ANALIZA PODATAKA U PYTHONU

Sesar, Andrea

Undergraduate thesis / Završni rad

2023

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:228:335592

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-22**

Repository / Repozitorij:

Repository of University Department of Professional Studies

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Informacijska tehnologija

ANDREA SESAR

Z A V R Š N I R A D

ANALIZA PODATAKA U PYTHONU

Split, rujan 2023.

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Informacijska tehnologija

Predmet: Sigurnost računala i podataka

Z A V R Š N I R A D

Kandidat: Andrea Sesar

Naslov rada: Analiza podataka u Pythonu

Mentor: Lada Sartori, viši predavač

Split, rujan 2023.

SADRŽAJ

SAŽETAK **2**

SUMMARY **2**

1.UVOD **4**

2. TEHNOLOGIJE **5**

2.1.Programski jezik Python 5

2.2. Streamlit 5

2.2.1. Instalacija razvojnog okvira Streamlit 6

2.2.2. Grafički elementi 7

2.3. Matplotlib i Seaborn 7

2.4. SQLite 8

2.5. DB Browser 9

2.6. Visual Studio Code 10

2.7. Virtualno okruženje 10

2.8. Grafički prikazi i njihovo kreiranje 11

2.8.1. Grafikon 12

2.8.2. Histogram 12

2.8.3. Dijagram raspršenosti 13

2.8.4. Kružni graf 13

2.8.5. *Box plot* 14

3. FUNKCIONALNOSTI APLIKACIJE **16**

3.1. Kreiranje korisnika 16

3.2. Kriptiranje lozinke 17

4. NAČIN RADA APLIKACIJE **19**

4.1. Administrativno sučelje 19

4.1.1. Dodavanje podataka 19

4.1.2. Ažuriranje podataka 21

4.1.3. Brisanje podataka 22
4.2. Korisničko sučelje 23
4.2.1. Grafički prikazi podataka o putniku 24
5. ZAKLJUČAK 38
LITERATURA 39
POPIS SLIKA 40

Sažetak

Cilj ovog završnog rada je pružiti jasan vizualni prikaz podataka o putovanjima, omogućavajući korisnicima jednostavnu analizu i istraživanje. Aplikacija se sastoji od dva ključna dijela: korisničkog sučelja i administratorskog sučelja.

Poslužiteljski dio aplikacije (engl. *backend*) izrađen je korištenjem programskog jezika Python. Za manipulaciju podataka i izradu korisničkog sučelja (engl. *frontend*) korištena je *Streamlit* biblioteka, kroz koju je korištenjem ostalih biblioteka programskog jezika *Python,* poput biblioteke *Pandas,* omogućeno upravljanje, filtriranje i analiza podataka. Vizualizacija podataka provedena je koristeći *Plotly Express*, pomoću kojeg su izrađeni grafički prikazi. Baza podataka je implementirana pomoću relacijske baze podataka *SQLite*, koja se temelji na sučelju strukturiranog jezika upita (engl. *Structured Query Language - SQL)*. Za pregled baze podataka korišten je *DB browser*.

Razvojni alat Visual Studio Code (dalje u tekstu: *VS Code*) korišten je kao glavni uređivač kôda kroz cijelu izradu projekta.

Ključne riječi: *Python, Streamlit, SQLite, Pandas, Plotly Express*

Summary

Data analysis using the programming language Python

The main goal of this thesis is to provide a clear visual representation of travel data, in a simpler and better way for users. The application consists of two parts meant to be used by the user and administrator.

The backend part of the application was created using the Python programming language. The Streamlit library is used for data manipulation and creation of the user interface (frontend), through which other libraries of the Python programming language are used, such as the Pandas library, enabling data management, filtering, and analysis.

The database is implemented using the SQLite relational database, which is based on the Structured Query Language (SQL) interface. DB browser is used to view the database.

The Visual Studio Code (VS Code) development tool is the primary code editor throughout the entire project creation.

Keywords: *Python, Streamlit, SQLite, Pandas, Plotly Express*

1.Uvod

Od davnih vremena putovanje je bilo poznanica u svakodnevici, a kako vrijeme ide dalje, tako postaje jedan od važnijih segmenata ljudskog života. Osim što pruža mogućnost odmora i opuštanja, uvelike doprinosi kulturnoj razmjeni, obrazovanju i napretku svake osobe. U današnjem dobu, kada digitalno sve više prevladava, u svakodnevnoj upotrebi se prikuplja velika količina podataka, kako za ostala područja tako i za stavke vezane uz putovanja. Prikupljanje takvih podataka omogućuje bolje razumijevanje potreba i zahtjeva, preferencija i trendova, te što sve može pomoći u svrhu prilagodbe i unapređenja u iznimno razvijenoj industriji putovanja.

Ovaj završni rad iznosi projekt koji koristeći različite tehnologije omogućuje vizualni prikaz podataka o putovanjima. Kroz upotrebu programskog jezika *Python* i raznih biblioteka kao što su *Streamlit, Plotly Express* i *Pandas*, razvijena je aplikacija koja korisniku omogućava pregled različitih aspekata putovanja, filtriranje podataka, ali i analizu.

U drugom poglavlju rada opisane su tehnologije korištene tijekom izrade aplikacije, zbog čega se točno koriste te usko povezane stavke uz svaku navedenu tehnologiju. U trećem poglavlju iznesene su funkcionalnosti koje su bile od veće važnosti pri izradi aplikacije. Četvrto poglavlje prikazuje način rada aplikacije, podijeljen na dva načina korištenja aplikacije, pobliže prikazan slikama u svrhu jednostavnijeg razumijevanja rada.

2. Tehnologije

2.1.Programski jezik Python

Python je moderni programski jezik koji pruža objektno orijentiran, funkcionalan i imperativan programski stil [1]. Njegova fleksibilnost i jednostavna sintaksa čine ga jednim od najčešće korištenih programskih jezika, što ga ujedno i čini jednim od najpopularnijih u svijetu.

Prvo pojavljivanje događa se tijekom 1990. godine, a stvorio ga je Guido van Rossum. Trenutna posljednja dostupna verzija je 3.11.4, te ju je moguće koristiti na svim operativnim sustavima.

Jedan od razloga zbog kojih je *Python* toliko popularan, osim svoje jednostavnosti, je taj što ga se koristi na više različitih područja. Često se koristi za analizu i obradu podataka. Biblioteke poput *Pandas, NumPy, Matplotlib* i ostalih, omogućuju vrlo jednostavno obrađivanje podataka, ali i vizualni prikaz istih. Osim toga, *Python* je postao ključan jezik u polju umjetne inteligencije i strojnog učenja, a koristi se i u području bioinformatike, što je još samo jedna dodatna potvrda kako se *Python* zaista koristi u više različitih područja.

2.2. Streamlit

Streamlit je *Python* biblioteka otvorenog kôda (engl. *open-source*) koja se vrlo često koristi za izrade interaktivnih web aplikacija, koristeći jednostavan kôd [2]. Zbog jednostavnosti koju pruža za analizu podataka, vizualizaciju, ali ujedno i dijeljenje rezultata te brzu izradu interaktivnih alata bez potrebe za većim znanjem internetskog razvoja, ova biblioteka je vrlo često korištena među osobama koji se bave analizom podataka, znanstvenim istraživanja te strojnim učenjem.

Kao i programski jezik *Python,* podrazumijeva se da i njegove biblioteke imaju karakteristiku jednostavne sintakse, što je naravno i jedna od glavnih prednosti korištenja *Streamlit* biblioteke. *Streamlit* podržava razne Python biblioteke, pa tako i već navedene, vrlo često korištene poput *Pandas, Matplotlib, Plotly* i druge.

Zbog spomenute jednostavne sintakse, omogućava korisniku vrlo brzu izradu web aplikacije i sučelja (engl. *frontend*), do te razine da već samo par linija kôda može rezultirati interaktivnom aplikacijom. Važno je istaknuti kako je jedna od najvećih prednosti vizualizacija podataka, gdje je moguće kroz razne grafove, mape i ostale interaktivne elemente vrlo dosljedno i detaljno prikazati podatke, ali ujedno i omogućiti korisnicima da te podatke istražuju. Proces istraživanja podataka je uvelike olakšan zbog mogućnosti izrade raznih interaktivnih elemenata (engl. *widget*), poput dugme, klizača, polja za unos i drugo.

2.2.1. Instalacija razvojnog okvira Streamlit

Prije samog početka instalacije, potrebno je imati integrirano razvojno okruženje (engl. *Integrated Development Environment, IDE*) ili tekstualni uređivač (engl. *editor*) po želji. Osim toga, potrebno je imati instaliranu podršku za programski jezik *Python*. Važno je naglasiti kako *Python* mora imati verziju 3.8 - 3.11. *Streamlit* još zahtijeva prethodno instaliran upravitelj paketa za *Python* pakete (engl. PIP). Uobičajeno je da je PIP već instaliran prilikom instalacije programskog jezika *Python*.

Proces instalacije je vrlo jednostavan i ne zahtjeva puno vremena, međutim tijekom instalacije dolazi do manjih poteškoća.

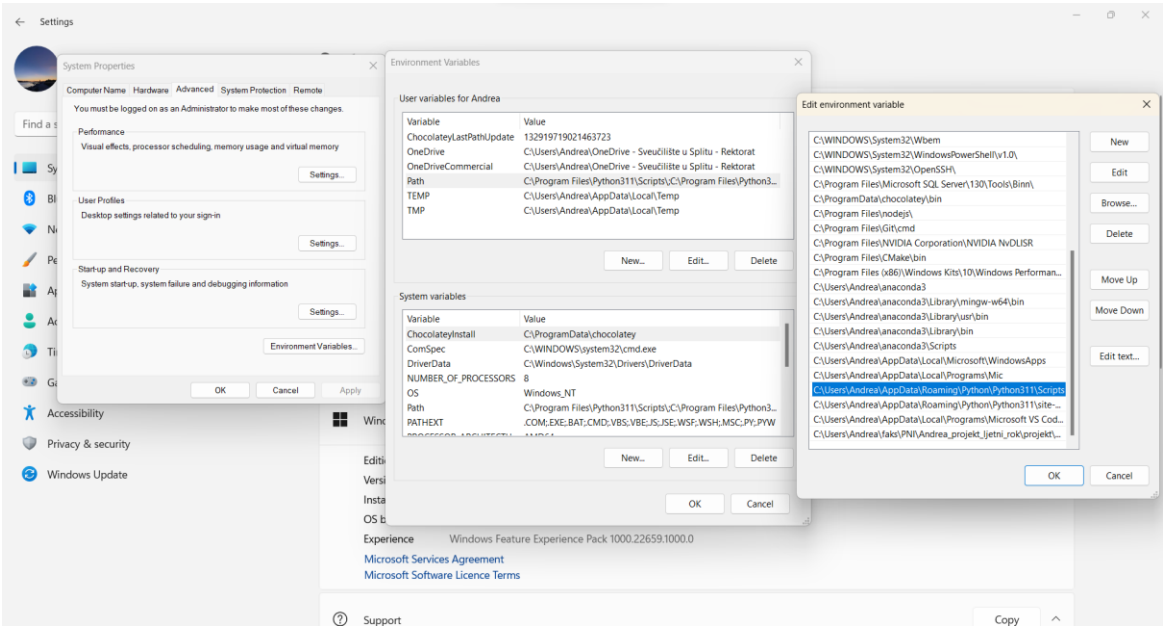
Prilikom pokušaja prvog korištenja i pokretanja programa dolazi do greške. Kada korisnik pokrene naredbu s ključnom riječi "*streamlit*", bilo to u uređivaču teksta ili naredbenom retku, dobit će povratnu poruku kako navedena naredba nije prepoznata. Ovaj problem se događa na Windows računalima. Program ne pronalazi putanju (engl. *path*) do mjesta tj. mape na kojem je instaliran *Streamlit*. Kako bi se ovaj problem riješio, potrebno je ručno unošenje putanju do mape, a to je moguće izvršiti na način opisan u nastavku.

