, Flask slijedi minimalistički pristup, pružajući osnovni alatni okvir za izgradnju aplikacija bez pretjeranih ograničenja. Ovaj pristup omogućuje programerima veliku slobodu i fleksibilnost u odabiru biblioteka i komponenti koje će koristiti, čime se Flask spretno prilagođava specifičnim potrebama svakog projekta [11].

Flask podržava razvoj aplikacija koristeći RESTful API-je, što je ključno za moderne web aplikacije koje se temelje na mikroservisima i klijent-poslužitelj arhitekturi.

Njegov jednostavan sustav ruta omogućuje lako definiranje URL-ova i povezivanje s funkcijama koje obrađuju korisničke zahtjeve, a podrška za ekstenzije daje mogućnost proširivanja funkcionalnosti bez složenih podešavanja.

3.2.4 MongoDB

MongoDB je nerelacijski sustav za upravljanje bazama podataka koji pohranjuje podatke u obliku dokumenata, a ne u tradicionalnim tablicama i redovima, karakterističnim za relacijske baze podataka. Ovakve baze podataka spadaju u kategoriju NoSQL, što znači da ne koriste relacijski model i omogućuju fleksibilniji pristup pohrani podataka. Ova prilagodljivost omogućava korisnicima jednostavno spremanje i dohvaćanje podataka različitih struktura, što čini MongoDB idealnim za aplikacije koje rade s velikim količinama raznovrsnih podataka.

Jedna od ključnih prednosti MongoDB-a je njegova skalabilnost i lakoća upravljanja, posebno kod aplikacija koje zahtijevaju brzo širenje i prilagodbu u skladu s rastom baze podataka. NoSQL sustavi, poput MongoDB-a, manje su strukturirani u usporedbi s tradicionalnim SQL bazama podataka, što ih čini posebno pogodnima za aplikacije koje obrađuju velike količine nestrukturiranih ili polustrukturiranih podataka. Dodatno, jednostavna konfiguracija i minimalna potreba za složenim kodom pri radu s bazom čine MongoDB korisnim za brzi razvoj i postavljanje aplikacija.

Još jedna značajna prednost MongoDB-a je njegova fleksibilnost u pogledu implementacije – može se pokrenuti u oblaku (engl. *cloud*) ili na lokalnom poslužitelju, čime se omogućuje dodatna prilagodljivost i skalabilnost ovisno o potrebama projekta. MongoDB Atlas je platforma u oblaku koja omogućava pokretanje, upravljanje i skaliranje MongoDB baza podataka. Dostupna je na popularnim pružateljima usluga u oblaku kao što su AWS, Google Cloud i Microsoft Azure. Atlas automatizira zadatke poput postavljanja baze, sigurnosnih ažuriranja, sigurnosnih kopija i nadgledanja baze podataka, čime omogućuje brži i sigurniji razvoj aplikacija [12].

3.2.5 D3.js

D3.js je besplatna JavaScript knjižnica koja omogućuje izradu dinamičnih i interaktivnih vizualizacija podataka u web preglednicima. Zbog velike korisničke zajednice,

koja pruža podršku i mnoštvo resursa, ova tehnologija predstavlja jedan od od najpopularnijih alata za vizualizaciju podataka. D3.js je razvijen sa standardnim web tehnologijama kao što su SVG (Scalable Vector Graphics), HTML (HyperText Markup Language) i CSS (Cascading Style Sheets), čime omogućuje potpunu kontrolu nad prikazom podataka [13].

Za razliku od drugih popularnih biblioteka za vizualizaciju, poput Chart.js ili Google Charts, koje pružaju unaprijed definirane grafikone i vizualizacije, D3.js nudi mnogo veću prilagodljivost. Omogućuje potpunu kontrolu nad svakim elementom vizualizacije i omogućuje korisniku fleksibilnost razvoja vizualizacije, što ga čini idealnim rješenjem za složene prikaze podataka.

Jedna od ključnih značajki je sposobnost povezivanja podataka s elementima DOM-a, što omogućuje dinamičko ažuriranje vizualizacija u stvarnom vremenu na temelju promjena podataka. To je posebno korisno u aplikacijama čiji se podaci brzo mijenjaju, poput financijskih tržišta ili praćenja performansi sustava.

Za razliku od tradicionalnih alata za vizualizaciju, D3.js omogućuje izradu izrazito interaktivnih prikaza, u koje je moguće ugraditi animacije, filtriranje podataka, promjene prikaza na temelju korisničkih akcija i druge napredne funkcionalnosti.

Prednost korištenja SVG elemenata u D3.js je i prilagodljivost vizualizacija različitim veličinama ekrana bez gubitka kvalitete. SVG je vektorski format, što znači da se grafika može skalirati na različite veličine ekrana bez pikselizacije, čime se osigurava oštrina i čitljivost vizualizacija na uređajima različitih rezolucija.

3.2.6 Clerk

Clerk je platforma za autentifikaciju i upravljanje korisnicima koja omogućava jednostavnu integraciju sigurnih i prilagodljivih rješenja za prijavu u moderne web aplikacije. Podržava različite metode autentifikacije, uključujući tradicionalne prijave putem e-maila i lozinke, kao i OAuth prijave putem popularnih servisa kao što su Google, Facebook i GitHub [14].

Platforma je dizajnirana kako bi razvoj aplikacija učinila bržim i jednostavnijim, nudeći unaprijed pripremljene komponente i bogatu dokumentaciju. Na taj način, programeri mogu brzo implementirati napredne sustave za autentifikaciju bez potrebe

za razvojem kompleksnih procesa od nule.

Clerk također omogućuje jednostavno upravljanje korisničkim podacima, statusima prijave, resetiranjem lozinki te kontrolom pristupa. Uz to, pruža detaljan uvid u aktivnosti korisnika i upravljanje sjednicama, što je ključno za aplikacije koje obrađuju osjetljive podatke i zahtijevaju visoku razinu sigurnosti.

3.3 Implementacija poslužiteljske strane sustava

U ovom potpoglavlju detaljno će biti objašnjen razvoj poslužiteljske strane sustava. Sustav je izgrađen korištenjem dvaju radnih okvira: Next.js i Flask. Iako je Next.js prvenstveno poznat kao okvir za React, njegova ključna značajka je mogućnost objedinjavanja klijentske i poslužiteljske strane unutar jednog okvira. Na taj način omogućuje stvaranje API krajnjih točaka bez potrebe za klasičnim poslužiteljskim sustavom. Ova arhitektura omogućava pojednostavljenje tehnološkog stoga i olakšava proces razvoja, čineći ga intuitivnijim i efikasnijim. U aplikaciji su kroz Next.js razvijene funkcionalnosti kao što su stvaranje, uređivanje i brisanje projekata, te učitavanje i brisanje CSV datoteka. Međutim, iako Next.js nudi mnoge prednosti, izazov se javlja u domeni inferencijalne statistike, jer JavaScript kao programski jezik nije optimiziran za statističke izračune. Budući da JavaScript primarno služi za razvoj korisničkih sučelja, bilo je potrebno implementirati rješenje koje podržava složene statističke operacije. Zbog toga je odabran Flask, Pythonov radni okvir, za izračun statističkih testova na poslužiteljskoj strani. Python je dobro poznat po svojoj sposobnosti da efikasno rješava matematičke i statističke probleme, što ga čini idealnim za ovaj dio aplikacije.

U razvoju aplikacija temeljenih na Next.js i Reactu, često se koristi MongoDB kao baza podataka zbog svoje fleksibilnosti i sposobnosti efikasnog upravljanja velikim količinama podataka. Kako bi se olakšao rad s bazom podataka, koristi se MongoDB Atlas, platforma u oblaku koja pojednostavljuje postavljanje, upravljanje i skaliranje baza podataka. Atlas omogućava brzo stvaranje baza podataka, jednostavno povezivanje s aplikacijom te automatsko upravljanje i kontroliranje sigurnosnim postavkama i performansama.

Za strukturirano upravljanje podacima i interakciju s MongoDB-om, koristi se knjižnica Mongoose. Mongoose omogućava rad s MongoDB-om na objektno orijen-

Poglavlje 3. Razvoj Web aplikacije za deskriptivnu i inferencijalnu analizu podataka

tirani način, pružajući sloj apstrakcije između aplikacije i baze podataka. Pruža strukturirani način definiranja modela i shema podataka, što olakšava validaciju, upravljanje vezama između entiteta i održavanje integriteta podataka. Na taj način, aplikacija može koristiti ORM pristup za rad s nerelacijskim podacima, što smanjuje složenost koda i povećava održivost projekta.

Baza podataka se sastoji od četiri glavne kolekcije:

- `user`
- `project`
- `graph options`
- `axis`

Njihov sadržaj i akcije biti će objašnjene u sljedećim poglavljima.

3.3.1 Kolekcija *user*

Kolekcija *user* pohranjuje ključne opće informacije o korisnicima aplikacije, a sastoji se od sljedećih polja:

- `id` - jedinstveni identifikator korisnika
- `clerkId` - jedinstveni identifikator generiran putem Clerk servisa
- `username` - korisničko ime
- `email` - e-mail adresa korisnika
- `picture` - poveznica na profilnu sliku korisnika s društvenih mreža
- `createdAt` - vremenska oznaka registracije korisnika

Polje `clerkId` označava jedinstveni identifikator koji dodjeljuje Clerk platforma tijekom korisničke registracije koristeći jednu od podržanih metoda autentifikacije. Clerk API omogućuje registraciju i upravljanje korisničkim podacima kroz njegovo korisničko sučelje, a radi održavanja dosljednosti između Clerk servisa i baze podataka, koristi se funkcionalnost *webhooks*. *Webhooks* omogućuju automatsko okidanje događaja poput stvaranja, uređivanja ili brisanja korisnika, osiguravajući tako ažuriranje podataka u bazi kad god se na Clerk servisu dogodi promjena.

Za stvaranje, uređivanje i brisanje dokumenata iz baze podataka koriste se asinkrone funkcije iz Mongoose knjižnica i njene klase `Model`:

- `User.create()`
- `User.findOneAndUpdate()`
- `User.findOneAndDelete()`

Osim ovih, funkcije klase `Model` koriste se i za druge radnje povezane s korisnicima, poput dohvaćanja korisnika prema `id`-u, dohvaćanja trenutačno prijavljenog korisnika ili dohvaćanja korisnika putem e-mail adrese, što omogućava brzu i fleksibilnu interakciju između aplikacije i baze podataka.

3.3.2 Kolekcija *project*

Kolekcija *project* je ključna kolekcija u kojoj se nalaze sve bitne informacije za razvoj projekta. Sadrži sljedeća polja:

- `name` - ime projekta
- `fileName` - naziv učitane datoteke
- `uniqueFileName` - unikatni naziv učitane datoteke
- `size` - veličina učitane datoteke
- `delimiter` - graničnik CSV datoteke
- `graphOptions` - jedinstveni identifikator dokumenta *graphOptions* koji je povezan s ovim projektom
- `collaborators` - polje (*engl. array*) koje sadrži jedinstvene identifikatore korisnika koji su dodani kao suradnici u projektu
- `owner` - jedinstveni identifikator korisnika koji je generirao projekt
- `createdAt` - vremenska oznaka stvaranja projekta
- `updatedAt` - vremenska oznaka uređivanja projekta

Kako bi aplikacija pravilno obavljala funkcionalnosti vezane uz rad s CSV datotekama, potrebno je pohraniti ključne informacije o svakoj učitanoj datoteci, poput njezinog naziva, veličine i korištenog graničnika. Budući da se lako može dogo-

diti da se učita više datoteka s istim nazivom, uvedeno je polje `uniqueFileName`. Ovo polje sadrži jedinstveni naziv datoteke, stvoren kombinacijom vremenske oznake (`timestamp`) i originalnog imena datoteke, čime se osigurava prepoznavanje svake datoteke kao zasebnog entiteta.

CSV datoteke predstavljaju jednostavan i široko korišten format za pohranu tabličnih podataka u tekstualnom obliku. Svaka linija u CSV datoteci predstavlja redak podataka, pri čemu su vrijednosti unutar redaka odvojene graničnikom (`delimiter`), koji je najčešće zarez, ali može biti i bilo koji drugi znak. Zbog toga, prilikom učitavanja CSV datoteka, korisnik mora specificirati koji je znak korišten kao graničnik u datoteci kako bi aplikacija ispravno obradila podatke.

Entiteti *graph-options* i *project* su u relaciji 1:1. Te kolekcije su namjerno odvojene kako bi se jasnije razlikovale informacije koje se pohranjuju, omogućujući bolje razumijevanje njihovih uloga u sustavu.

Jedna od ključnih funkcionalnosti aplikacije je omogućavanje kolaboracije među korisnicima prilikom razvijanja projekata. Vlasnik projekta, čiji se jedinstveni identifikator nalazi u polju `owner`, ima mogućnost dodavanja drugih korisnika u polje `collaborators`. U tom se polju nalaze identifikatori korisnika koji dobivaju pristup projektu, čime se olakšava timski rad i dijeljenje podataka.

3.3.3 Kolekcija *graph-options*

Kolekcija *graph-options* pohranjuje podatke koje unosi korisnik, a odnose se na parametre potrebne za stvaranje grafova koji prikazuju rezultate deskriptivne statistike. Ova kolekcija sadrži varijable koje korisnik specificira za generiranje odgovarajućih statističkih prikaza, poput odabira tipova grafova, varijabli za analizu, te parametara za vizualizaciju podataka. Na taj način aplikacija može generirati relevantne statističke grafove koji odgovaraju korisnikovim potrebama i analitičkim zahtjevima.

Kolekcija sadrži ova polja:

- `title` - naslov grafa
- `type` - vrsta grafa
- `axes` - polje jedinstvenih oznaka dokumenata *axes*, odnosno osi, koji su asociirani s *graph-options* dokumentom

Poglavlje 3. Razvoj Web aplikacije za deskriptivnu i inferencijalnu analizu podataka

U web aplikaciji podržano je pet tipova grafova: histogram, linijski graf, linijski graf s više linija (multilinijski), stupčasti graf i kutijasti dijagram (*engl. box-plot*). Ovi grafovi omogućuju raznolike načine prikaza podataka, od distribucije vrijednosti (histogram) i vremenskih serija (linijski i multilinijski graf), do usporedbe kategorija (stupčasti graf i pravokutni dijagram). Svaka vrsta grafa pruža specifične uvide, prilagođene različitim vrstama analiza koje korisnici mogu provoditi unutar aplikacije.

Svaki dokument u kolekciji *graph-options* povezan je s jednim ili više dokumenata iz kolekcije *axis*. Polje *axes* u dokumentu *graph-options* pohranjuje jedinstvene identifikatore dokumenata iz kolekcije *axis* koji su povezani s određenim grafom. Ova arhitektura omogućava fleksibilno definiranje više osi za jedan graf, a više o kolekciji *axis* biti će objašnjeno u sljedećem potpoglavlju.

3.3.4 Kolekcija *axis*

Kolekcija *axis* služi za povezivanje vrijednosti jednog od stupaca iz učitane CSV datoteke na odgovarajuću os grafičkog prikaza. Ova kolekcija omogućava povezivanje podataka s odgovarajućim osima grafova, što olakšava prikaz i analizu odabranih varijabli. Sadrži sljedeća polja:

- **header** - naziv zaglavlja iz odabranog stupca učitane datoteke
- **axis** - oznaka osi na kojoj će se prikazivati podaci (x ili y)
- **createdAt** - vremenska oznaka stvaranja dokumenta

Svaki dokument u kolekciji *graph-options* mora biti povezan s jednim ili više dokumenata iz kolekcije *axis*, ovisno o vrsti grafa koji je odabran. Primjerice, za histogram se koristi samo x-os, dok će se za druge vrste grafova povezati i x-os i y-os. Poseban slučaj predstavlja multilinijski graf, gdje korisnik ima mogućnost prikazati više linija na istom grafu.

3.3.5 Implementacija statističkih testova

Kod provođenja statističke analize od velike važnosti je provjeriti jesu li dobiveni podaci značajni sa znanstvene strane, ili se oni mogu shvatiti kao slučajnost. To postizemo inferencijalnom statistikom. Za generiranje statističkih testova putem

Flask API-ja, korištena je Python knjižnica SciPy i Statsmodels koje pružaju opsežne funkcije za provođenje različitih statističkih testova. Kako bi se pokrio što širi spektar vrsta podataka implementirani su sljedeći statistički testovi: *t-test* s nezavisnim uzorcima, *t-test* sa zavisnim uzorcima, *Wilcoxonov* test, *Mann-Whitneyev* test, *Friednmanov* test, jednosmjerna analiza varijance, analiza varijance s ponovljenim mjerenjima, te *post hoc* testovi *Tukeyev* test, *Bonferronijeva* korekcija i *Wilcoxonov* test. Svaki od njih prilagođen je specifičnim istraživačkim potrebama za analizu podataka. U Flask radnom okviru, svaki test je implementiran kao API krajnja točka koja omogućava korisnicima pozivanje specifičnih statističkih testova slanjem HTTP zahtjeva s potrebnim podacima u JSON formatu.

Ovim pristupom razvijena je fleksibilnost korištenja različitih statističkih metoda bez direktnog pristupa podacima na poslužitelju. Unutar svake API krajnje točke koristi se odgovarajuća funkcija iz SciPy ili StatsModels knjižnice za izvršavanje traženog testa, kao što je prikazano na primjerima:

- Krajnja točka za *t-test* s nezavisnim uzorcima nalazi se na putanji `/flask_api/ttest_ind` i prima dva niza podataka koji se žele usporediti. Za izračun inferencijalne statistike koristi se funkcija `statsmodels.ttest_ind`, koja vraća varijable *t-statistike* i *p-vrijednosti*. *T-statistika* prikazuje udaljenost između srednjih vrijednosti dviju grupa izraženih u standardnoj devijaciji, dok *p-vrijednost* označava vjerojatnost da je zabilježena razlika između dvije skupine podataka nastala slučajno, pomažući u procjeni statističke značajnosti razlike između njih.

Ova implementacija je identična i za ove testove koji također uspoređuju dva niza podataka:

- *t-test* sa zavisnim uzorcima, putanja `/flask_api/ttest_rel`, funkcija `statsmodels.ttest_rel`
 - *Wilcoxonov* test, putanja `/flask_api/wilcoxon`, funkcija `statsmodels.wilcoxon`
 - *Mann-Whitneyev* test, putanja `/flask_api/mann_whitney`, funkcija `statsmodels.mannwhitneyu`
- Implementirani su i testovi koji omogućuju usporedbu tri ili više skupina po-

dataka: jednosmjerna analiza varijance (*engl. One-way ANOVA*), analiza varijance s ponovljenim mjerenjima (*Repeated measures ANOVA*) i *Friedmanov* test. Jednosmjerna analiza varijance koristi se za nezavisne grupe podataka, dok analiza varijance s ponovljenim mjerenjima i *Friedmanov* test uspoređuju zavisne grupe. Svaki od ovih testova primjenjuje se u različitim situacijama, ovisno o pripremljenim podacima.

Za implementaciju jednosmjerne analize varijance koristi se funkcija `statsmodels.f_oneway`, dok *Friedmanov* test koristi funkciju `statsmodels.friedmanchisquare`. Za analizu varijance s ponovljenim mjerenjima upotrebljava se `statsmodels.AnovaRM`, pri čemu je potrebno specificirati stupac za sudionike u eksperimentu, zavisnu varijablu te unutarispitanički faktor (*engl. within-factor*) koji označava uvjete u kojima svaki sudionik sudjeluje.

Sve funkcije vraćaju vrijednosti *F-statistike* i *p-vrijednosti*. *F-statistika* služi za procjenu razlika između grupa te analizu varijabilnosti unutar i između njih, dok *p-vrijednost* označava vjerojatnost da je primijećena razlika između dviju skupina rezultat slučajnosti. Ako se tijekom analize utvrdi značajna razlika, potrebno je provesti *post hoc* analize kako bismo precizno identificirali koje se točno grupe međusobno razlikuju. Ove dodatne analize pružaju dublje razumijevanje rezultata i omogućuju istraživačima da donesu informirane zaključke o svojim podacima.

- *Post hoc* analize su testovi koji se provode nakon inicijalnih testova koji uspoređuju više od dvije grupe podataka. Ovi testovi omogućuju dublje razumijevanje rezultata i pomažu u donošenju informiranih zaključaka o podacima. U aplikaciji su implementirana tri *post hoc* testa: *Tukeyev* test, *Bonferronijeva* korekcija i *Wilcoxonov* test. Ako se u analizi koristi *Friedmanov* test, *post hoc* analize provode se uz pomoć *Wilcoxonovog* testa, koji je prilagođen za analizu razlika povezanih grupa s neparametarskim podacima. Nakon jednosmjerne analize varijance najčešće se primjenjuje *Tukeyev* test, a *Bonferronijeva* korekcija se može primijetiti na bilo koji test. *Tukeyev* test je koristan za usporedbu svih parova grupa nakon analize varijance testa, dok *Bonferronijeva* korekcija služi kada se unaprijed definiraju specifične usporedbe koje treba testirati.

API krajnja točka za provođenje *Tukeyevog* testa koristi funkciju `statsmodels.pairwise_tukeyhsd`, koja vraća razliku u srednjim vrijednos-

tima, *p-vrijednost* i *interval pouzdanosti* za svaki par grupa podataka. U slučaju *Bonferronijeve* korekcije, podaci se prvo organiziraju u objekt razreda `statsmodels.MultiComparison`, a zatim se koristi funkcija `allpairtest` za provođenje testa na svim parovima grupa. Ova funkcija vraća detaljne rezultate, uključujući *p-vrijednosti*, intervale pouzdanosti i statističke testove za svaku usporedbu. Za *Wilcoxonov* test, implementacija prolazi kroz svaki par grupa unutar `for` petlje, provodeći test koristeći funkciju `statsmodels.wilcoxon` i vraćajući rezultate u obliku statistike testa i *p-vrijednosti*.