Prvenstveno, potrebno je znati točnu lokaciju, odnosno putanju do mape gdje je *Streamlit* instaliran. Potom u postavkama računala se ide do sljedeće lokacije: `sustav -> o sustavu -> napredne postavke sustava -> varijable okoline -> putanje` (engl. *PATH*). Nakon svega toga, aplikacija se može pokrenuti iz bilo kojeg uređivača kôda izvršavanjem naredbe koja je prikazan u ispisu 1.

```
$ streamlit run naziv_programa
```

Ispis 1: Naredba za pokretanje aplikacije

Prikaz svih potrebnih prozora koje kroz korake treba otvoriti kako bi se uspješno dodala putanja na računalo prikazan je na slici 1.



Slika 1: Prikaz zaslona prilikom dodavanje putanje

2.2.2. Grafički elementi

Streamlit sadrži grafičke elemente (engl. *widget*) koji pružaju mogućnost lakšeg unosa podataka. Sintaksa za stvaranje je vrlo jednostavna, a omogućuje korisniku direktnu interakciju s aplikacijom koristeći gumbe, klizače, unose teksta i drugo. Prilikom svakog korištenja uvijek se prvo poziva *Streamlit* naredbom „*st*“, iza čega slijedi točka, a potom naziv željenog elementa.

2.3. Matplotlib i Seaborn

Matplotlib je jedna od najpopularnijih datoteka za vizualizaciju podataka u programskom jeziku *Python*. Omogućava stvaranje različitih vrsta grafova i dijagrama te vizualni prikaz podataka. *Matplotlib* pruža širok spektar opcija za prilagodbu izgleda i stilova grafikona, čime korisnicima omogućuje kreiranje vizualizacija koje su jasne i informativne.

Seaborn je nadogradnja nad *Matplotlib* bibliotekom koja olakšava kreiranje estetski što privlačnijih grafikona [3]. Dolazi s unaprijed definiranim paletama boja i stilova, što omogućava brzo kreiranje grafičkih prikaza bez potrebe za daljnjim prilagodbama. Osim toga, *Seaborn* ima ugrađene funkcije za vizualizaciju statističkih podataka kao što su raspodjele, *box plot* i regresije. Sve to skupa ga čini izuzetno korisnim za analizu i vizualizaciju podataka.

Ove dvije datoteke zajedno čine moćan alat za sve vrste vizualizacija u području analize podataka.

2.4. SQLite

SQLite je samostalna baza podataka bez poslužitelja [4]. Zbog toga što je javna i ne traži nikakvu licencu, vrlo je jednostavna za korištenje i uključenje u aplikaciju u svrhu spremanja podataka. Popularna je za male i srednje aplikacije jer omogućava efikasno upravljanje podacima bez potrebe za konfiguracijom ili pak održavanjem odvojenog poslužiteljskog sustava.

SQLite podržava programski jezik *SQL* za upravljanje podacima, što ujedno omogućuje i kreiranje tablica, izvršavanje upita, ali i dodavanje, brisanje i izmjenu podataka. Nema posebnih ograničenja vezanih za količinu podataka s kojima može raditi, tako da podržava od malih lokalnih do većih aplikacija. Međutim, važno je naglasiti da u slučaju velikih i kompleksnih aplikacija koje zahtijevaju istovremeni pristup od strane više korisnika nije najbolja podloga za korištenje, već se u tim slučajevima najčešće koriste potpune relacijske baze podataka poput *MySQL*, *MS SQL Server* i slično.

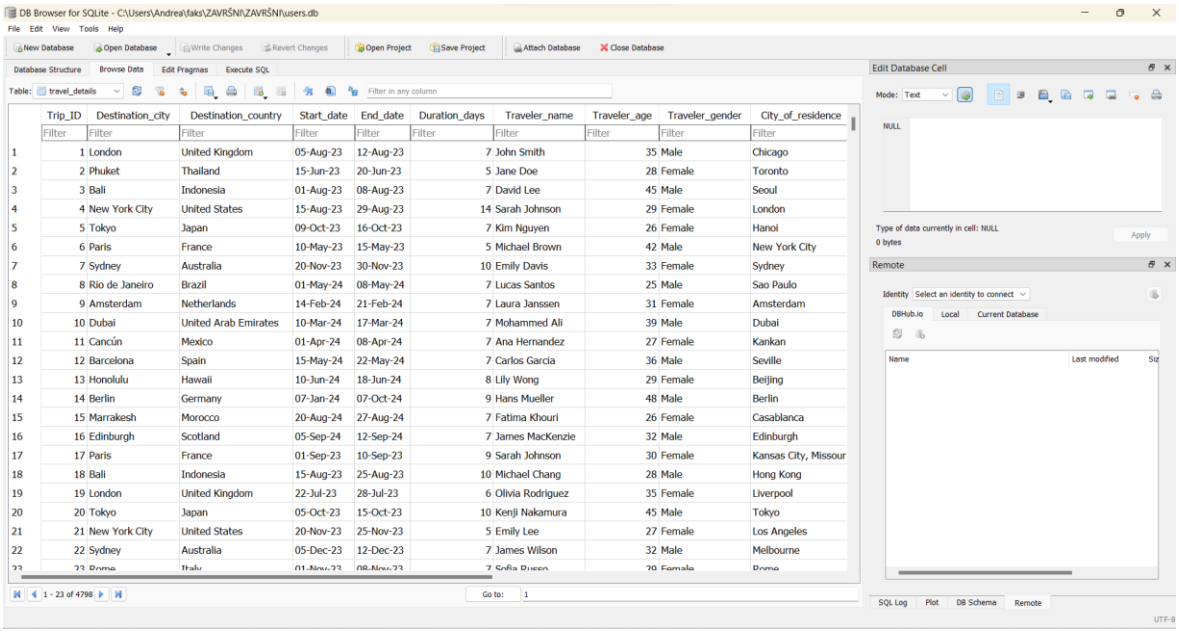
Iako se ova baza podataka vrlo često koristi prilikom rada s razvojnim okvirom *Django*, korisnika ništa ne sprječava da je koristi i u radu s bibliotekom *Streamlit*. S obzirom na to da su za ovaj rad podaci koji se analiziraju preuzeti s internetske stranice *Kaggle*, svi podaci su prikazani u datoteci s vrijednostima odvojenim zarezom (engl. *comma-separated values - CSV*). Kako bi se moglo jednostavnije manipulirati podacima, u radu se koristi biblioteka *Pandas* koja pruža mogućnost pohrane svih podataka u podatkovni okvir (engl.

DataFrame). To ujedno pruža i mogućnost povezivanja podataka s bazom SQLite, što omogućuje uvid svih podataka koristeći DB browser.

2.5. DB Browser

DB Browser za *SQLite* je kvalitetan vizualni alat za kreiranje, dizajn, i uređivanje datoteka baze podataka koje su kompatibilne sa *SQLite* [5]. Prednost korištenja DB Browsera je u tome što koristi jasna, jednostavna sučelja koja su nalik proračunskim tablicama, a za samo korištenje nije potrebno poznavanje zahtjevnih *SQL* naredbi.

Dostupan je korisnicima u svrhu kreiranja datoteke baze podataka, definiranja i mijenjanja te brisanja stranica. Omogućuje pregledavanje, uređivanje i brisanje zapisa. Pruža i mogućnost uvoza i izvoza tablica iz/u CSV datoteke. Korisnik može i izvesti jednostavne grafičke prikaze koristeći podatke iz tablica ili upita (engl. *query*). Tijekom ovog projekta koristit će se dvije tablice s podacima o korisnicima (engl. *users*) i podacima o putovanjima (engl. *travel_details*). Obje tablice su detaljnije opisane u nastavku rada. Na slici 2 moguće je vidjeti izgled tablice o putovanjima u DB browseru.



	Trip_ID	Destination_city	Destination_country	Start_date	End_date	Duration_days	Traveler_name	Traveler_age	Traveler_gender	City_of_residence
1	1	London	United Kingdom	05-Aug-23	12-Aug-23	7	John Smith	35	Male	Chicago
2	2	Phuket	Thailand	15-Jun-23	20-Jun-23	5	Jane Doe	28	Female	Toronto
3	3	Bali	Indonesia	01-Aug-23	08-Aug-23	7	David Lee	45	Male	Seoul
4	4	New York City	United States	15-Aug-23	29-Aug-23	14	Sarah Johnson	29	Female	London
5	5	Tokyo	Japan	09-Oct-23	16-Oct-23	7	Kim Nguyen	26	Female	Hanoi
6	6	Paris	France	10-May-23	15-May-23	5	Michael Brown	42	Male	New York City
7	7	Sydney	Australia	20-Nov-23	30-Nov-23	10	Emily Davis	33	Female	Sydney
8	8	Rio de Janeiro	Brazil	01-May-24	08-May-24	7	Lucas Santos	25	Male	Sao Paulo
9	9	Amsterdam	Netherlands	14-Feb-24	21-Feb-24	7	Laura Janssen	31	Female	Amsterdam
10	10	Dubai	United Arab Emirates	10-Mar-24	17-Mar-24	7	Mohammed Ali	39	Male	Dubai
11	11	Cancún	Mexico	01-Apr-24	08-Apr-24	7	Ana Hernandez	27	Female	Kankan
12	12	Barcelona	Spain	15-May-24	22-May-24	7	Carlos Garcia	36	Male	Seville
13	13	Honolulu	Hawaii	10-Jun-24	18-Jun-24	8	Lily Wong	29	Female	Beijing
14	14	Berlin	Germany	07-Jan-24	07-Oct-24	9	Hans Mueller	48	Male	Berlin
15	15	Marrakesh	Morocco	20-Aug-24	27-Aug-24	7	Fatima Khouri	26	Female	Casablanca
16	16	Edinburgh	Scotland	05-Sep-24	12-Sep-24	7	James MacKenzie	32	Male	Edinburgh
17	17	Paris	France	01-Sep-23	10-Sep-23	9	Sarah Johnson	30	Female	Kansas City, Missour
18	18	Bali	Indonesia	15-Aug-23	25-Aug-23	10	Michael Chang	28	Male	Hong Kong
19	19	London	United Kingdom	22-Jul-23	28-Jul-23	6	Olivia Rodriguez	35	Female	Liverpool
20	20	Tokyo	Japan	05-Oct-23	15-Oct-23	10	Kenji Nakamura	45	Male	Tokyo
21	21	New York City	United States	20-Nov-23	25-Nov-23	5	Emily Lee	27	Female	Los Angeles
22	22	Sydney	Australia	05-Dec-23	12-Dec-23	7	James Wilson	32	Male	Melbourne

Slika 2: Prikaz tablice o putovanjima

2.6. Visual Studio Code

VS Code je jednostavan, ali vrlo moćan uređivač kôda koji se koristi na računalima. Razvio ga je Microsoft, a dostupan je za operacijske sustave macOS, Windows i Linux [6]. Dolazi s ugrađenom podrškom za JavaScript, TypeScript i Node.js. VS Code, kako ga skraćeno nazivaju, pruža jako širok spektar mogućnosti korištenja drugih programskih jezika, zbog velikog broja proširenja (engl. *extensions*) kao što su C++, C#, PHP i raznih drugih. Naravno, pruža i mogućnost korištenja Python programskog jezika, što možemo vidjeti i u ovom radu. Osim programskih jezika, VS Code ima i ugrađenu *git* kontrolu te GitHub.

Proširenja omogućuju dodavanje novih programskih jezika, tema, programa za otklanjanje pogrešaka (engl. *debuggers*), ali i mogućnost povezivanja s dodatnim uslugama. Zbog činjenice da je sučelje prilagodljivo, korisnici se osjećaju vrlo ugodno koristeći ga jer mogu određene stavke prilagoditi svojim željama i potrebama, poput teme, prečaca na tipkovnici te ostale funkcionalnosti za koje VS Code pruža mogućnost dodatne instalacije. Proširenja se pokreću u zasebnim, odvojenim procesima, što uvelike pridodaje da neće usporiti uređivač.