Provođenjem jednog od ovih testova moguće je precizno utvrditi koje specifične grupe pokazuju značajne razlike u rezultatima. Ovi testovi pružaju detaljne informacije o međusobnim odnosima između grupa, omogućavajući istraživačima da donesu informirane zaključke i bolje razumiju obrasce u prikupljenim podacima. Na taj način, statističke analize ne samo da identificiraju postojanje razlika, već i omogućuju njihovu dublju interpretaciju.

Generalno, razvijanje API kranjih točaka u Flasku omogućuje fleksibilan, skalabilan način pružanja različitih statističkih testova, omogućujući korisnicima analizu podataka kroz HTTP zahtjeve i interpretiranje rezultata u JSON formatu.

3.4 Implementacija klijentske strane sustava

U ovom potpoglavlju opisat će se implementacija klijentske strane sustava, naglašavajući razvoj ključnih komponenti koje omogućuju funkcionalnost aplikacije. Svaka od njih ima specifičnu ulogu u interakciji s korisnikom te obradi i prikazu podataka.

3.4.1 Komponenta *FileUpload*

Stvaranje projekta zasnovanog na statističkoj analizi odvija se kroz nekoliko komponenti, od kojih je prva

`FileUpload.tsx`, zadužena za prijenos CSV datoteke i unos ključnih podataka o projektu. Ova komponenta omogućava unos naziva projekta, izbor sudionika i postavljanje simbola za razdvajanje podataka. Za upravljanje obrascem koristi se knjižnica `react-hook-form`, dok `zod` osigurava validaciju unesenih podataka.

Učitavanje datoteke moguće je na dva načina: klikom na označeno područje unutar obrasca ili povlačenjem datoteke u taj prostor, a za ovu funkcionalnost koristi se `react-dropzone`. Nakon odabira datoteke, funkcija `uploadFile` pokreće prijenos putem API poziva i prati napredak učitavanja, dok funkcija `onCancel` omogućava otkazivanje prijenosa. Uspješnim prijenosom datoteka se sprema na poslužitelj.

Također, komponenta nudi funkcionalnost pretraživanja korisnika putem e-maila radi dodavanja sudionika u projekt, što se ostvaruje pozivanjem funkcije `findUsersByEmail`. Ako korisnik odabere drugog korisnika kao suradnika, suradnik stječe prava čitanja i uređivanja projekta.

3.4.2 Komponenta *Parser*

Komponenta *Parser* koristi knjižnicu `Papa Parse` za obradu CSV podataka. Jedan od ključnih argumenata funkcije je varijabla `rowData`, koja sadrži podatke učitane u prethodnoj komponenti. Funkcija `papaparse.parse` koristi vrijednost graničnika i `{rowData}` kako bi odvojila zaglavlje od podataka te identificirala tipove podataka unutar CSV datoteke. Nakon što se uklone `null` ili prazne vrijednosti te se stringovi s numeričkim podacima pretvore u decimalne brojeve, komponenta vraća dva niza: `headers` i `values`. `headers` sadrži vrijednosti zaglavlja, dok `values` predstavlja podatke organizirane po redovima.

3.4.3 Komponenta *GraphOptions*

Komponenta *GraphOptions* omogućava korisnicima odabir opcija za prikaz deskriptivne statistike, što je ključni korak u radu na projektu. Središnji dio komponente je interaktivni obrazac koji koristi `react-hook-form` za upravljanje obrascem te knjižnicu `zod` za validaciju. Od korisnika se očekuje unos sljedećih parametara:

- naslov grafa (opcionalno)
- tip grafa — opcije uključuju histogram, linijski graf, multilinijski graf, stupčasti graf i kutijasti dijagram
- odabir stupca na x-osi
- odabir stupca na y-osi

- sortiranje podataka (samo kod stupčastog grafa)

Komponenta koristi varijablu `formSchema`, definiranu putem Reactove `useState` udice, kako bi prilagodila validaciju obrasca prema odabranom tipu grafa. Multilinijski graf dodatno omogućava unos više stupaca za y-osi, stupčasti graf ima opciju sortiranja, dok `useEffect` funkcije osiguravaju dinamičko prilagođavanje sučelja na temelju odabranog tipa grafa.

Na temelju korisničkog unosa, ažurira se konfiguracija opcija za grafove, uključujući validaciju tipa podataka na x i y osima putem funkcije `validateData`, kako bi se osigurala kompatibilnost s odabranim prikazom. Ovisno o vrsti grafa, pravila za odabir stupaca su sljedeća:

- Histogram: Odabire se samo jedan stupac za x-os, koji mora sadržavati numeričke vrijednosti.
- Stupčasti graf i kutijasti dijagram: Odabire se po jedan stupac za x- i y-os. Na x-osi stupac mora sadržavati tekstualne podatke, dok na y-osi treba biti numerički stupac.
- Linijski graf: Za x-os se odabire stupac s vremenskim oznakama, dok na y-osi mora biti stupac koji sadrži brojeve.
- Multilinijski graf: Na x-osi se odabire stupac s vremenskim oznakama, dok je moguće odabrati više stupaca za y-os, pri čemu svi stupci moraju sadržavati numeričke podatke.

Nakon popunjavanja obrasca, podaci se šalju na poslužitelj pomoću funkcija poput `createAxis` i `createGraphOptions`, koje stvaraju i ažuriraju postavke grafova u bazi podataka.

3.4.4 Komponenta *Graph*

Nakon uspješnog ispunjavanja obrasca na komponenti `GraphOptions` korisnik je preusmjeren na komponentu `Graph` kojoj je glavna zadaća prikaz odabranog grafa putem knjižnice `d3.js`. Na početku komponente definirana je varijabla `svgRef` putem Reactove funkcije `useRef`. "Udica" `useRef` stvara postojanu referencu koja može čuvati promjenjivu vrijednost ili referencu na DOM element, a pritom ne uzrokuje ponovno

generiranje prikaza nakon ažuriranja. U našem slučaju koristimo ju za SVG vektor grafa. Potom pripremamo podatke na x-osi i y-osi, računamo dimenzije SVG vektora na temelju količine podataka i tipa grafa. Ukoliko je tip grafa linijski ili multilinijski, pretvaramo vrijednosti x-osi iz znakovnog niza u vremensku oznaku putem funkcije `d3.timeParse`.

Zatim je definirana "udica" `useEffect` u kojoj se prvo definiraju skale x i y osi na temelju definirane vrste grafa, definiraju se dimenzije SVG vektora te se potom po njemu crtaju dijelovi grafa, osi i njihove skale.

Histogram

Ukoliko je odabran tip grafa histogram, prvo se stvaraju intervali (*engl. bins*) na x-osi koristeći funkciju `d3.bin()` te se definira broj intervala koristeći funkciju `d3.thresholds()`. Potom se definiraju linearne skale za x i y os, te im se pridružuje domena, odnosno minimalne i maksimalne vrijednosti podataka. Prikaz grafa događa se u funkciji `renderHistogram()`, koji se iterira kroz svaki interval i pomoću funkcije `d3.append("rect")` dodaje pravokutnike kao SVG elemente na graf, gdje pravokutnici predstavljaju frekvenciju podataka unutar svakog intervala. Na SVG referencu se dodaju atributi s informacijom o boji pravokutnika, podacima koji će odrediti vrijednost na y osi, početnu i krajnju točku na x-osi svakog pravokutnika te njegovu širinu.

Linijski grafovi (linijski i multilinijski)

Za linijske grafove na x osi je vremenska skala koju definiramo funkcijom `d3.scaleUtc()`, dok je na y osi linearna brojevana skala. Linija se crta pomoću funkcije `d3.line()`, koja omogućava crtanje putanje kroz točke podataka, čime se dobiva vizualizacija koja prikazuje promjene u vrijednosti kroz vrijeme. Na SVG referencu dodaju se još i informacije o boji linije i njenoj debljini. Kod multilinijskog grafa boje se odabiru korištenjem funkcije `d3.schemeCategory10()`, koja vraća unaprijed definiranu paletu s 10 boja, te je korisna za jasno razlikovanje među linijama kada se vizualizira više varijabli na istom grafu.

Stupčasti graf

Stupčasti grafovi prikazuju kategorijske vrijednosti na x-osi, s opcijom sortiranja podataka, pri čemu se za definiranje skale koristi funkcija `d3.scaleBand`. Na temelju odabranog redosljeda definirana je domena skale x-osi, dok y-os koristi linearnu numeričku skalu kako bi prikazala kvantitativne vrijednosti. Proces crtanja započinje stvaranjem novog grupnog elementa unutar SVG elementa, zatim se određuje boja stupaca i dodaju pravokutnici za svaki red podataka, pri čemu se za svaki pravokutnik definira x-koordinata početka i y-koordinata gornjeg ruba, kao i njegova širina i visina.

Kako bi graf pružio dublji uvid u podatke, za svaku kategoriju na x-osi izračunava se srednja vrijednost i standardna devijacija, čime se prikazuje varijabilnost podataka i raspršenje oko srednje vrijednosti. Ovo raspršenje grafički se prikazuje pomoću traka pogreške (*engl. error bars*), koje ukazuju na raspon standardne devijacije za svaku vrijednost na grafu.

Kutijasti graf

Kutijasti graf ilustrira raspodjelu podataka kroz pet ključnih statističkih mjera: minimalnu vrijednost podataka, prvi kvartil (Q1), medijan, treći kvartil (Q3), i maksimalnu vrijednost, omogućujući tako vizualizaciju centralne tendencije i raspona podataka te identifikaciju mogućih odstupanja. Poput stupčastog grafa, x-os prikazuje kategorijske vrijednosti, dok y-os koristi linearnu skalu za prikaz numeričkih vrijednosti.

Za izračun kvartila koriste se funkcije `d3.quantile()`, gdje se specificira pozicija kvartila decimalnim brojem između 0 i 1. Dodatno, funkcija `d3.mean()` izračunava aritmetičku sredinu, a `d3.deviation()` standardnu devijaciju za svaki skup podataka. Na temelju ovih vrijednosti, SVG elementu dodaju se pravokutnici čiji rubovi označavaju prvi i treći kvartil, linija koja predstavlja medijan, te linije koje predstavljaju raspon vrijednosti (*engl. whiskers*). Na taj način, kutijasti graf omogućuje jasan prikaz ključnih značajki distribucije podataka i identifikaciju odstupanja.

3.4.5 Komponenta *Statistics*

Komponenta *Statistics* omogućuje analizu ključnih statističkih parametara skupa podataka korištenog za generiranje grafa. Ova komponenta koristi funkcije iz knjižnice `d3.js` za izračun statističkih vrijednosti kao što su: aritmetička sredina (`d3.mean()`), medijan (`d3.median()`), standardna devijacija (`d3.deviation()`), te najmanja i najveća vrijednost (`d3.extent()`). Pored toga, izračunava i asimetriju, kako bi se dobila dodatna perspektiva o obliku distribucije podataka.