Jedna od dodatnih stavki koja pruža dodatan razlog za korištenje je i mogućnost otvaranja više prozora istovremeno, što ubiti znači mogućnost rada na više aplikacija istovremeno, a više prozora može biti sačuvano kao radni prostor za daljnju ponovnu upotrebu.

Posljednja verzija VS Codea je 1.81 i izašla je u srpnju 2023. godine, te je verzija koja se koristila u ovom radu.

2.7. Virtualno okruženje

Virtualno okruženje omogućuje izolaciju i upravljanje Python projektom. Korištenje virtualnog okruženja je izrazito korisno kada se radi više projekata u programskom jeziku Python, pošto svaki može zahtijevati korištenje različitih biblioteka ili paketa. Stvaranje

virtualnih okruženja, skraćeno nazvano *venv*, omogućuje kreiranje odvojenih okruženja za svaki projekt. Tim načinom osigurava da svaki projekt koristi svoje zavisnosti.

Korištenje *venv* okruženja može osigurati da projekt ne koristi sistemsku instalaciju Pythona i njegovih biblioteka, što čini projekt urednim te osigurava lakšu distribuciju projekta drugima. Virtualno okruženje se kreira na postojeću Python instalaciju, koja je poznata kao baza okruženja. Prilikom instalacije koristi se instalacijski alat *pip*, koji služi kako bi se mogli instalirati Python paketi. Obično je *pip* automatski instaliran ako se koristi virtualno okruženje ili pak Python koji je preuzet s python.org stranice.

Kreiranje virtualnog okruženja provodi se jednostavnom naredbom, a potrebno je biti u direktoriju u kojem želimo stvoriti virtualno okruženje. Naredbe kojima se kreira i pokreće okruženje može se vidjeti u ispisu 2.

```
$ python -m venv venv

$ .\venv\Scripts\activate
```

Ispis 2: Naredbe za kreiranje i pokretanje virtualnog okruženja

Prilikom svakog sljedećeg ponovnog pokretanja projekta, potrebno je pokrenuti i virtualno okruženje jer se u protivnom aplikacija neće uspješno pokrenuti s obzirom da su svi potrebni paketi instalirati jedino u virtualnom okruženju.

2.8. Grafički prikazi i njihovo kreiranje

Prilikom izrade grafova korištene su funkcionalnosti otvorenog izvora Plotly biblioteke za Python [7]. Plotly biblioteka pruža mogućnost grafičke izrade interaktivnih, grafički visoko kvalitetnih grafova pogodnih za korištenje u svrhu vizualnog prikaza analize podataka. Postoji zavidan broj različitih primjera grafičkih prikaza koji se mogu postići korištenjem ove biblioteke. Korištenjem pod biblioteke Plotly Express izrađeni su svi grafički prikazi u radu. U aplikaciju je biblioteka uvezena pod nazivom px, pa se tako svi grafovi pozivaju tom kraticom, točkom te nazivom vrste grafičkog prikaza na engleskom jeziku. Tijekom izrade ovog rada korišteno je nekoliko mogućnosti koje su detaljno opisane u nastavku.

2.8.1. Grafikon

Grafikon (engl. *bar chart*) je interaktivno sučelje koje radi s različitim vrstama podataka koje je jednostavno grupirati u lako čitljiv grafički prikaz.

Grafikon dijeli podatke u odvojene pravokutne oznake, odnosno svaka pravokutna oznaka predstavlja podatke iste vrijednosti koje se dohvaćaju iz baze podataka. Osim toga, moguće je uključiti i više podataka koji će se zajedno prikazivati u pravokutnoj oznaci na grafikonu, a to je moguće postići spajanjem podataka, što je prikazano u ispisu 3.

```
df_common_prices =
filtered_df[filtered_df['Accommodation_cost'].isin(most_common_prices.ind
ex)]

df_common_prices = df_common_prices.sort_values(by='Accommodation_cost',
ascending=True)

fig_accomm_price = px.histogram(

      df_common_prices,

      x='Accommodation_type',

      color='Accommodation_cost',

      title='Accommodation price by accommodation type',
labels={'Accommodation_cost': 'Price'}

)

fig_accomm_price.update_xaxes(title_text="Accommodation type")

st.plotly_chart(fig_accomm_price, use_container_width=True)
```

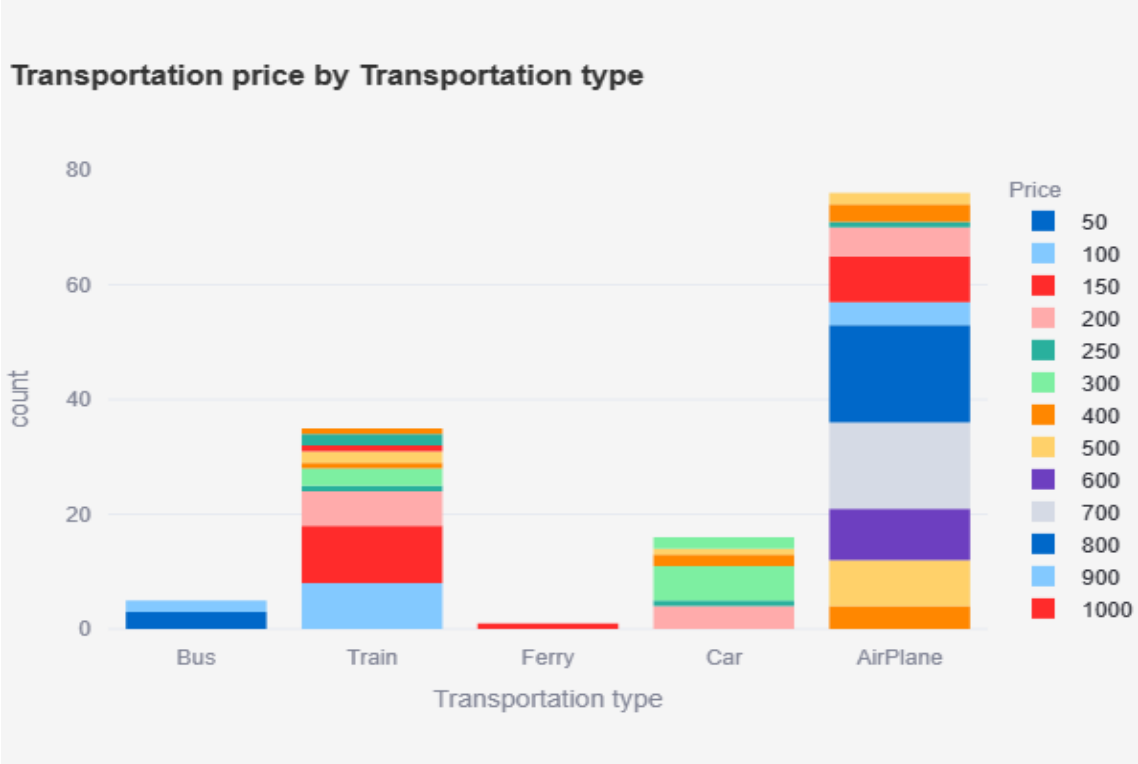
Ispis 3: Spajanje podataka za izradu histograma

Grafikon je moguće prilagoditi vlastitim željama na više načina, bilo to regulirajući visinu i širinu grafikona, varijacije i kombinacije boja koje se koriste ili pak način grupiranja podataka. Osim toga, moguće je dodati i tekst unutar grafikona.

2.8.2. Histogram

Histogram se najčešće koristi za izradu grafičkih prikaza u području statistike zbog pružanja mogućnosti zbrajanja, brojanja, izračuna prosječne vrijednosti i sličnih matematičkih funkcionalnosti. Podaci budu grupirani i prikazan je broj za svaki izvedeni

spremnik. Korištenjem ovog prikaza se također pruža mogućnost prilagodbe vizualnih aspekata poput boje, veličine i načina grupiranja podataka po želji korisnika biblioteke. Primjer histograma nalazi se na slici 3.



Slika 3: Prikaz histograma

2.8.3. Dijagram raspršenosti

Dijagram raspršenosti (engl. *scatter plot*) prikazuje svaki podatak kao zasebnu oznaku prikazanu na grafičkom prikazu. Najčešće je podatak označen kao točka, iako korisnik može mijenjati boje i oznake koje se koriste. Prikaz detaljnih podataka moguć je prelaskom preko označene točke pokazivačem na ekranu. U ovom radu je još korišten i geografski dijagram raspršenosti, koji prikazuje na geografskoj karti podatke povezane s geografskim obilježjima podataka u bazi.

2.8.4. Kružni graf

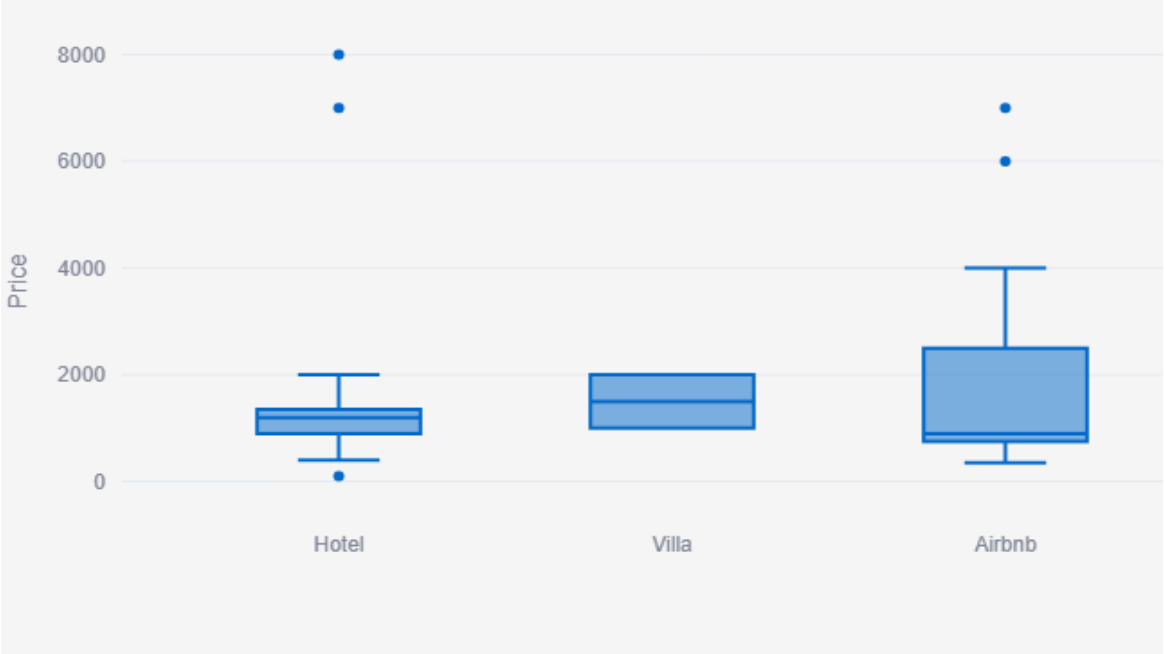
Kružni graf (engl. *pie chart*) pruža statističke podatke, prikazane u kružnom grafičkom obliku, podijeljene u sektore razdvojene bojama, kako bi jednostavno prikazao proporcionalne omjere korištenih podataka. Vrijednosti podataka prikazane su brojkama, a oznake nazivima. Primjer funkcije za izradu kružnog grafa nalazi se u ispisu 4.

```
fig_accomodation_type = px.pie(
      accommodation_type_counts,
      names=accommodation_type_counts.index,
      values=accommodation_type_counts.values,
      title="<b>Ratio & number of Accomodation by Type</b>",
      labels={'names': 'Accommodation Type', 'values': 'Count'},
      template="plotly_white"
)
st.plotly_chart(fig_accomodation_type, use_container_width=True)
```

Ispis 4: Funkcija izrade kružnog grafa

2.8.5. *Box plot*

Koristi se za statističku prezentaciju distribucije različitih vrijednosti kroz kvartile. Kvartil označava svaku od tri vrijednosti koje mogu podijeliti skupinu brojeva, poredanih od najveće prema najmanjoj, na četiri jednaka dijela. Donja granica kvartila predstavlja nižu vrijednosnu granicu, a gornji kvartil gornju graničnu vrijednost. Medijana je vrijednost koja se određuje nakon što su sve vrijednosti raspoređene te se nalazi na poziciji drugog kvartila, odnosno dijeli podatke između gornjeg i donjeg kvartila na dvije skupine. Na taj način podaci koji su izvan gornje i donje granice ne utječu na prikaz prosjeka kojem pripada najveća količina podataka. Primjer grafičkog prikaza se nalazi na slici 4.



Slika 4: Box plot

3. Funkcionalnosti aplikacije

Glavna ideja ove aplikacije je omogućiti korisnicima na jednostavan i pristupačan način interaktivno istraživanje skupa podataka o putovanjima. Kroz korištenje biblioteke Streamlit, te ostalih biblioteka programskog jezika Python, razvijeno je više vrsta grafičkih prikaza podataka. Prilikom izrade ove aplikacije, primarni cilj je bio da korisniku vizualni prikazi podataka budu razumni, kako bi svi oni željni istraživanja tog područja mogli jednostavno pristupiti takvoj vrsti podataka. Vrlo često kada dobiju "surove" podatke u tablicama, poput *Excel* tablice ili pak kako je i u ovom radu korišteno, CSV datoteke, korisnici odustanu od bilo kakve analize i proučavanja tih podataka.

Kroz ovu aplikaciju je korisniku omogućen pregled podataka, prikazanih interaktivnim grafovima kako bi se svaki podatak što jednostavnije mogao analizirati. Osim toga, prikazane su i neke veze između određenih varijabli, čime se pruža još više podataka koji mogu pomoći u donošenju vlastitih odluka i prosudbi. Vrlo važna činjenica je što je omogućeno i filtriranje podataka po osobnim željama, što je još samo jedna od prednosti zbog čega je ovaj program iznimno jednostavan i produktivan za korištenje svakom korisniku.

Aplikaciju je moguće koristiti kroz 2 uloge, a to su:

- Administrator
- Korisnik

Sučelje aplikacije i funkcionalnosti su u potpunosti prilagođene tim korisnicima.

3.1. Kreiranje korisnika

Iako je primarni cilj aplikacije prikaz grafičkih podataka koji je svim omogućen, stvorena je registracijska forma koja korisniku dopušta da registraciju kako bi imao pristup svim interakcijskim grafovima. Osim toga, za korisnika s administratorskom ulogom stvoreno je potpuno novo sučelje kako bi mogao dodavati nove podatke, ažurirati podatke koji se trenutno nalaze u bazi podataka, te ujedno i brisati postojeće podatke.

Forme se otvaraju na temelju korisničkog imena, a u slučaju unaprjeđenja aplikacije u budućnosti, bilo bi moguće dodati uvjete koji bi prikazivali forme s obzirom na ulogu prijavljenog korisnika.

Kako bi podatke za prijavu o korisnicima imali dostupnima za provjeru, prvenstveno je potrebno stvoriti vezu s bazom podataka kako bi se u njoj moglo kreirati tablicu koja će sadržavati podatke za prijavu. Za ovaj projekt stvorena je tablica korisnici (engl. *users)* koja u sebi sadrži pet stupaca koji sadrže podatke o identifikacijskom broju (engl. *ID*), elektroničkoj pošti (engl. *email*), korisničkom imenu (engl. *username*), lozinki (engl. *password*) i datumu registracije (engl. *data joined*). Kako bi se tablica uspješno stvorila u bazi, odnosno kako bi slanje podataka bilo uspješno, potrebno je kroz funkciju poslati podatke. Iznimno bitna stavka je kriptirati lozinku kako bi podaci za prijavu bili sigurni za korištenje.

Forma za registraciju provjerava je li adresa elektroničke pošte netočna i upisuje li se već postojeća, provjerava valjanost unesene lozinke i korisničkog imena, te ako su svi podaci ispunili uvjete stvara se novi korisnik čiji se podaci pohranjuju u bazu. Izgled registracijske forme vidljiv je na slici 5.

Slika 5: Registracijska forma

3.2. Kriptiranje lozinke

Enkripcijski mehanizam pretvara lozinku u niz različitih znakova koji ne predstavljaju smislenu riječ, a koju nije moguće dekriptirati bez privatnog ključa. Sigurnost

kriptirane lozinke ovisi o sigurnosti ključeva pohranjenih na poslužiteljima kompanija čije se internetske stranice koriste.

Danas se najčešće koriste metode „*salt*“ i „*hash*“, a u ovom projektu je korišten „*salt*“ način kriptiranja lozinke [8]. To je jednosmjerni algoritam kriptiranja lozinke, koji lozinku pretvara u niz znakova koji su nejasni i neprepoznatljivi za čitanje. U izradi projekta koristila se *'bcrypt'* biblioteka koja služi za skladištenje lozinki. Naredba kojom se instalira biblioteka nalazi se u ispisu 5.

```
$ pip install bcrypt
```

Ispis 5: Naredba za instalaciju biblioteke 'bcrypt'

Prvenstveno se originalna lozinka kriptira u bajtove koristeći UTF-8 kodiranje. Ovaj korak je potreban jer biblioteka očekuje bajtove kao ulazne podatke. Zatim *„salt“* ustvari predstavlja nasumične znakove koji će se koristiti zajedno s lozinkom kako bi se stvorila sigurna kriptirana lozinka, odnosno nasumičan niz bajtova. U nastavku kôda se uzima kriptirana lozinka koja se generira, te je jedinstvena za svako stvaranje lozinke, a kao posljednji korak lozinka se dekodira iz bajtova u UTF-8 string. Posljednji korak služi kako bi se kriptiranu lozinku moglo lakše skladištiti u bazi podataka. Naredba za kriptiranje lozinki nalazi se u ispisu 6.

```
hashed_password = bcrypt.hashpw(password.encode('utf-8'),
bcrypt.gensalt()).decode('utf-8')
```

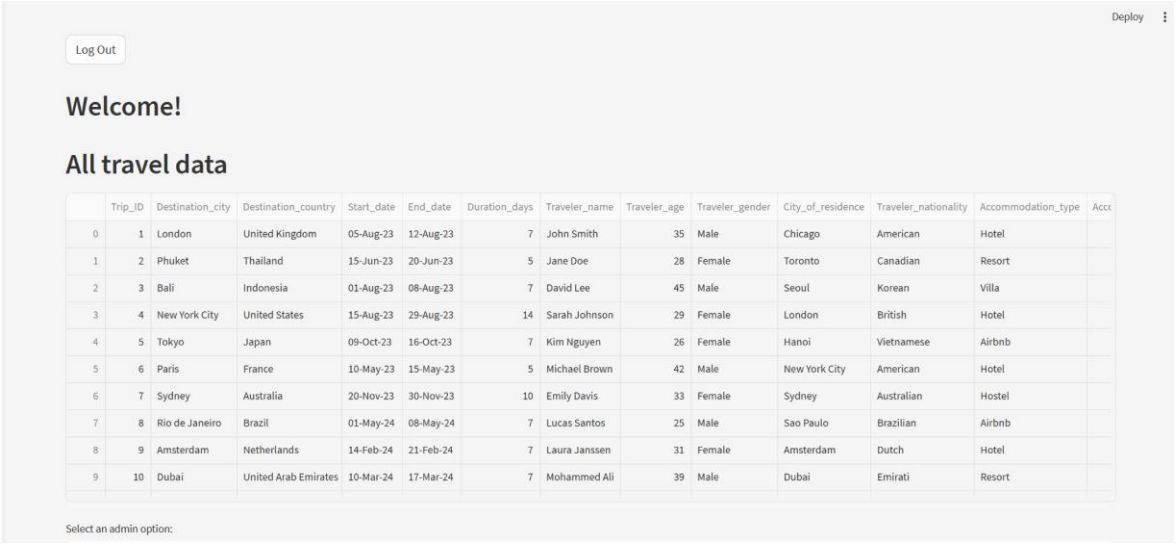
Ispis 6: Naredba za kriptiranje lozinki

Prilikom prijave korisnika, lozinka se opet kriptira te se uspoređuje s kriptiranom lozinkom u bazi podataka. Ako se lozinke podudaraju prijava je uspješna.

4. Način rada aplikacije

4.1. Administrativno sučelje

Nakon uspješne prijave, otvara se sučelje koje je osmišljeno specifično za administratora. Tu se nalaze funkcionalnosti koje administratoru pružaju uvid u tablicu svih podataka koji se trenutno nalaze u bazi podataka, dodavanje novih podataka, ažuriranje postojećih podataka ili pak njihovo brisanje iz baze podataka. Cilj je bio za korisnika s ulogom administrator ispuniti CRUD uvjet (engl. *create, read, update, delete*). Na vrhu stranice se nalazi dugme za odjavu (engl. *log out*) korisnika. Početni zaslon korisnika prijavljenog kao administrator prikazuje se na slici 6.



	Trip_ID	Destination_city	Destination_country	Start_date	End_date	Duration_days	Traveler_name	Traveler_age	Traveler_gender	City_of_residence	Traveler_nationality	Accommodation_type	Acc
0	1	London	United Kingdom	05-Aug-23	12-Aug-23	7	John Smith	35	Male	Chicago	American	Hotel	
1	2	Phuket	Thailand	15-Jun-23	20-Jun-23	5	Jane Doe	28	Female	Toronto	Canadian	Resort	
2	3	Bali	Indonesia	01-Aug-23	08-Aug-23	7	David Lee	45	Male	Seoul	Korean	Villa	
3	4	New York City	United States	15-Aug-23	29-Aug-23	14	Sarah Johnson	29	Female	London	British	Hotel	
4	5	Tokyo	Japan	09-Oct-23	16-Oct-23	7	Kim Nguyen	26	Female	Hanoi	Vietnamese	Airbnb	
5	6	Paris	France	10-May-23	15-May-23	5	Michael Brown	42	Male	New York City	American	Hotel	
6	7	Sydney	Australia	20-Nov-23	30-Nov-23	10	Emily Davis	33	Female	Sydney	Australian	Hostel	
7	8	Rio de Janeiro	Brazil	01-May-24	08-May-24	7	Lucas Santos	25	Male	Sao Paulo	Brazilian	Airbnb	
8	9	Amsterdam	Netherlands	14-Feb-24	21-Feb-24	7	Laura Janssen	31	Female	Amsterdam	Dutch	Hotel	
9	10	Dubai	United Arab Emirates	10-Mar-24	17-Mar-24	7	Mohammed Ali	39	Male	Dubai	Emirati	Resort	

Slika 6: Početni zaslon za administratora

Sve mogućnosti se prvenstveno nalaze u padajućem izborniku (engl. *dropdown*). Prilikom odabira željene funkcije, otvara se forma ovisno o odabiru.