Za određene vrste grafova, komponenta vrši prilagođenu analizu:

- Stupčasti i kutijasti grafovi - Osim statistike cijelog skupa podataka, računaju se i statističke vrijednosti zasebno za svaku grupu podataka na x-osi, što omogućuje usporedbu među različitim kategorijama.
- Multilinijski graf - Osim statistike cijelog skupa podataka, statističke vrijednosti računaju se i pojedinačno za svaku liniju, omogućujući uvid u distribuciju podataka za svaku varijablu predstavljenu na grafu.

Komponenta obuhvaća sve bitne statističke mjere potrebne za tumačenje rezultata, pružajući detaljan uvid u raspodjelu i varijabilnost podataka na grafu.

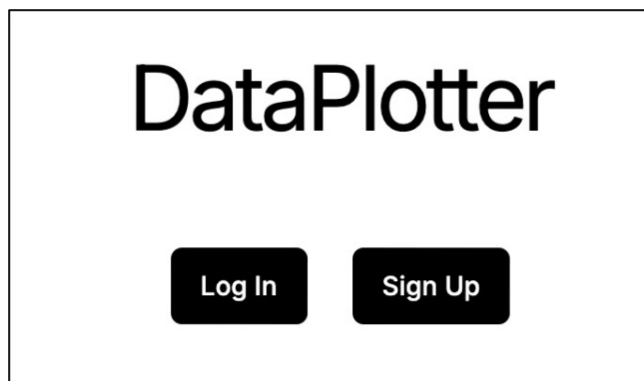
Poglavlje 4

Prikaz karakterističnih slučajeva korištenja

U ovom poglavlju prikazani su karakteristični slučajevi korištenja aplikacije, koji obuhvaćaju ključne funkcionalnosti sustava te demonstriraju način na koji aplikacija olakšava rad s istraživačkim podacima. Prikaz započinje postupkom prijave korisnika i upravljanjem korisničkim podacima, nakon čega slijedi pregled postojećih projekata i stvaranje novih istraživačkih inicijativa. Detaljno su obrađeni koraci učitavanja eksperimentalnih podataka te opcije za grafički prikaz i analizu kako bi se dobio vizualni uvid u podatke. Uz to, objašnjen je proces korištenja funkcionalnosti za pregled grafova, izvođenje deskriptivne statistike te provođenje statističkih testova i post-hoc analiza. Time se naglašava kako aplikacija omogućava korisnicima učinkovitu analizu podataka i donošenje informiranih zaključaka.

4.1 Prijava u sustav i upravljanje korisničkim podacima

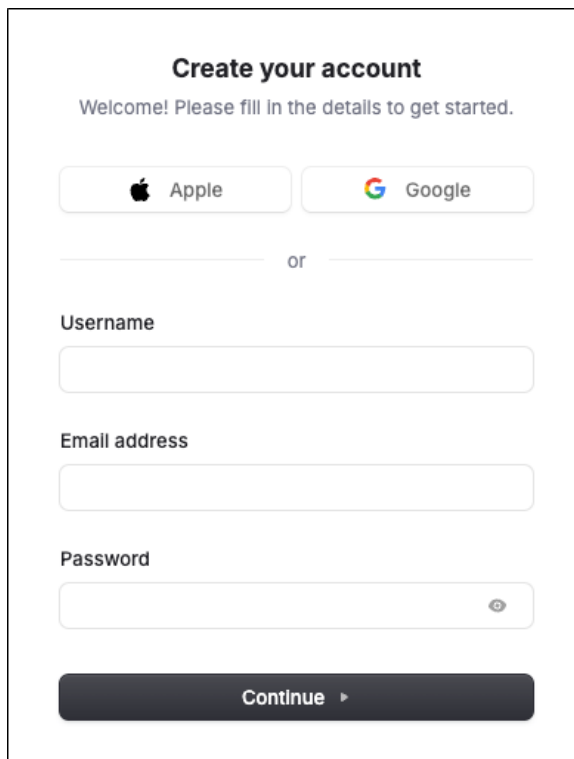
Pri otvaranju početne stranice aplikacije ispisuje se naslov aplikacije "DataPlotter", te opcija za prijavu (*engl. log in*) i registraciju (*engl. sign up*), kao što je prikazano na slici 4.1.



Slika 4.1 Početna stranica aplikacije

Klikom na gumb za registraciju otvara se novi prozor, prikazan na slici 4.2, koji koristi aplikacijsko programsko sučelje Clerk za jednostavnu i sigurnu autentifikaciju korisnika. Korisnici mogu birati između više opcija prijave: putem Apple i Google računa, ili klasičnom registracijom unosom korisničkog imena, e-mail adrese i lozinke. Ove opcije omogućuju fleksibilnost i ubrzavaju postupak prijave. Ako se odabere prijava putem Apple ili Google računa, aplikacija prvo preusmjerava korisnika na sigurnu autentifikaciju putem odabranog servisa, a nakon uspješne prijave vraća ih u aplikaciju, gdje se traži samo unos korisničkog imena. Kod klasične prijave e-mailom, potrebno je unijeti lozinku koja sadrži najmanje osam znakova. Clerk također automatski provjerava je li lozinka prisutna u bazama podataka kompromitiranih lozinki radi dodatne sigurnosti profila. Nakon toga, na registriranu e-mail adresu šalje se šesteroznamenkasti kod za potvrdu e-maila i osiguranje točnosti unesene adrese.

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

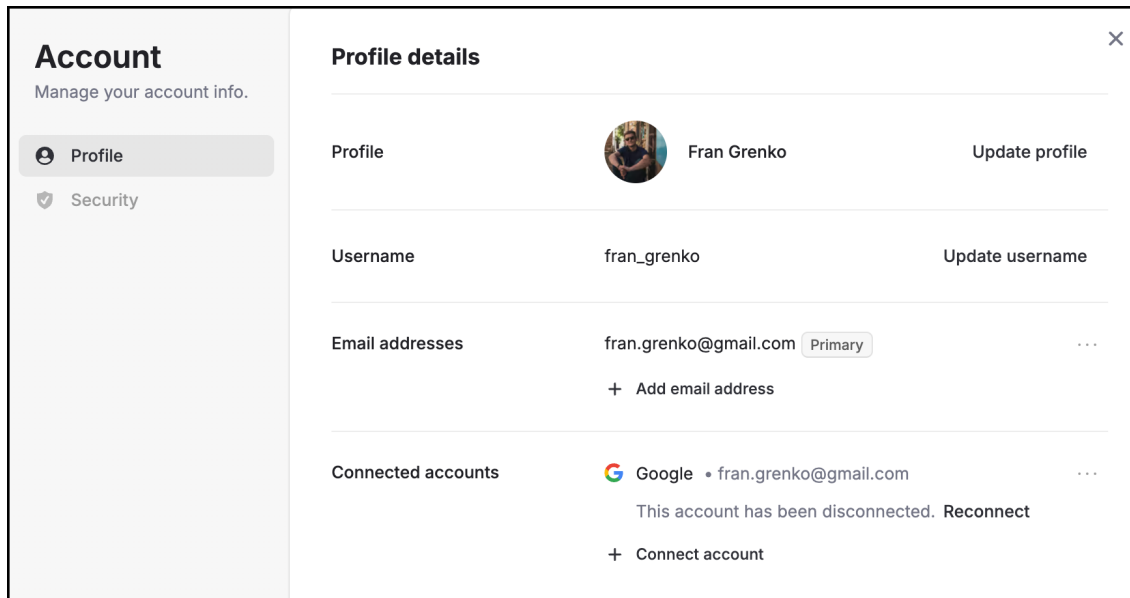


The image shows a registration form titled "Create your account". Below the title is a welcome message: "Welcome! Please fill in the details to get started." There are two buttons for social login: "Apple" and "Google". Below these is an "or" separator. The form contains three input fields: "Username", "Email address", and "Password". The "Password" field has a toggle icon for visibility. At the bottom is a dark "Continue" button with a right-pointing arrow.

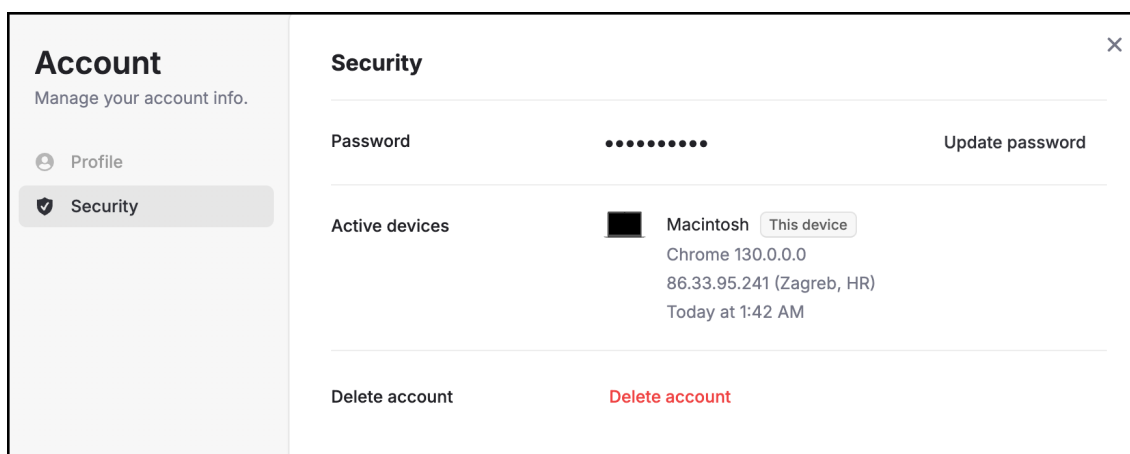
Slika 4.2 Prozor za registraciju u sustav

Nakon prijave u sustav, u gornjem desnom kutu navigacijske trake prikazuje se ikona profila korisnika. Klikom na ovu ikonu otvaraju se opcije za postavke korisničkog računa i odjavu iz aplikacije. Kroz aplikacijsko sučelje Clerk korisnici mogu postaviti osnovne informacije o računu, poput profilne fotografije, korisničkog imena, dodati ili promijeniti e-mail adresu te omogućiti prijavu putem vanjskih servisa. Prikaz navedenih postavki može se vidjeti na slici 4.3. Clerk također nudi sigurnosne postavke, prikazane na slici 4.4, koje uključuju promjenu lozinke, pregled svih aktivnih uređaja na kojima je korisnik prijavljen te mogućnost trajnog brisanja korisničkog računa.