4.1.1. Dodavanje podataka

Dakle, ako korisnik u padajućem izborniku odabere Dodaj podatke (engl. *Add Data*), u nastavku sučelja otvara se forma s poljima za unos podataka. Sva polja već postoje u bazi podataka te su s njom ujedno i povezana kako bi prilikom unosa novih bili pravilno i uspješno spremljeni te prikazani u bazi podataka. Identifikacijski broj (engl. *identification-*

ID) svih podataka se u bazi nalazi pod nazivom Trip_ID, koji je kao primarni ključ jedinstven, te je namješten kao automatski inkrement (engl. *autoincrement*). Zbog ove stavke administrator prilikom unosa ne unosi podatke za stupac Trip_ID, već se on sam dodjeljuje automatski. Ostale varijable korisnik unosi samostalno, no i tu postoje neka ograničenja. Ispis 7 prikazuje dio koda kojim se postavljaju ograničenja prilikom unošenja podataka.

```
if column != "Trip_ID":

      if column == "Start_date" or column == "End_date":

            new_date = st.date_input(column)

            new_data[column] = new_date.strftime("%d-%b-%y") if new_date
else None

      elif column == "Traveler_age":

           new_value = st.number_input("Traveler Age", key=column,
step=1, min_value=18)

           new_data[column] = int(new_value) if new_value is not None
else None

      elif column == "Traveler_gender":

         gender_options = ["Male", "Female"]

         selected_gender = st.selectbox("Gender", gender_options)

         new_data[column] = selected_gender

      elif column == "Accommodation_type":

         accommodation_options = ["Hotel", "Resort", "Villa", "Airbnb",
"Hostel", "Riad", "Vacation rental", "Guest house"]

         selected_accommodation = st.selectbox("Accommodation Type",
accommodation_options)

         new_data[column] = selected_accommodation

      elif column == "Accommodation_cost":

         new_value = st.number_input("Accommodation cost", key=column,
step=1)

         new_data[column] = int(new_value) if new_value is not None else
None
```

Ispis 7: Dio kôda za prikaz unosnih polja

Polja za odabir datuma početka putovanja (engl. *start date*) te datuma završetka (engl. *end date*) su prikazana preko grafičkog sučelja za unos datuma. Nadalje, polja za odabir vrste smještaja i vrste prijevoza su padajući izbornici (engl. *dropdown*), te su ograničeni samo na one vrste koje se već nalazi u bazi podataka. Osim toga, trošak prijevoza i smještaja je ograničen na unos jedino brojki (engl. *number input*), isto kao i unos dobi putnika. Zbog toga se na stranici u tim poljima prikazuje brojka 0 prije unosa bilo kakvih podatka.

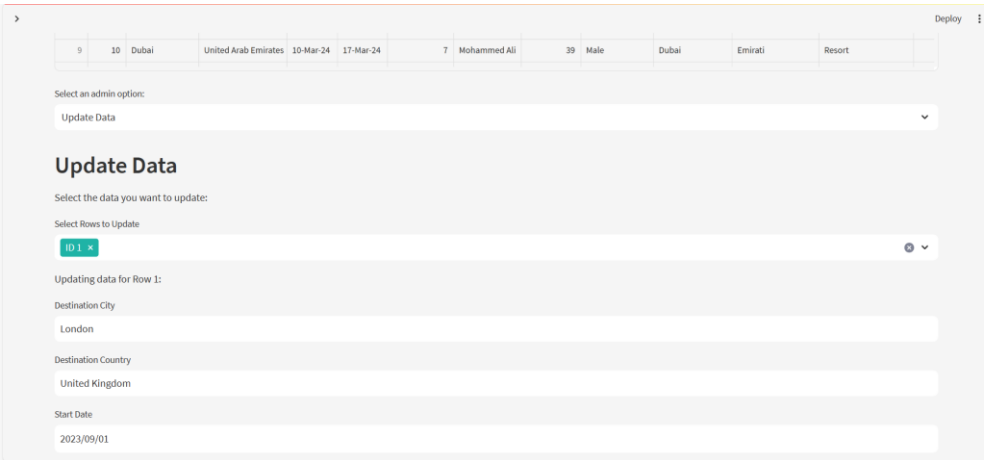
Polja za unos grada odredišta (engl. *destination city*), države odredišta (engl. *destination country*) te imena putnika nemaju nikakvih posebnih ograničenja, osim što su postavljena za tekstualni unos.

Kada su svi željeni podaci uneseni, na dnu forme se nalazi dugme „Add new data“ kojim se potvrđuje dodavanje unesenih podataka u bazu.

4.1.2. Ažuriranje podataka

Administrator ima i mogućnost ažuriranja podataka koji se već nalaze u bazi podataka. Kada korisnik u padajućem izborniku odabere ažuriranje podataka (engl. *Update Data*) otvara se dodatni okvir za odabir u kojem korisnik odabire podatke s kojim ID brojem želi ažurirati.

Kako bi se otvorila forma s poljima za ažuriranje podataka, potrebno je odabrati podatke s kojim ID brojem želimo ažurirati, a odabire se u novom padajućem izborniku koji se otvori prilikom odabira na ažuriranje podataka. Na isti princip se i u ovoj formi potvrđuje unos podataka, a to je pritiskom na dugme „Update Selected Row“. Izgled forme za ažuriranje podataka prikazan je na slici 7.



Slika 7: Forma za ažuriranje podataka

4.1.3. Brisanje podataka

Posljednja funkcionalnost koja se nudi administratoru je brisanje podataka iz baze. Ima mnogo sličnosti s prethodnom funkcijom za ažuriranje podataka. Administrator kroz padajući izbornik po ID broju podatka odabire koji redak želi izbrisati. Kao posljednja akcija koju je potrebno izvršiti kako bi se podaci zaista obrisali, potrebno je kliknuti dugme „Delete Selected Rows“, nakon čega se u slučaju uspješnog brisanja podatka prikazuje poruka „Selected rows have been deleted“. Funkcija za brisanje podataka prikazana je u ispisu 8.

```
rows_to_delete = st.multiselect(
      "Select Rows to Delete",
      options=[f"Row {data[0]}" for data in all_data],
)
if st.button("Delete Selected Rows"):
      cursor = connection.cursor()
      for row in rows_to_delete:
          row_id = int(row.split()[1])
          cursor.execute("DELETE FROM travel_details WHERE Trip_ID=?",
(row_id,))
      connection.commit()
      st.success("Selected rows have been deleted.")
```

Ispis 8: Funkcija za brisanje podataka

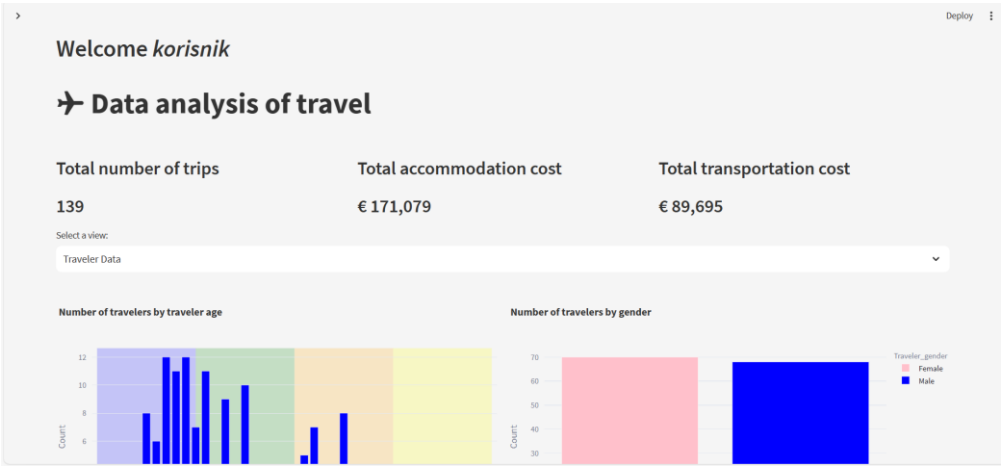
4.2. Korisničko sučelje

Za razliku od sučelja kod administrativne uloge, sučelje za korisnika sadrži prikaz svih grafova koji su osmišljeni za ovaj projekt. Nakon prijave korisnika, otvara se prikaz koji mu pruža uvid u dio grafova. Na isti način kako administrator kroz padajući izbornik odabire za koju funkciju želi otvoriti formu, tako i korisnik odabire za koje područje želi otvoriti prikaz grafova.

Grafovi su podijeljeni u tri područja, a to su:

- Podaci o putniku (engl. *Traveler Data*)
- Podaci o smještaju (engl. *Accommodation Data*)
- Podaci o prijevozu (engl. *Transportation Data*)

Podjela na ovaj način omogućuje korisniku lakši protok kroz vrste podataka, kako bi se što jednostavnije mogao fokusirati na određeni dio. Na primjer, korisnik koji želi prvenstveno imati bolji uvid u vrste smještaja i cijene, u padajućem izborniku ima mogućnost odabrati podatke o smještaju (engl. *Accommodation Data*) kako bi imao grafički prikazane sve poznate varijable o tim stavkama, ujedno povezane s ostalim ključnim parametrima. Kroz grafičke prikaze korisniku se na jednostavan i pristupačan način prikazuje poveznica između cijena različitih vrsta smještaja ovisno o lokaciji putovanja. Slika 8 prikazuje početni zaslon korisničkog sučelja.



Slika 8: Početni zaslon korisničkog sučelja

Na početnom zaslonu se otvara i bočna traka (engl. *sidebar*) koja korisniku pruža interaktivno iskustvo koje omogućuje filtriranje podataka. Za svaku formu se otvara drugačiji skup podataka u bočnoj traci. Osim toga, kroz sve stranice se pruža prikaz informacija o ukupnom broju podataka o kojima se provodi analiza, kolika je ukupna potrošnja na smještaj te ukupna potrošnja na transport (izraženo u valuti euro - €). Kod korisnika se dugme za funkciju odjave nalazi u bočnom izborniku.

4.2.1. Grafički prikazi podataka o putniku

Kada korisnik otvori grafove za podatke o putniku, u bočnoj traci je moguće mijenjati željene podatke za dob putnika, spol putnika ili odredišni grad. Kroz ovu funkcionalnost korisnici imaju direktan utjecaj na grafove koji prikazuju relevantne informacije. Konkretno, filter za odabir dobi putnika se prikazuje kao klizač (engl. *slider*). S obzirom na trenutne podatke u bazi, raspon dobi se prikazuje od najmanje unesenog broja za taj podatak, do najvećeg unesenog broja. Kako se podaci u bazi mijenjaju tako se i mijenja raspon. Funkcija za filtriranje dobi putnika prikazana je u ispisu 9.

```
age_range = st.sidebar.slider(
      "Select age range:",
      min_value=int(min(unique_ages)),
      max_value=int(max(unique_ages)),
      value=(int(min(unique_ages)), int(max(unique_ages))),
      step=1
)
filtered_df = df[
      (df["Traveler_age"] >= age_range[0]) &
      (df["Traveler_age"] <= age_range[1])
]
```

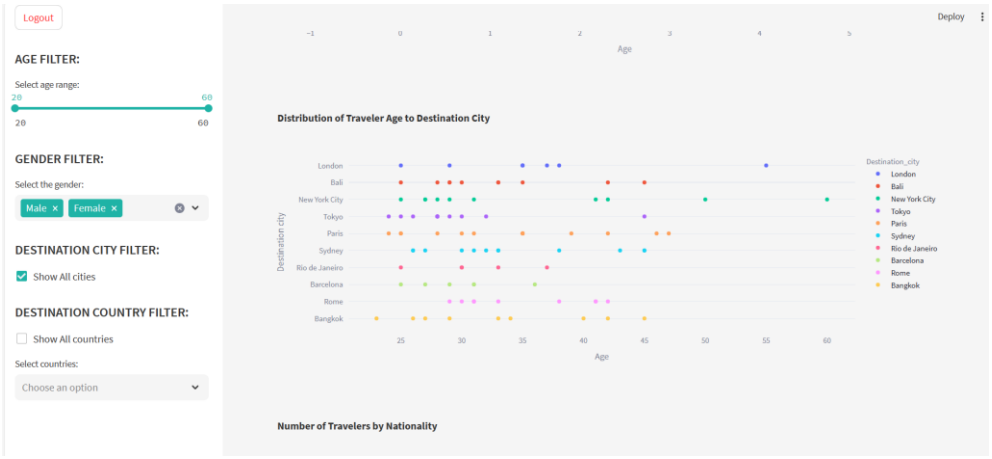
Ispis 9: Funkcija za filtriranje dobi putnika

Mogućnost odabira spola postavljena je na dvije podijele, muški (engl. *male*) i ženski (engl. *female*), pa je tako i odabir omogućen tako da se mogu prikazati podaci za oba spola, samo jednog od ponuđenih ili pak isključiti te podatke iz odabira.