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja



Slika 4.3 Postavke osnovnih informacija



Slika 4.4 Sigurnosne postavke

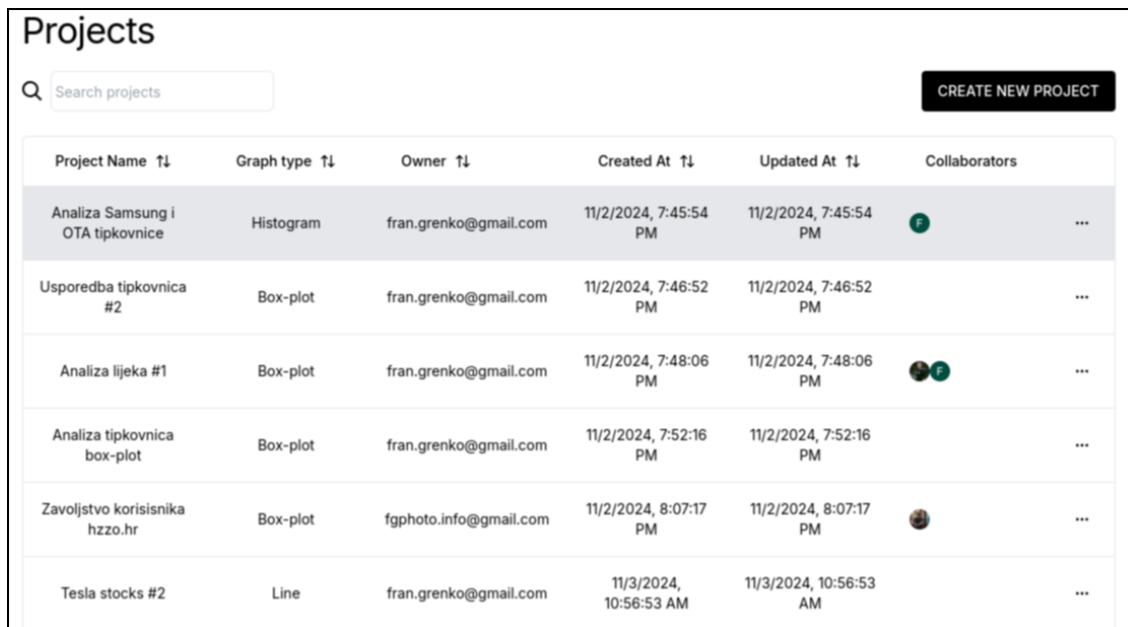
4.2 Prikaz postojećih projekata

Nakon uspješne prijave, korisnik se preusmjerava na putanju `/project`, koja sadrži pregled postojećih projekata, gdje se prikazuje tablica s popisom stvorenih projekata i opcijama za filtriranje. Ovdje su vidljivi svi projekti kojima je korisnik vlasnik ili je dodan kao sudionik. Tablica (Slika 4.5) uključuje sljedeće stupce:

- Ime projekta – prikazuje naziv projekta za lakšu identifikaciju.
- Vrsta grafa – informacija o vrsti grafa koja pomaže korisnicima razlikovati projekte po vizualizaciji podataka.
- Korisnik (vlasnik) – prikazuje korisničko ime osobe koja je stvorila projekt.
- Datum stvaranja – vremenska oznaka kada je projekt stvoren.
- Datum zadnjeg uređivanja – vremenska oznaka posljednje izmjene projekta.
- Sudionici – profilne fotografije korisnika uključenih u projekt; prelaskom pokazivača miša preko slike prikazuje se korisničko ime.
- Opcije – omogućuje pristup dodatnim radnjama za upravljanje projektom.

Korisnici imaju mogućnost uređivanja osnovnih informacija o projektu, prilagodbe postavki grafa, dupliciranja i brisanja projekta. Svi tekstualni stupci u tablici mogu se sortirati abecednim ili obrnutim redoslijedom radi bržeg pronalaženja projekata. Filtriranje se može provoditi prema nazivu projekta, a radi učinkovitosti prikaza koristi se straničenje, pri čemu se u početnom prikazu učitava samo prvih 10 projekata kako bi aplikacija imala bolje performanse.

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja



The screenshot shows a web interface titled "Projects". At the top left is a search bar with the placeholder text "Search projects". At the top right is a button labeled "CREATE NEW PROJECT". Below these is a table with the following columns: "Project Name", "Graph type", "Owner", "Created At", "Updated At", and "Collaborators". The table contains six rows of project data.

Project Name	Graph type	Owner	Created At	Updated At	Collaborators
Analiza Samsung i OTA tipkovnice	Histogram	fran.grenko@gmail.com	11/2/2024, 7:45:54 PM	11/2/2024, 7:45:54 PM	...
Usporedba tipkovnica #2	Box-plot	fran.grenko@gmail.com	11/2/2024, 7:46:52 PM	11/2/2024, 7:46:52 PM	...
Analiza lijeka #1	Box-plot	fran.grenko@gmail.com	11/2/2024, 7:48:06 PM	11/2/2024, 7:48:06 PM	...
Analiza tipkovnica box-plot	Box-plot	fran.grenko@gmail.com	11/2/2024, 7:52:16 PM	11/2/2024, 7:52:16 PM	...
Zavoljstvo korisnika hzzo.hr	Box-plot	fgphoto.info@gmail.com	11/2/2024, 8:07:17 PM	11/2/2024, 8:07:17 PM	...
Tesla stocks #2	Line	fran.grenko@gmail.com	11/3/2024, 10:56:53 AM	11/3/2024, 10:56:53 AM	...

Slika 4.5 Prikaz stvorenih projekata

4.3 Stvaranje novog projekta

Novi projekt može se stvoriti putem putanje `/project/new`, gdje korisnik unosi opće informacije o projektu, učitava eksperimentalne podatke i definira opcije za grafički prikaz.

4.3.1 Učitavanje eksperimentalnih podataka

Prvi korak u stvaranju projekta uključuje unos osnovnih informacija i učitavanje eksperimentalnih podataka. Slika 4.6 prikazuje polja dostupna za unos:

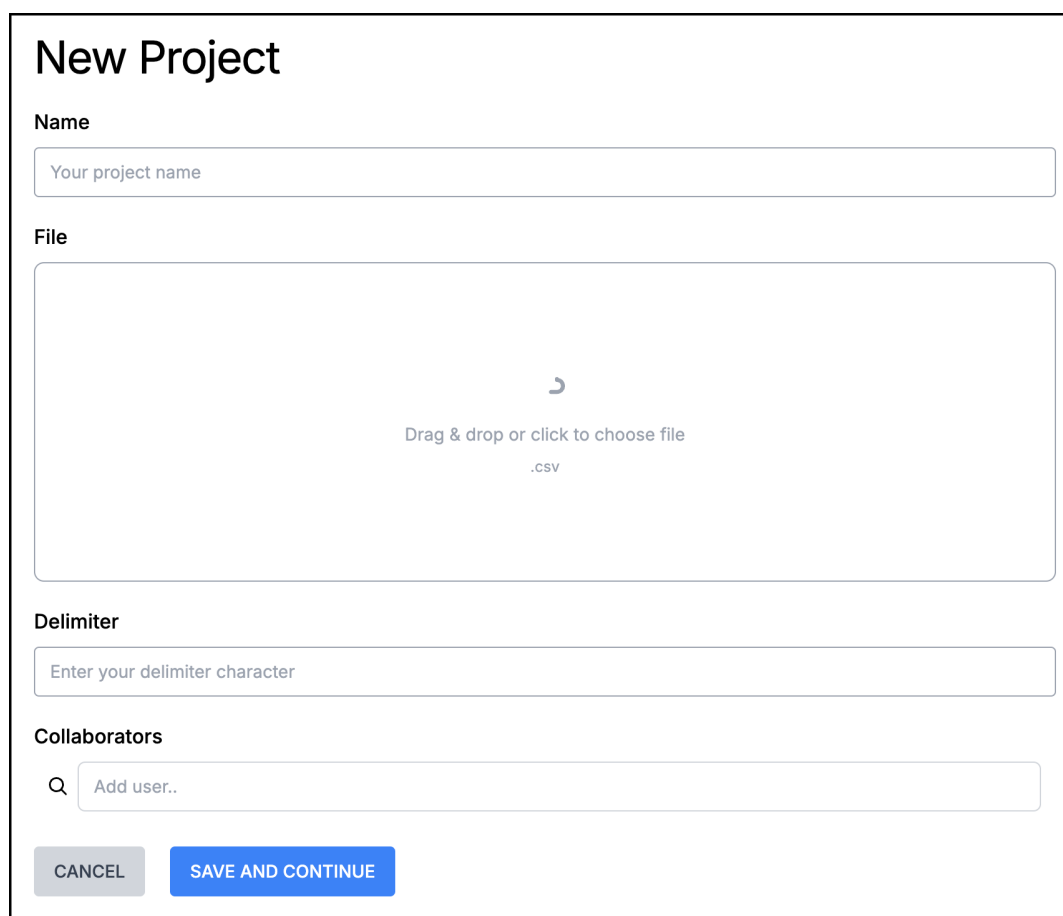
- Ime projekta – naziv projekta koji omogućuje jednostavnu identifikaciju.
- Datoteka – CSV datoteka s eksperimentalnim podacima.
- Graničnik – znak koji razdvaja podatke u CSV datoteci.
- Sudionici u projektu – e-mail adrese korisnika koje želimo dodati kao sudionike.

Datoteku je moguće učitati klikom na polje za odabir datoteke ili povlačenjem

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

datoteke u navedeno polje, a podržan format datoteke je CSV. Nakon učitavanja, prikazano na slici 4.7, vidljiv je naziv datoteke, njezina veličina i traka napretka. Nakon uspješnog prijenosa datoteke, korisnik može poništiti odabir u slučaju greške.

Ispravan unos graničnika je ključan jer se koristi u sljedećem koraku prilikom postavljanja opcija za grafički prikaz. Na kraju, korisnici mogu pretražiti i dodati sudionike unosom njihovih e-mail adresa. Iz sigurnosnih razloga pretraga zahtijeva unos pune e-mail adrese kako bi se identificirali točni korisnici.



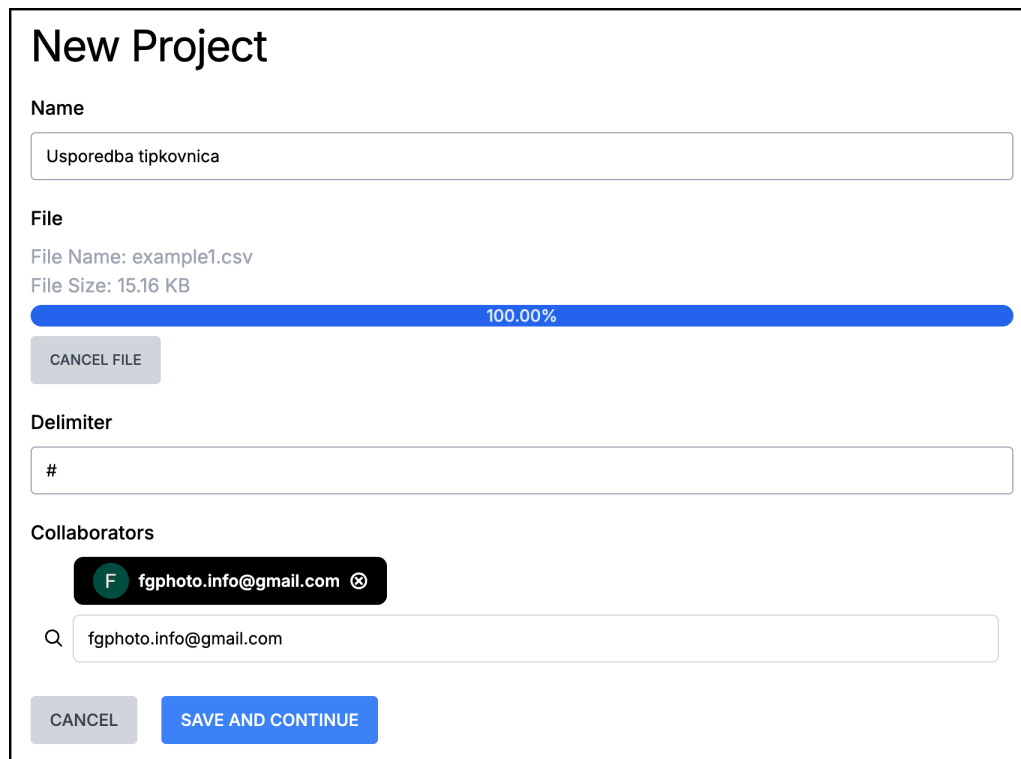
The image shows a web form titled "New Project". It contains the following sections:

- Name:** A text input field with the placeholder text "Your project name".
- File:** A large rectangular area with a curved arrow icon and the text "Drag & drop or click to choose file" and ".csv" below it.
- Delimiter:** A text input field with the placeholder text "Enter your delimiter character".
- Collaborators:** A search input field with a magnifying glass icon and the placeholder text "Add user..".

At the bottom of the form, there are two buttons: a grey "CANCEL" button and a blue "SAVE AND CONTINUE" button.

Slika 4.6 Prikaz praznog obrasca za unos općenitih informacija o projektu

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja



The image shows a 'New Project' form with the following sections:

- Name:** A text input field containing 'Usporedba tipkovnica'.
- File:** Displays 'File Name: example1.csv' and 'File Size: 15.16 KB'. Below this is a blue progress bar at 100.00% and a 'CANCEL FILE' button.
- Delimiter:** A text input field containing '#'.
- Collaborators:** Shows a list of collaborators with one entry: 'F fgphoto.info@gmail.com'. Below the list is a search input field containing 'fgphoto.info@gmail.com'.

At the bottom of the form are two buttons: 'CANCEL' and 'SAVE AND CONTINUE'.

Slika 4.7 Primjer ispunjenog obrasca za unos općenitih informacija o projektu

4.3.2 Opcije grafičkog prikaza

U ovom koraku korisnik definira sve opcije prikaza grafa kako bi omogućio detaljnu deskriptivnu i inferencijalnu statistiku za odabrane parametre. Potrebno je unijeti podatke u sljedeća polja:

- Naslov grafa
- Tip grafa – dostupne opcije uključuju histogram, linijski, multilinijski, stupčasti i kutijasti graf.
- Naslov stupca na x-osi
- Naslov stupca na y-osi
- Sortiranje – opcija dostupna samo kod stupčastog grafa.

Ovisno o odabiru vrste grafa, obrazac se automatski prilagođava i primjenjuje

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

odgovarajuću validaciju podataka. U slučaju neispravnog unosa, aplikacija prikazuje poruku o grešci iznad polja u kojem je došlo do problema, olakšavajući korisniku ispravak. U nastavku su opisane karakteristične situacije za pojedine vrste grafikona:

- Histogram – Za ovaj graf potrebno je unijeti samo naziv stupca na x-osi, koji mora sadržavati numeričke vrijednosti. Polje za unos naslova stupca na y-osi je onemogućeno.
- Stupčasti graf – Naslov stupca na x-osi treba odgovarati tipu podataka koji sadrži tekstualne vrijednosti, budući da x-os prikazuje različite grupe. Stupac na y-osi mora sadržavati numeričke vrijednosti. Također, korisnik može odabrati opciju sortiranja vrijednosti na x-osi (abecedno ili obrnuto) ili na y-osi (uzlazno ili silazno).
- Linijski i multilinijski graf – Stupac na x-osi mora sadržavati vremenske oznake jer se linijski graf koristi za prikaz promjena varijable kroz vrijeme, dok stupac na y-osi treba sadržavati numeričke vrijednosti. Kod multilinijskog grafa, moguće je odabrati više stupaca za y-os, a dodavanje nove vrijednosti na y-os omogućeno je putem gumba "*Add new Y value*". Kod dodavanja novog polja onemogućene su vrijednosti koje su već odabrane, te je dodatna polja moguće i ukloniti.
- Kutijasti graf – Ovdje x-osi prikazuje grupne podatke, pa stupac mora sadržavati tekstualne nizove, dok vrijednosti na y-osi trebaju biti numeričke.

Slike 4.8, 4.9 i 4.10 prikazuju primjere unosa za stupčasti i multilinijski graf, kao i primjer obavijesti o pogrešci pri unosu.

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

Graph options

Title

Primjer kutijastog grafa

Graph Type

Box-plot

X value

Keyboard type

Data type: string

Y value

WPM: Words per minute

Data type: number

Back SAVE AND CONTINUE

Slika 4.8 Primjer ispunjenog obrasca o opcijama vizualnog prikaza za stupčasti graf

Graph options

Title

Histogram primjer

Graph Type

Histogram

X value

Timestamp

Data type: number

Y value

Select Y-axis

Error: X axis values are not of type number.

Back SAVE AND CONTINUE

Slika 4.9 Primjer obavijesti o pogrešci na x-osi

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

Graph options

Title

Graph Type

X value

Data type: timestamp

Y value

Data type: number

Y value

Data type: number

Slika 4.10 Primjer ispunjenog obrasca o opcijama vizualnog prikaza za multilinijski graf

4.4 Pregled grafova i deskriptivne statistike

Primjer generiranja grafova prikazan je kroz HCI istraživanje koje uspoređuje učinkovitost unosa teksta pomoću dvije tipkovnice. U istraživanju su mjerene performanse korisnika dok su unosili određene rečenice koristeći svaku od tipkovnica, uz bilježenje različitih parametara kao što su naziv tipkovnice, ime korisnika, broj znakova u sekundi, broj pritisaka po sekundi i broj riječi po minuti. Testirane tipkovnice označene su kao "Samsung" i "OTA".

Cilj istraživanja je u tom oglednom primjeru utvrditi koja tipkovnica omogućava brže i intuitivnije pisanje. Ključna metrika za ovu analizu je broj riječi po minuti ("Words per Minute"). Na slici 4.11 prikazan je stupčasti grafikon, na kojem vidimo rezultate za ovu metriku.

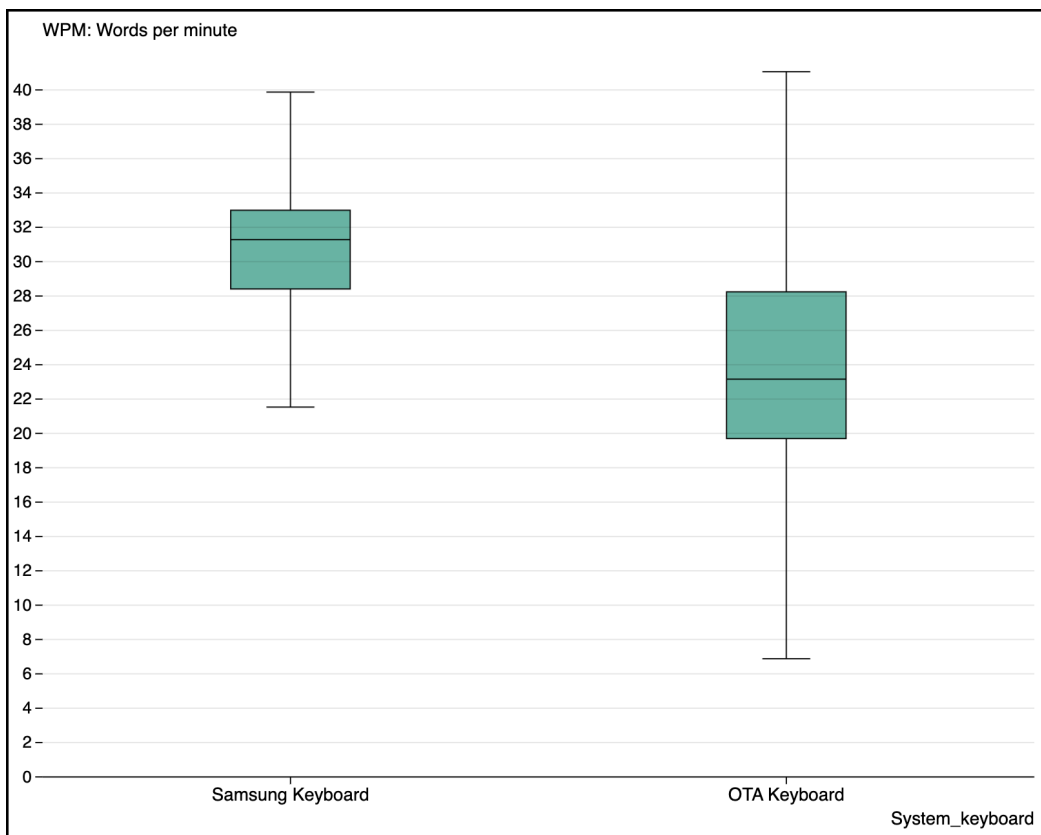


Slika 4.11 Primjer generiranog stupčastog grafa

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

Iznad grafika nalaze se gumbi za povratak na uređivanje osnovnih informacija o projektu ili opcija grafičkog prikaza, kao i opcija za preuzimanje grafika u .png formatu. Grafikon prikazuje dva stupca, svaki za jednu tipkovnicu, s naznačenim prosječnim brojem riječi po minuti. Vidljivo je da tipkovnica "Samsung" ima veću prosječnu vrijednost. Crvene linije označavaju interval varijabilnosti podataka, pri čemu je varijabilnost kod tipkovnice "OTA" izraženija. Uz to, na grafikonu su jasno prikazane skale i oznake za x i y osi, zajedno s nazivima odabranih stupaca.

Za iste podatke možemo generirati i kutijasti graf, koji je prikazan na slici 4.12, te dodatno ilustrira razliku u varijabilnosti između dvije tipkovnice. Brkovi na grafu jasno pokazuju znatnu razliku u rasponima vrijednosti. Središnji kutijasti dio grafa otkriva razlike u interkvartilnim rasponima i medijanima, što podržava zaključak sa stupčastog grafa: tipkovnica "Samsung" pokazuje stabilnije performanse s manjom varijabilnošću.