S obzirom na veliku količinu različitih podataka o odredišnim gradovima, mogućnost filtriranja je prikazana na što jednostavniji način. Prilikom samog pokretanja aplikacije, prikazuju se podaci za sve gradovi koji se nalaze u bazi, a to je postignuto tako što je stavka za odabir svih gradova (engl. *all cities*) postavljena na označenu vrijednost (engl. *value True*). S obzirom na to da je nekolicina grafičkih prikaza povezana s nazivima odredišnih gradova, korisniku funkcionalnost potvrdnog okvira da odabere sve gradove uvelike olakšava korištenje filtera.

Nadalje, u slučaju kada korisnik ne želi koristiti odabir prikaza svih odredišnih gradova, već samostalno odabrati određeni grad (jedan ili više njih), potrebno je poništiti odabir potvrdnog okvira, nakon čega se u bočnoj traci otvara višestruki odabir s popisom svih gradova.

Na potpuno isti način funkcionira i filtriranje odredišnih država. Slika 9 prikazuje bočnu traku s navedenim filterima te graf distribucije dobi putnika u odnosu na odredišni grad.

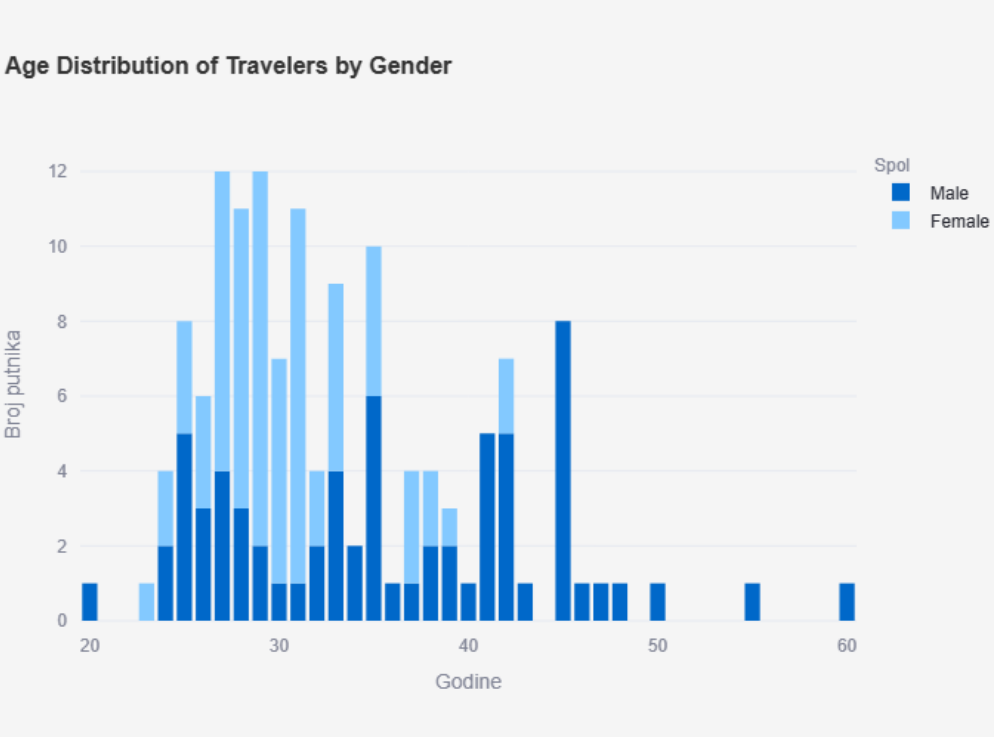


Slika 9: Bočna traka i graf distribucije

Grafikon dobi putnika dat će prikaz količine putnika ovisno o njihovoj dobi. S obzirom na to da je ovaj grafički prikaz povezan s filterom za dob putnika, vrijednost grafa se može mijenjati ovisno o vrijednosti koju korisnik postavi. Kako bi korisniku bilo jednostavno procijeniti razliku podataka ovisno o dobi, pozadina grafa je prikazana različitim bojama po rasponu dobnih skupina u intervalima po 10 godina.

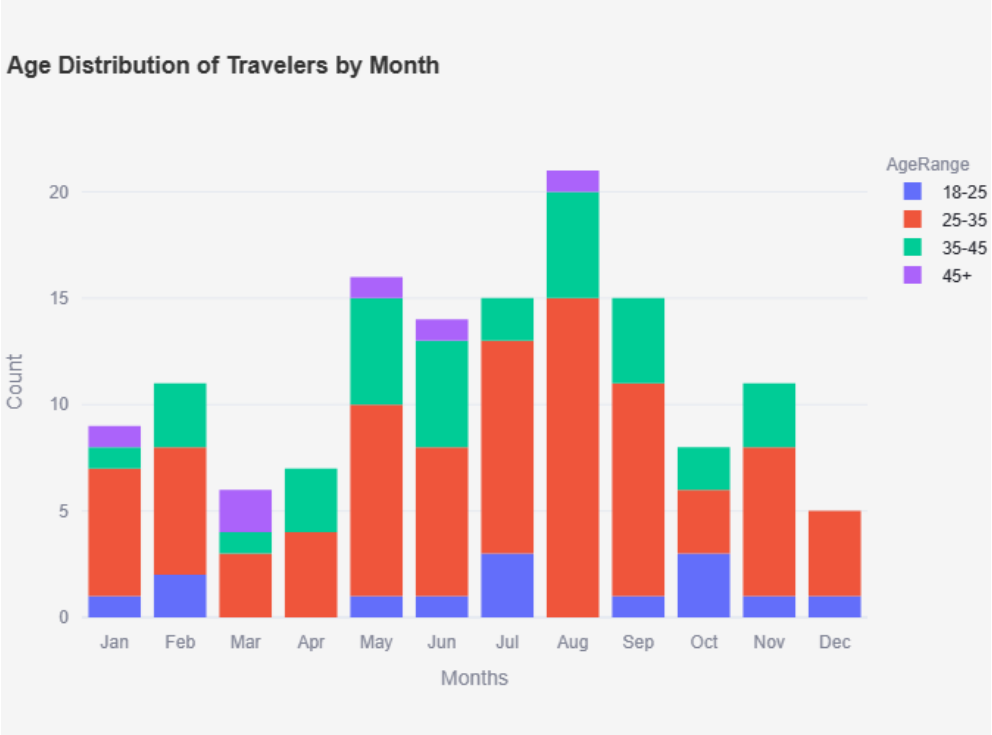
Grafički prikaz spola putnika će prikazati omjer muških i ženskih putnika te potencijalne razlike ovisno o spolu. Graf je prikazan različitim bojama za svaki spol, pa roza boja označava podatke s odabirom spola „*female*“, a plava boja označava odabir spola „*male*“. Legenda se nalazi u desnom gornjem uglu grafičkog prikaza kako bi korisnik točno znao što koja boja označava.

S obzirom na to da prethodna dva grafička prikaza prikazuju odvojene podatke o spolu i dobi svakog putnika, sljedeći grafikon prikazuje distribuciju putnika s obzirom na dob i spol. Kao i u prethodnom primjeru, u desnom gornjem uglu se nalazi legenda koja prikazuje koja boja označava koji spol, te korisniku omogućuje točan prikaz broja putnika određenog spola s određenom dobi. Zbog toga što za istu dob se može imati podatke kako za ženski spol tako i za muški spol putnika, u istom stupcu se mogu prikazivati podaci za obje opcije. Graf je prikazan na slici 10.



Slika 10: Prikaz distribucije dobi putnika u odnosu na spol putnika

Osim što je pružen grafički prikaz podataka ovisnosti broja putnika o dobi i spolu, ujedno je prikazana i ovisnost broja putnika o dobi i mjesecima u godini kada se putuje. Kako je vidljivo u grafu, s obzirom na trenutne podatke koji se nalaze u bazi, najveći broj putnika je tijekom kolovoza (engl. *August*), a dobna skupina 25-35 koja prikazuje putnike u tom dobnom rasponu ima najveću količinu podataka tijekom cijele godine. Slika 11 prikazuje opisano.



Slika 11: Distribucija dobi putnika u odnosu na mjesec u godini

U nastavku korisnik ima uvid u veći broj grafikona koji su povezani s gradovima i državama, kako s onim odakle korisnik dolazi tako i s podacima o gradovima i državama kamo se putuje.

Sljedeća tri grafička prikaza izražena su korištenjem funkcionalnosti biblioteke Plotly Express za dijagram raspršenosti (engl. *scatter*). Za sva tri grafa je korišten prikaz samo 10 najviše ponavljanih zapisa.

Prvi grafikon prikazuje distribuciju putnika ovisno o njegovoj dobi, a podatak o njegovoj nacionalnosti je označen različitim bojama i znakovima za svaku nacionalnost. Svaki znak prikazan na grafikonu označuje jednog putnika. Što je više znakova na liniji određene nacionalnosti, to označava veći broj putnika te nacionalnosti. Ako korisnik želi točno znati dob tog putnika, pokazivačem na ekranu može prijeći preko znaka tog podatka, te će mu se prikazati dodatni prozor u kojem je osim napisane nacionalnosti prikazana i dob tog određenog putnika.

Ovaj grafički prikaz je povezan s filterom za dob putnika u bočnom izborniku, te je moguće korištenjem filtera mijenjati prikaz grafa ovisno o tom faktoru. S desne strane se nalazi legenda koja prikazuje popis nacionalnosti s prikazom znakova koji se koriste.

Sljedeći dijagram raspršenosti na isti način prikazuje broj putnika ovisno o dobi, ali povezan s odredišnom državom. S obzirom na to da korisnik ima mogućnost korištenjem filtera mijenjati željene odredišne države, ovaj je grafički prikaz moguće mijenjati koristeći dva filtera: filter o dobi putnika i filter o odredišnoj državi. S desne strane se nalazi legenda koja prikazuje popis odredišnih država s prikazom znakova koji se koriste za oznaku svakog putnika koji ispunjava taj uvjet.

Na isti način prikazan je i sljedeći grafikon koji pruža podatke o distribuciji putnika u ovisnosti o dobi i odredišnom gradu. Kako korisnik ima mogućnost filtriranja podataka o odredišnim državama, isto tako ima mogućnost i filtriranja podataka o odredišnim gradovima. S desne strane se nalazi legenda koja prikazuje popis odredišnih gradova s prikazom znakova koji se koriste.

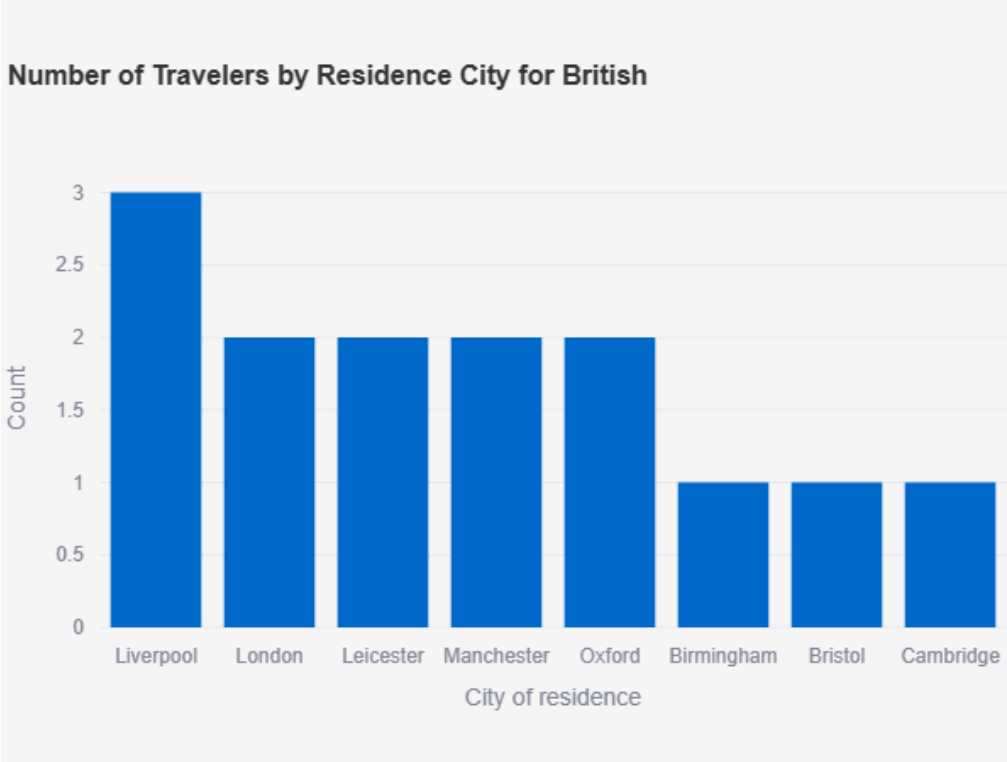
Sve legende imaju mogućnost isključivanja prikaza podataka iz grafa i ponovnog vraćanja u prikaz. Korisnik klikom na određenu informaciju u legendi uklanja znakove koji ispunjavaju taj podatak, a ponovnim klikom se vraća prikaz znakova.