Slika 4.12 Primjer generiranog kutijastog grafa

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

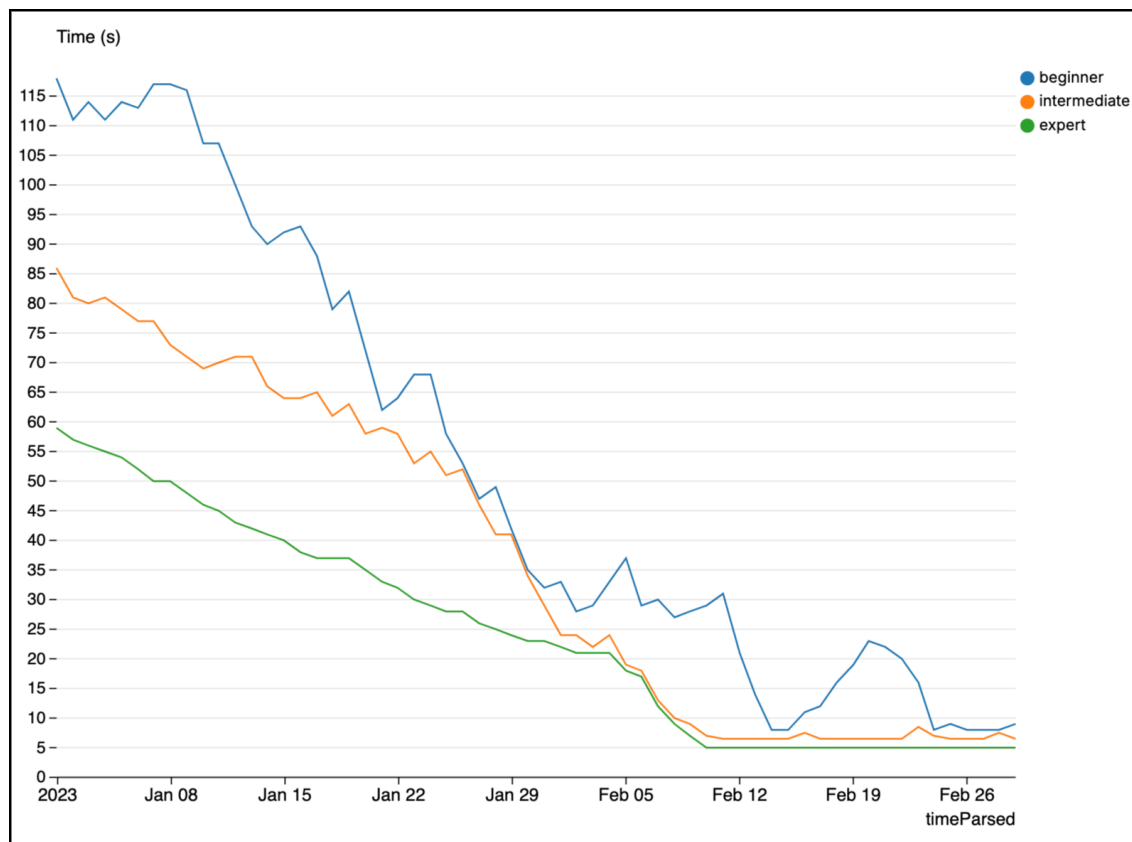
Ispod grafikona prikazane su vrijednosti deskriptivne statistike koje omogućuju dodatno potvrđivanje vizualnih zapažanja broječanim podacima te ih možemo vidjeti na slici 4.13. Prikazane su sljedeće statističke varijable za svaku grupu podataka: aritmetička sredina, medijan, standardna devijacija, asimetrija, minimalna i maksimalna vrijednost. Na primjer, medijan za tipkovnicu "Samsung" pokazuje 8,12 riječi po minuti više nego kod tipkovnice "OTA", što podupire zaključke iz grafičkog prikaza. Osim toga, raspon između minimalne i maksimalne vrijednosti za tipkovnicu "OTA" znatno je veći nego za "Samsung", što upućuje na veću varijabilnost u performansama kod tipkovnice "OTA".

Statistics:	
WPM: Words per minute	
<u>Samsung Keyboard</u>	<u>OTA Keyboard</u>
Mean: 30.6025	Mean: 23.9085
Median: 31.2850	Median: 23.1600
Standard deviation: 2.9544	Standard deviation: 4.4729
Skewness: -0.6930	Skewness: 0.5020
Min: 25.0700	Min: 18.1000
Max: 34.7000	Max: 31.0300

Slika 4.13 Generirana deskriptivna statistika za usporedbu tipkovnica

Kako bi se demonstrirala primjena linijskog grafa, koriste se simulirani podaci koji prikazuju brzinu učenja korisnika nove funkcionalnosti programa, mjerenu brzinom izvršavanja zadatka. U istraživanju su testirana tri tipa korisnika: početnik, napredni i eksperti. Za vizualni prikaz deskriptivne statistike izabran je multilinijski graf, s datumom na x-osi i trima linijama koje predstavljaju svakog od korisnika. Na slici 4.14 prikazan je multilinijski graf, koji jasno ilustrira najbrži pad vremena izvršavanja kod početnika, što odražava strmu krivulju učenja. Srednji pad zabilježen je kod naprednog korisnika, dok je najblaži trend prisutan kod eksperta. U gornjem desnom kutu nalazi se legenda koja objašnjava prikazane linije, dok je na dnu grafikona prikazan vremenski interval.

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja



Slika 4.14 Primjer generiranog multilinijskog grafa

4.5 Statistički testovi i post-hoc analize

Kao što je prethodno objašnjeno, u znanstvenom istraživanju nije dovoljno jednostavno usporediti dvije varijable ili grupe na temelju osnovnih statističkih mjera poput srednje vrijednosti. Potrebno je primijeniti statističke testove kako bi se ustanovilo postoji li statistički značajna razlika između njih. Bez odgovarajuće statističke analize, svaki uočeni obrazac ili razlika može biti posljedica slučajnosti, što smanjuje pouzdanost zaključaka i čini ih manje korisnima za širu primjenu, odnosno generalizaciju. Aplikacija omogućuje izvođenje različitih statističkih testova prilikom usporedbe dvije ili više vrijednosti. Odabir odgovarajućeg testa ovisi o načinu provedbe istraživanja, uključujući je li tip eksperimenta bio s nezavisnim uzorcima, s ponovljenim mjerenjima ili jesu li podaci normalno distribuirani. Aplikacija podržava sljedeće testove: T-test za nezavisne uzorke, T-test za zavisne uzorke, Wilcoxonov test,

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

Mann-Whitneyev test, jednosmjernu analizu varijance, analizu varijance ponovljenih mjerenja, Friedmanov test i odgovarajuće post-hoc testove.

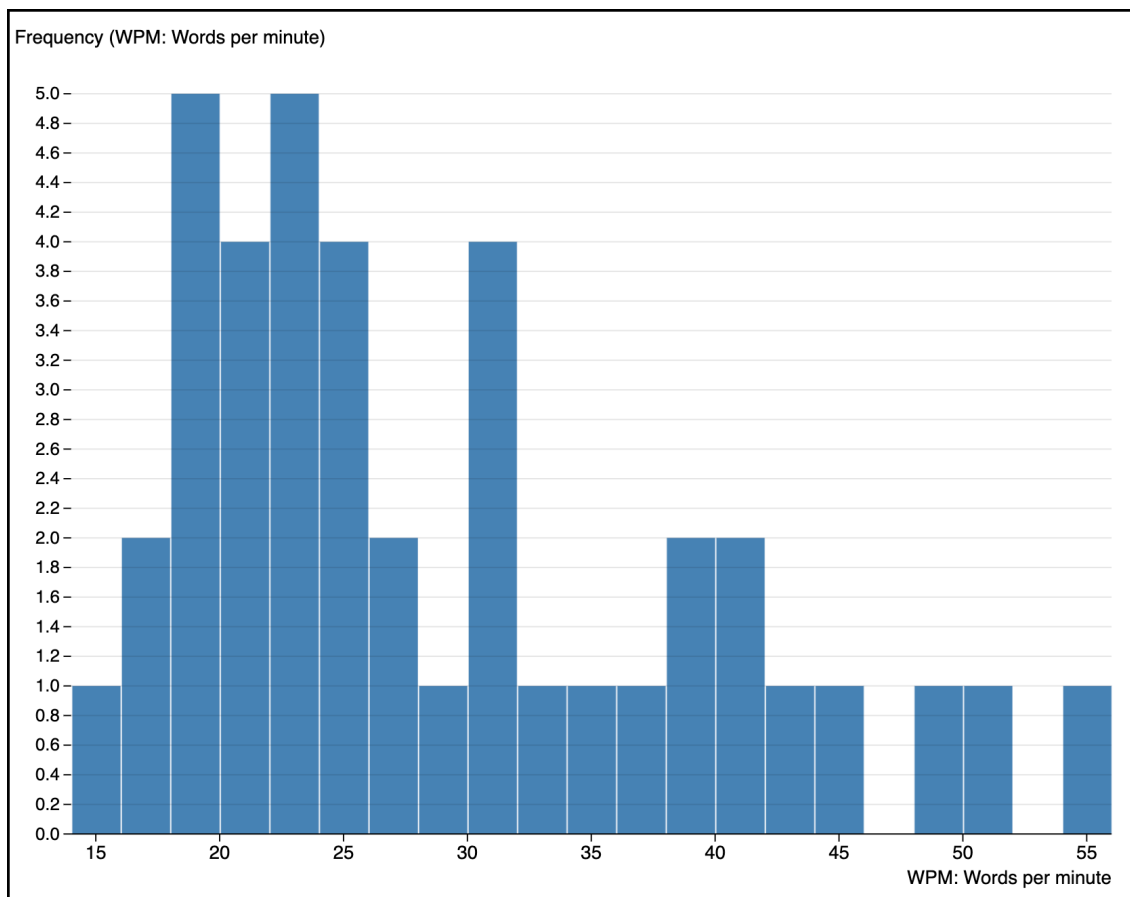
Za odabir statističkog testa potrebno je unijeti sljedeće informacije:

- Razina značajnosti, odnosno p-vrijednost koja služi kao gornja granica za zaključivanje o značajnosti razlika.
- Broj grupa - izbor između dvije ili tri i više grupa.
- Tip uzoraka - mogu biti nezavisni ili zavisni.

Na temelju odabranih opcija, aplikacija automatski prilagođava dostupne testove kako bi olakšala korisniku izbor pravog statističkog testa. Također, potrebno je odabrati koje grupe želimo usporediti. Ako uspoređujemo tri ili više grupa, prikazuje se gumb "Add new Y value" za dodavanje nove vrijednosti za usporedbu. Nakon što su sve vrijednosti odabrane, zahtjev se šalje na Flask API za izvršenje odabranog testa.

U oglednom primjeru usporedbe tipkovnica, ispitanici su testirani unutar iste skupine, što zahtijeva korištenje zavisnog statističkog testa. Prije odabira odgovarajućeg testa, ključno je utvrditi je li distribucija podataka normalna. U tu svrhu možemo generirati histogram. Nakon izrade histograma, imamo mogućnost prilagoditi broj prikazanih intervala, što olakšava procjenu normalne distribucije i pomaže u smanjenju "šuma" ili maskiranja podataka. Na slici 4.15 moguće je primjetiti da je histogram asimetričan i naginje ulijevo, što sugerira da podaci nisu normalno distribuirani.

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja



Slika 4.15 Primjer generiranog histogram grafa

Pošto podaci nisu normalno distribuirani najprikladniji izbor statističkog testa je Wilcoxonov test, s granicom značajnosti p-vrijednosti od 0.001. Rezultati testa, prikazani na slici 4.16, pokazali su p-vrijednost manju od 0.001, što sugerira da je razlika statistički značajna. Dakle, ovaj test potvrdio je postojanje značajne razlike između tipkovnica, pri čemu korisnici "Samsung" tipkovnice postižu brže utipkavanje.