Dijagram raspršenosti s podacima o odredišnim državama se nalazi na slici 12.

Slika 12: Dijagram raspršenosti

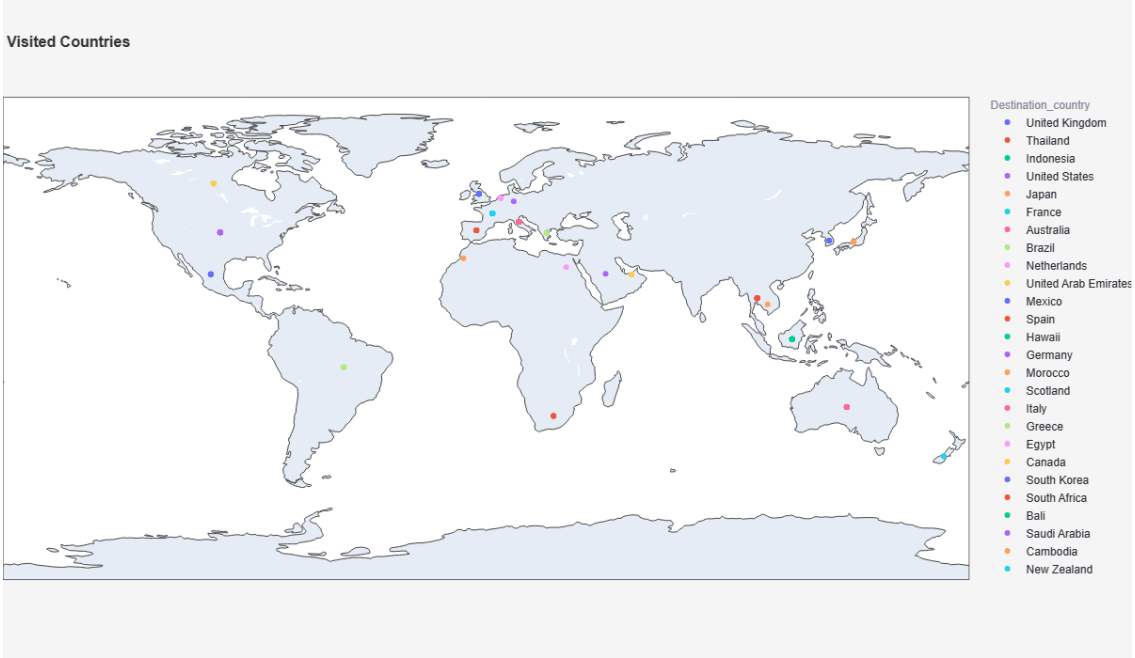
Sljedeći grafikon pruža prikaz broja putnika ovisno o nacionalnosti, te se u nastavku na taj grafikon prikazuje okvir za odabir s listom svih nacionalnosti koje se nalaze u bazi, a

odabirom na određenu nacionalnost novi grafikon će prikazati sve gradove stanovanja koji su povezani s tom nacionalnošću te zbroj putnika koji dolaze iz svakog od tih gradova. Graf koji prikazuje broj putnika ovisan o nacionalnosti nalazi se na slici 13.



Slika 13: Broj putnika u odnosnu na nacionalnost

Za posljednji grafički prikaz na ovoj stranici koristio se geografski prikaz raspršenosti. Na stranici je moguće vidjeti prikaz karte svijeta, a na njoj su točkama različitih boja označene države odredišta. Osim što legenda s desne strane opisuje koja država odredišta je označena kojom bojom na karti, ako korisnik pokazivačem na ekranu se zaustavi na točki označenoj na karti, na ekranu će mu se prikazati o kojoj državi se radi. Navedeni grafički prikaz nalazi se na slici 14.

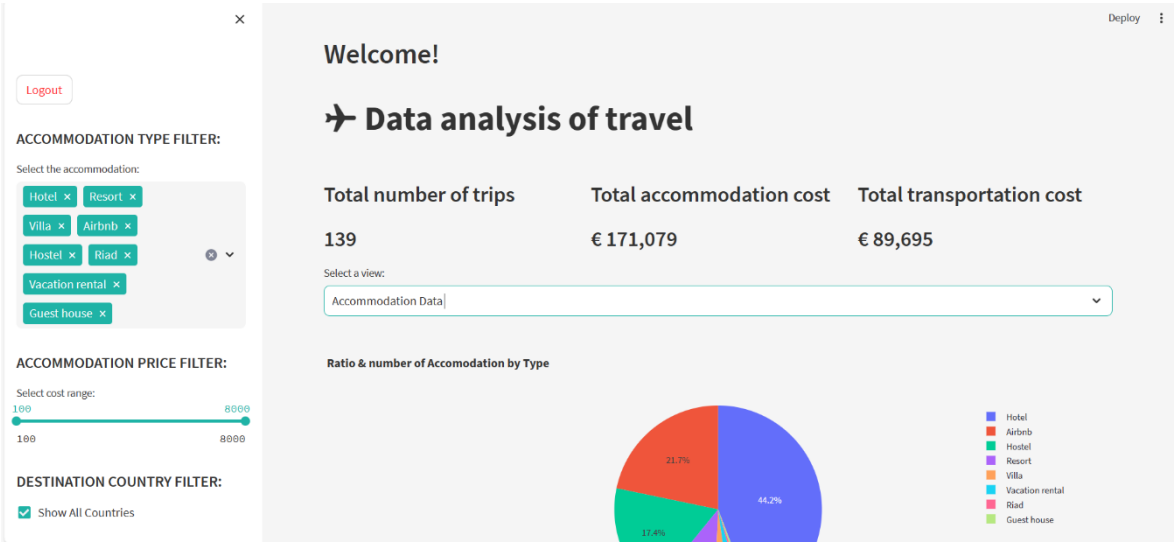


Slika 14: Geografski prikaz država odredišta

Prilikom prikaza grafova za podatke o smještaju korisniku se pruža mogućnost istraživanja različitih aspekata smještaja, dok istovremeno otvara mogućnost za filtriranje dvije vrste podataka: vrsta smještaja i cijena smještaja. Korisnik se može fokusirati na prikaz rezultata prema svojim preferencijama, istražujući određenu vrstu smještaja ili pak ovisno o rasponu cijena koje odgovaraju njegovom financijskom okviru.

U bočnoj traci se i dalje nalazi dugme za odjavu korisniku, te se ta funkcionalnost kroz cijelo korisničko sučelje nalazi kao početna funkcionalnost u bočnoj traci. Osim toga, korisnik ima mogućnost filtriranja podataka ovisno o vrsti smještaja. S obzirom na to da se u bazi ne nalazi previše različitih podataka o vrsti smještaja, korišten je višestruki odabir kao način prikaza podataka kako bi ih se moglo filtrirati.

Pri pokretanju prikaza podataka o smještaju sve vrste smještaja su označene u filtriranju. U slučaju kada korisnik želi neku vrstu smještaja ukloniti iz grafičkih prikaza, kraj naziva svake stavke nalazi se oznaka „x“ kako bi se poništio odabir tog smještaja. Ako korisnik u budućnosti ponovno želi uključiti tu stavku, stvara mu se padajući izbornik s popisom svih vrsta smještaja koje nisu uključene u trenutne filtere. Izgleda stranice za prikaz podataka o smještaju nalazi se na slici 15.



Slika 15: Podaci o smještaju

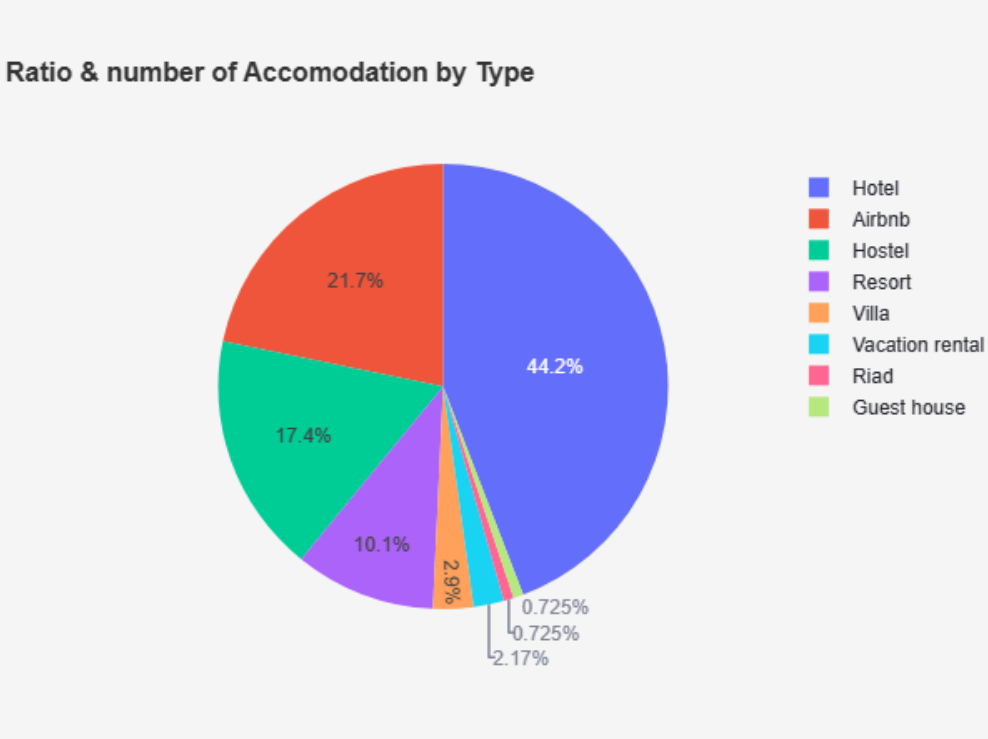
Osim toga, pružena se i mogućnost odabir raspona cijena smještaja. Korišten je klizač (engl. *slider*) kako bi korisnik mogao na što jednostavniji način imati mogućnost odabira cijena smještaja za koje želi imati uvid u grafove. Klizač radi na princip da se uvijek izračuna najmanja trenutna cijena smještaja i najveća trenutna cijena smještaja koja se nalazi u bazi, a korisniku se pruža filtriranje podataka u tom rasponu cijena. Primjer kôda za izradu filtera nalazi se u ispisu 10.

```
cost_range = st.sidebar.slider("Select cost range:",
      min_value=int(min(unique_cost)),
      max_value=int(max(unique_cost)),
      value=(int(min(unique_cost)), int(max(unique_cost))),
      step=1
)
```

Ispis 10: Kôd za filtriranje podataka o rasponu cijena smještaja

Kao prvi grafički prikaz koristeći kružni graf (engl. *pie chart*) prikazan je omjer broja smještajnih objekata prema vrsti smještaja. U ovom slučaju, ovakav tip grafa koristi se za prikazivanje udjela svake vrste smještaja u ukupnom broju smještajnih objekata. Svaki udio

prikazan na x-osi predstavlja različitu vrstu smještaja, koji je površinom proporcionalan broju smještajnih objekata koji pripadaju toj vrsti smještaja. Kako bi čitanje podataka iz grafa bilo što jednostavnije, svaki udio je bojan drugačijom bojom. Kako bi korisnik znao koja boja označava koju vrstu smještaja, s desne strane se nalazi legenda koja to ukratko i opisuje. U slučaju da korisnik želi znati točan broj ponavljanja određene vrste smještaja u bazi, prelaskom pokazivačem na ekranu preko svakog udjela, prikaže mu se naziv vrste smještaja i broj koliko puta se ta vrsta smještaja pojavljuje u bazi podataka. Prikaz kružnog grafa nalazi se na slici 16.