Kako bi se prikazao rezultat statističkog testa s tri grupe, stvoren je novi projekt s dodatnim podacima za treću tipkovnicu, nazvanu "Test". Na slici 4.17 prikazan je stupčasti graf koji ilustrira aritmetičke sredine učinkovitosti unosa za sve tri tipkovnice. Vizualnim pregledom moguće je uočiti da brzina unosa tipkovnicom "Samsung" ima najveću aritmetičku sredinu, slijedi je tipkovnica "Test", dok tipkovnica "OTA" ima najnižu aritmetičku sredinu. Kako bismo provjerili postoji li statistički

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja

Statistical tests

Significance level (p-value)	Group count	Data type	Statistical test
0.001 (0.1%)	2	Paired	Wilcoxon

Selected data 1

Samsung Keyboard

Selected data 2

OTA Keyboard

GENERATE TEST

Result:

W-statistic: **3.0000**
p_value: **< 0.0001**
result: **W = 3.000, p = 0.000**
test: **Wilcoxon**

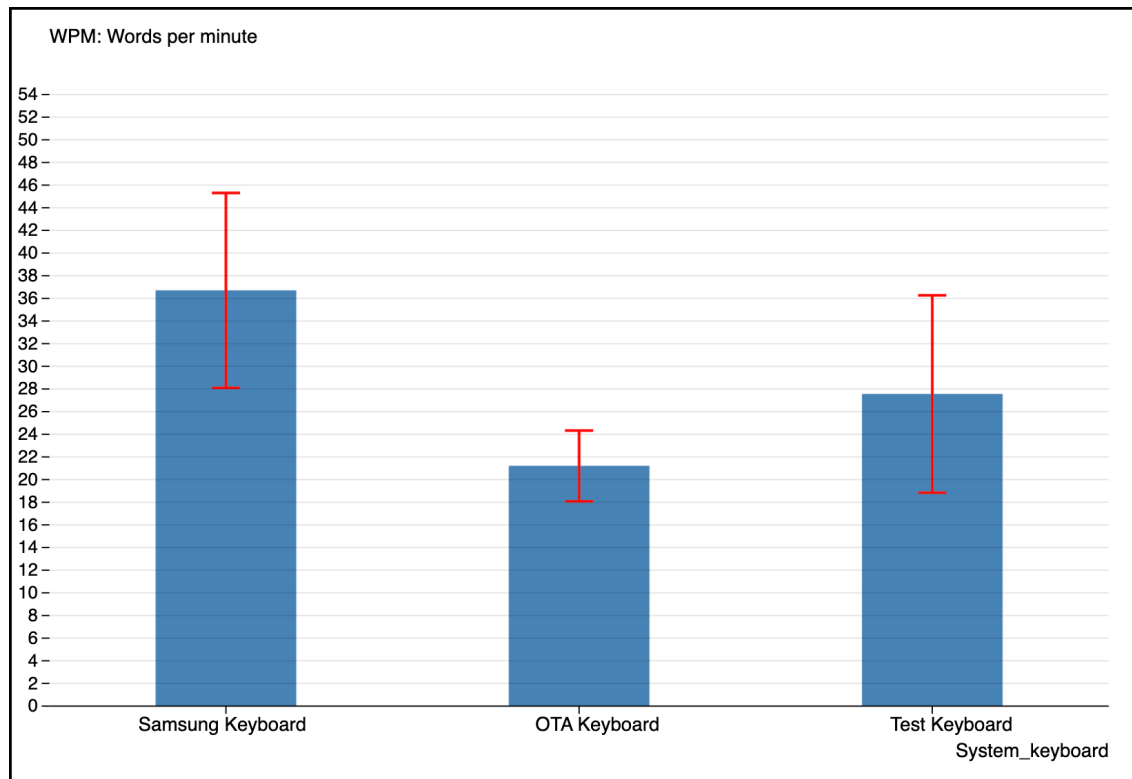
P-value is smaller than 0.001, therefore there is **significant** difference between the data.

Slika 4.16 Rezultati Wilcoxonovog testa za usporedbu učinkovitosti unosa dvije tipkovnice

značajna razlika između brzine unosa kod ovih tipkovnica, primijenjen je Friedmanov test. Rezultat testa (Slika 4.18) ukazuje na značajnu razliku između tipkovnica ($p < 0.0001$).

Kada testiramo tri ili više grupa i utvrdimo statistički značajnu razliku, nužno je provesti *post hoc* analizu kako bismo identificirali između kojih grupa postoji ta razlika. U ovom slučaju korišten je Wilcoxonov test, a rezultati su prikazani na slici 4.19. Sve p-vrijednosti su manje od 0.001, što potvrđuje da postoji statistički značajna razlika između svih uspoređenih tipkovnica. To nam omogućuje zaključiti da je brzina unosa teksta signifikantno različita kod korištenja svake od tipkovnica, pri čemu tipkovnica "Samsung" značajno nadmašuje ostale.

Poglavlje 4. Prikaz karakterističnih slučajeva korištenja



Slika 4.17 Prikaz stupčastog grafa za usporedbu 3 tipkovnice

Result:

chi_squared_statistic: **24.1000**

p_value: **< 0.0001**

result: **$\chi^2(2) = 24.100, p < 0.0001$**

test: **Friedman Test**

P-value is smaller than 0.05, therefore there is a **significant** difference between the data.

Slika 4.18 Prikaz rezultata Friedmanovog testa

Post hoc analysis

Post hoc test

Wilcoxon ▾

GENERATE POST HOC

Result:

Post Hoc Test Results

Comparison 1

Groups: Samsung Keyboard vs. OTA Keyboard
p-Value: < 0.0001
Test statistic: 0.0000

Comparison 2

Groups: Samsung Keyboard vs. Test Keyboard
p-Value: 0.0009
Test statistic: 21.0000

Comparison 3

Groups: OTA Keyboard vs. Test Keyboard
p-Value: 0.0049
Test statistic: 32.0000

Slika 4.19 Prikaz rezultata post hoc analize (međusobne usporedbe Wilcoxonovim testom)

Poglavlje 5

Zaključak

U ovom diplomskom radu predstavljena je web aplikacija za deskriptivnu i inferencijalnu analizu podataka iz empirijskih istraživanja u području interakcije čovjeka i računala. Aplikacija je razvijena s naglaskom na funkcionalne zahtjeve koji omogućuju korisnicima učinkovito upravljanje i analizu podataka. Uvođenje podataka putem CSV datoteka, zajedno s automatskim generiranjem deskriptivnih statistika, omogućava korisnicima brz i jednostavan način za analizu svojih podataka.

Deskriptivna statistika pruža temeljan uvid u podatke, dok inferencijalna statistika omogućuje korisnicima primjenu različitih statističkih testova, što dodatno obogaćuje analitičke mogućnosti aplikacije. Upravljanje korisničkim podacima i projektima putem višekorisničkog sustava osigurava suradnju i dijeljenje informacija među istraživačima, čime se potiče timski rad i razmjena znanja.

Tehnološka arhitektura aplikacije, koja kombinira Next.js, MongoDB, Flask i D3.js, osigurava visoku razinu performansi i fleksibilnosti. Korištenje ovih modernih tehnologija poboljšava korisničko iskustvo, omogućava intuitivno sučelje i dinamičke vizualizacije podataka.

S obzirom na sve navedeno, aplikacija predstavlja odgovarajući doprinos području HCI istraživanja, omogućavajući istraživačima ne samo analizu podataka, već i suradnju s kolegama na zajedničkim projektima, što može dovesti do novih uvida i otkrića u ovom dinamičnom području. Daljnji razvoj i implementacija dodatnih funkcionalnosti, poput naprednijih analitičkih alata i vizualizacija, mogli bi dodatno unaprijediti aplikaciju i proširiti njenu primjenu u istraživačkim projektima.

Bibliografija

- [1] I. S. MacKenzie, *Human-computer interaction: An empirical research perspective*. Elsevier, 2024.
- [2] J. M. Carroll, “Human computer interaction (HCI),” *Encyclopedia of Human-Computer Interaction*. Aarhus: The Interaction Design Foundation, 2009.
- [3] J. Lazar, J. H. Feng i H. Hochheiser, *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.
- [4] A. Field, *Discovering statistics using IBM SPSS statistics*. Sage publications limited, 2024.
- [5] D. C. Howell, *Statistical methods for psychology*. PWS-Kent Publishing Co, 1992.
- [6] B. C. Beins i M. A. McCarthy, *Research methods and statistics*. Cambridge University Press, 2017.
- [7] C. Alacaci, “Inferential statistics: Understanding expert knowledge and its implications for statistics education,” *Journal of Statistics Education*, sv. 12, br. 2, 2004.
- [8] J. M. Zhang, F. Li, D. Hao i dr., “A study of bug resolution characteristics in popular programming languages,” *IEEE Transactions on Software Engineering*, sv. 47, br. 12, str. 2684–2697, 2019.
- [9] A. Fedosejev, *React. js essentials*. Packt Publishing Ltd, 2015.
- [10] Vercel, *Next.js Introduction*, <https://nextjs.org/docs>, Preuzeto: 2024-10-30.
- [11] D. Ghimire, “Comparative study on Python web frameworks: Flask and Django,” 2020.

BIBLIOGRAFIJA

- [12] IBM, *What is MongoDB?* <https://www.ibm.com/topics/mongodb>, Preuzeto: 2024-10-29.
- [13] N. Q. Zhu, *Data visualization with D3.js cookbook*. Packt Publishing Ltd, 2013.
- [14] Clerk, *Welcome to Clerk Docs*, <https://clerk.com/docs>, Preuzeto: 2024-10-30.

Sažetak

U području interakcije čovjeka i računala (HCI), različite inačice korisničkih sučelja i različiti modaliteti interakcije se uobičajeno vrednuju provođenjem odgovarajućeg eksperimenta. Pri tome, izlazni podaci iz takvog eksperimenta predstavljaju osnovu za provođenje detaljne statističke analize. U ovome radu razvijena je web aplikacija za obradu postojećih dnevnčkih zapisa iz empirijskih HCI istraživanja. Aplikacija je zasnovana na modernom tehnološkom stogu i omogućava: (1) deskriptivnu analizu podataka zasnovanu na interaktivnom konfiguratoru grafičkog prikaza, (2) inferencijalnu analizu podataka provođenjem odgovarajućih statističkih testova, te (3) višekorisnički kontekst - kroz upravljanje korisnicima i zajednički pristup arhivi prethodno spremljenih statističkih analiza.

Ključne riječi — web aplikacija, statistička analiza, deskriptivna analiza podataka, inferencijalna analiza podataka, interakcija čovjeka i računala (HCI)

Abstract

In the field of Human-Computer Interaction (HCI), different versions of user interfaces and interaction modalities are usually evaluated by conducting an appropriate experiment. The output data of such an experiment forms the basis for a detailed statistical analysis. In this thesis, a web application was developed to process existing log files from empirical HCI research. The application is based on a modern technological stack and enables: (1) descriptive data analysis based on an interactive graphical display configurator, (2) inferential data analysis by performing appropriate statistical tests, and (3) multi-user context - through user management and shared access to the archive of previously saved statistical analyses.

Keywords — web application, statistical analysis, descriptive data analysis, inferential data analysis, Human-Computer Interaction (HCI)