Slika 16: Omjer vrsta smještaja

Kako bi korisnik imao uvid u cijene smještaja ovisno o vrsti smještaja, što je očigledno glavna varijabilna vrijednost koja utječe na raspon cijena, korišten je grafički prikaz skupine statističkih podataka, poznat i pod nazivom histogram.

Informacije su kroz kôd analizirane tako da se prikažu najčešće cijene smještaja. Da bi se to postiglo, koristio se skup podataka koji sadrži informacije o broju ponavljanja određene cijene smještaja. Cijene smještaja koje su se pojavile više od jednom u skupu podataka bile su uzete u obzir.

Kada su određene najčešće cijene smještaja, podaci se dalje filtriraju na temelju tih cijena. Provodi se sortiranje podataka po cijenama smještaja u uzlaznom redoslijedu. Ovaj korak omogućuje bolje razumijevanje distribucije cijena smještaja za svaku vrstu smještaja. U grafičkom prikazu se koriste različite boje tako da je svaka vrste smještaja označena drugačijom bojom.

Ovaj grafički prikaz je povezan s filterima za odabir vrste smještaja i raspona cijene smještaja za koje se želi imati uvid u podatke, te je prikazan na slici 17.

Slika 17: Cijene smještaja za vrste smještaja

Kako bi se korisniku pružila što jednostavnija predodžba kako se raspon cijena zaista kreće za svaku vrstu smještaja zasebno, izrađen je graf koji to pobliže najbolje opisuje.

S obzirom na to da se željela postići točna analiza s jednakim uvjetima za svaku vrstu smještaja, podaci su ograničeni samo na one čije je trajanje boravka iznosilo točno 7 dana. Svaki *box plot* predstavlja interkvartilni raspon (engl. *interquartile- IQR*), odnosno srednji 50% podataka. Donja granica označava 25. percentil, a gornja granica označava 75. percentil podataka. Također sadrži i crtu koja predstavlja srednju vrijednost (engl. *mediana*) cijena smještaja za svaku vrstu. Linije (engl. *whisker*) koje se protežu izvan rubova kutije prema

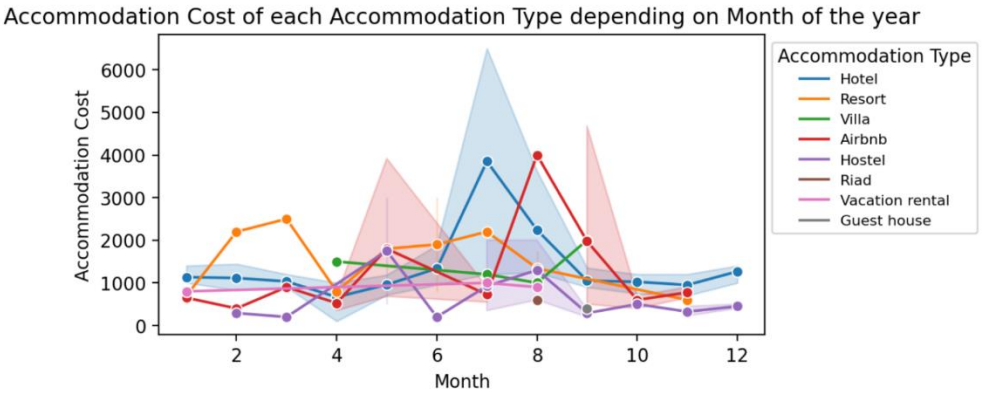
najmanjoj i najvećoj vrijednosti unutar određenog raspona smatraju se ekstremima (engl. *outliers*). Te vrijednosti, odnosno točke izvan rubova koje se smatraju ekstremima predstavljaju neobično visoke ili niske cijene smještaja koje se izdvajaju od ostalih podataka.

Analiza ovog grafa može svakom korisniku omogućiti lakše razumijevanje raspona cijena za određene vrste smještaja koje mogu očekivati. Prikaz kôda korištenog za izradu ovog grafa nalazi se u ispisu 11.

```
df_duration_filtered = df[df['Duration_days'] == 7]

fig_box_accomm = px.box(

      df_duration_filtered,

      x='Accommodation_type',

      y='Accommodation_cost',

      title='Box Plot shown Accommodation Price range for each Accommodation Type (7 days stays only)',

      labels={'Accommodation_cost': 'Accommodation cost'}

)
```

Ispis 11: Kôd za izradu box plot grafa

Cijene smještaja se uvelike razlikuju zavisno u koje doba godine se planira putovati. Sljedeći graf je fokusiran na analizu promjene cijene smještaja tijekom godine. Cilj je razumjeti sezonske promjene u cijenama smještaja te na neki način prikazati trendove, odnosno varijacije koje se mogu pojaviti tijekom različitih mjeseci. Izgled grafikona koji to prikazuje prikazan je na slici 18.

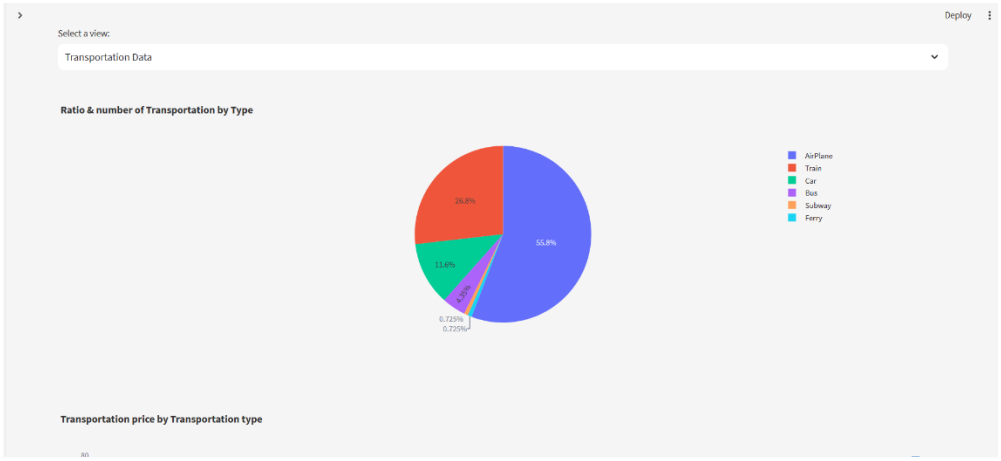


Slika 18: Prikaz cijena za vrste smještaja ovisno o mjesecu u godini

Kroz posljednji grafički prikaz na stranici koja prikazuje podatke povezane s informacijama o smještajima, nalazi se dijagram raspršenosti koji daje uvid u cijene različitih vrsta smještaja u različitim gradovima odredišta.

Svaka točka na grafu predstavlja jednu vrstu smještaja. Boja točaka određuje vrstu smještaja, a legenda s desne strane grafa prikazuje koja boja se koristi za prikaz koje vrste smještaja. Ovakva vrsta podataka može biti korisna za sve koji žele razumjeti gdje mogu pronaći povoljne smještaje u određenim gradovima.

Posljednja podjela po kojoj korisnik može proučavati podatke i prikazuju se odvojeni grafovi odnosi se na podatke vezane uz transport. Pošto podaci o transportu imaju jednake podatke kao i podaci o smještaju, a to su vrsta i cijena transporta, napravljeni su isti grafovi, koristeći iste funkcionalnosti, jedina je razlika što su se vrijednosti uzimale iz podataka o vrsti transporta i cijeni transporta. Prikaz podjele vidljiv je na slici 19.



Slika 19: Podaci o transportu

5. Zaključak

Aplikacija je osmišljena tako da korisniku omogući lakši proces razumijevanja različitih stavki putovanja, poput cijena, vrsta smještaja ili pak varijacija cijena ovisno o razdoblju putovanja. Kroz različite grafičke prikaze korisniku je velika količina podataka predstavljena na jednostavan, lako čitljiv način. Cilj je bio potaknuti korisnike na detaljniju analizu ponuđenih podataka, kako bi u svrhu planiranja putovanja imali što više potrebnih informacija koje bi mogle utjecati na njihove daljnje odluke.

Korištenjem programskog jezika Python te biblioteka koje nudi u svrhu izrade grafičkih prikaza, ponuđen je širok spektar različitih tipova grafičkih prikaza. Svaka vrsta najbolje odgovara određenim tipovima podataka, a kroz ovaj rad cilj je bio na što bolji način svaki pojedini graf prilagoditi određenom tipu podataka.

Streamlit biblioteka je za izradu ovog rada imala ključnu ulogu, s obzirom na to da nudi jednostavnost izrade vizualnog prikaza stranice, spajanja svih željenih grafičkih prikaza i željenih dodataka na vrlo jednostavan i interaktivan način. Osim toga, s već ugrađenom formom za provjeru autentikacije korisnika, proces stvaranje forme za prijavu i provjere korisnika je iznimno pojednostavljen.

Tijekom izrade cijelog rada, korištena je *SQLite* baza podataka koja je pružala mogućnost uvida u trenutno stanje svih korištenih podataka. Osim što je pružala mogućnost uvida i pregleda svih podataka, isto tako njena funkcionalnost spremanja podataka pruža mogućnost i dohvaćanja podataka iz tablica koje su stvorene u bazi.

Korištenjem svih navedenih alata zadovoljeni su postavljeni ciljevi napravljene aplikacije.

Literatura

[1] Službena stranica Python, https://www.python.org/about/ (posjećeno 01.09.2023.)

[2] Dokumentacija biblioteke Streamlit, https://docs.streamlit.io/ (posjećeno 01.09.2023.)

[3] Instalacija biblioteke Seaborn, https://seaborn.pydata.org/installing.html (posjećeno 25.08.2023.)

[4] Što je SQLite?, https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite (posjećeno 25.08.2023.)

[5] DB Browser for SQLite, https://sqlitebrowser.org/ (posjećeno 25.08.2023.)

[6] Visual Studio Code, https://visualstudio.microsoft.com/ (posjećeno 01.09.2023)

[7] Biblioteka Plotly, https://plotly.com/ (posjećeno 01.09.2023.)

[8] Kako kriptirati lozinku u Pythonu?, https://www.geeksforgeeks.org/how-to-hash-passwords-in-python/ (posjećeno 01.09.2023.)

Popis slika

Slika 1: Prikaz zaslona prilikom dodavanje putanje 7
Slika 2: Prikaz tablice o putovanjima 9
Slika 3: Prikaz histograma 13
Slika 4: Box plot..... 15
Slika 5: Registracijska forma 17
Slika 6: Početni zaslon za administratora 19
Slika 7: Forma za ažuriranje podataka 22
Slika 8: Početni zaslon korisničkog sučelja 23
Slika 9: Bočna traka i graf distribucije..... 25
Slika 10: Prikaz distribucije dobi putnika u odnosu na spol putnika 27
Slika 11: Distribucija dobi putnika u odnosu na mjesec u godini..... 28
Slika 12: Dijagram raspršenosti 29
Slika 13: Broj putnika u odnosnu na nacionalnost..... 30
Slika 14: Geografski prikaz država odredišta 31
Slika 15: Podaci o smještaju 32
Slika 16: Omjer vrsta smještaja 33
Slika 17: Cijene smještaja za vrste smještaja..... 34
Slika 18: Prikaz cijena za vrste smještaja ovisno o mjesecu u godini 36
Slika 19: Podaci o transportu 37