

FlutterFlow i Supabase tehnologije

Zrins, Marijan

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Applied Sciences in Information Technology / Veleučilište suvremenih informacijskih tehnologija**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:289:352896>

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

Download date / Datum preuzimanja: **2025-02-22**

Repository / Repozitorij:

[VSITE Repository - Repozitorij završnih i diplomskih radova VSITE-a](#)



VELEUČILIŠTE SUVREMENIH INFORMACIJSKIH TEHNOLOGIJA
STRUČNI PRIJEDIPLOMSKI STUDIJ INFORMACIJSKIH
TEHNOLOGIJA

Marijan Zrins

ZAVRŠNI RAD

FLUTTERFLOW I SUPABASE TEHNOLOGIJE

Zagreb, listopada 2024.



Veleučilište suvremenih informacijskih tehnologija
10000 Zagreb, Ulica Vjekoslava Klaića 7

Studij: Stručni prijediplomski studij informacijskih tehnologija
smjer programiranje
Student: **Marijan Zrins**
Matični broj: 2015013

Zadatak završnog rada

Predmet: Programiranje na Internetu
Naslov: **FlutterFlow i Supabase tehnologije**
Zadatak: Predstavljanje FlutterFlow i SupaBase tehnologija kroz primjer aplikacije za cjenike
Mentor: Mariza Maini, pred.
Zadatak uručen
kandidatu: 2.11.2023.
Rok za predaju rada: 24.10.2024.
Rad predan: _____

Povjerenstvo:

Jurica Đurić, v. pred.	član predsjednik	_____
Mariza Maini, pred.	mentor	_____
Dragana Čulina, pred.	član	_____

SADRŽAJ

1. UVOD	6
2. PREDSTAVLJANJE FLUTTERFLOW I SUPABASE TEHNOLOGIJA	8
2.1. FlutterFlow – Osnovne karakteristike	8
2.1.1. Flutter	9
2.1.2. Dart programski jezik	10
2.2. FlutterFlow – Prednosti i nedostaci	12
2.3. SupaBase - Pregled tehnologije	12
2.3.1. SupaBase autentikacija i autorizacija	14
2.3.2. SupaBase pohranjivanje datoteka	14
2.3.3. SupaBase studio pregled i značajke	14
3. PRISTUP PROBLEMU APLIKACIJE ZA CJENIKE	16
3.1. Razumijevanje potrebe razvoja aplikacija s FlutterFlowom i SupaBaseom	16
3.2. Postavljanje Projekta u FlutterFlowu	16
3.3. Integracija SupaBase baze podataka	17
4. PRAKTIČNI RAD – IZRADA APLIKACIJE ZA CJENIKE	18
4.1. Definiranje ciljeva i zahtjeva	18
4.2. Detaljni pregled funkcionalnosti i arhitekture aplikacije	19
4.2.1. Funkcionalnosti aplikacije	22
4.3. Razvoj aplikacije za cjenike koristeći FlutterFlow i SupaBase	22
4.3.1. FlutterFlow upravljanje stanjem i prilagođene funkcije	23
4.3.2. Pridruživanje akcije elementima	24
4.3.3. Animacije u FlutterFlowu	25
4.3.4. Testiranje i optimizacija FlutterFlow aplikacije	26
4.4. SupaBase za upravljanje podacima	27
4.5. SupaBase kreiranje tablica i pogleda	27
5. ZAKLJUČAK	30
LITERATURA	32
SAŽETAK	34
SUMMARY	35

POPIS SLIKA

Slika 1. Usporedba Fluttera s React-Native, Xamarin i Ionic-frameworkom (Stack Overflow)	10
.....	10
Slika 2. Vrijeme izvršavanja petlje (N=10) (nequalsonelifestyle).....	11
Slika 3. Top razvojni alati (Hacker News)	13
Slika 4. Schema Visualizer	15
Slika 5. Prikaz ekrana aplikacije: početni, narudžbe i detalji narudžbe	19
Slika 6. Pregled strukture aplikacije u storyboardu	20
Slika 7. Pregled SupaBase tablica i veza	21
Slika 8. Provider prikaz varijabli i konstanti (FlutterFlow Docs).....	23
Slika 9. Pridruživanje akcije elementima	25
Slika 10. Izbornik opcija animacija	26

POPIS KODOVA

Kôd 1. Definicija tablice roba.....	28
Kôd 2. Definicija pogleda narudzbe_det_view.....	29

1. UVOD

Sve veća popularnost mobilnih i mrežnih aplikacija zahtijeva inovativne alate i tehnologije kako bi se zadovoljile rastuće potrebe tržišta. U svijetu razvoja softvera, ključni čimbenici uspjeha uključuju brzinu razvoja, kvalitetu krajnjeg proizvoda te mogućnost prilagodbe različitim platformama. Odabir pravih tehnologija ključan je za uspješnu realizaciju projekta. Programeri se suočavaju s izazovima pronalaza alata koji će omogućiti ubrzani proces razvoja, kao i razvoj aplikacija za više platformi, bez žrtvovanja kvalitete i funkcionalnosti.

Zbog takvih izazova tehnologije poput FlutterFlowa privlače sve veću pažnju programerske zajednice. FlutterFlow je platforma za brzi razvoj aplikacija koja omogućuje izradu korisničkog sučelja bez potrebe za pisanjem kôda. Platforma je temeljena na Flutter okviru i Dart programskom jeziku koji omogućuje razvoj aplikacija visoke kvalitete. S druge strane SupaBase predstavlja moderno rješenje za baze podataka s fokusom na brzinu i jednostavnost upotrebe. Programeri mogu lako kreirati, uređivati i upravljati tablicama te definirati odnose između njih bez potrebe za velikim tehničkim znanjem. Osim toga SupaBase i FlutterFlow nude detaljnu dokumentaciju i veliku zajednicu korisnika koji pružaju podršku i resurse za učenje.

Ovaj rad istražuje osnovne karakteristike FlutterFlow tehnologije, analizira prednosti i nedostatke, te pruža uvid u njegovu integraciju sa SupaBase bazom podataka. U radu je također napravljena usporedba sličnih tehnologija te neke prednosti i mane u odnosu na njih.

Drugi dio rada posvećen je praktičnom dijelu, odnosno aplikaciji za cjenike. Opisane su funkcionalnosti i arhitektura same aplikacije te je dan uvid u njen razvojni proces. Istaknuta je upotreba FlutterFlowa za izradu korisničkog sučelja i kako su iskorištene njegove mogućnosti za izradu i prilagodbu elemenata sučelja. Također je prikazan način integracije sa SupaBase bazom podataka. Kroz proces izrade aplikacije istaknute su specifičnosti upravljanja stanjima u FlutterFlowu, dodavanje akcija elementima te testiranje i optimizacija aplikacije kako bi se osigurala njena pouzdanost i performanse.

U poglavlju 4 detaljnije su sagledane funkcionalnosti i arhitektura aplikacije te su demonstrirani koraci razvoja.

U zaključku su sažete ključne spoznaje i dojmovi istraživanja ističući doprinos FlutterFlow i SupaBase tehnologija u olakšavanju razvoja aplikacija.

Poseban naglasak stavljen je na sigurnosne aspekte i zaštitu podataka korisnika u kontekstu korištenja navedenih tehnologija. Kroz analizu i praktičan rad, pokazano je kako ove tehnologije mogu značajno unaprijediti proces razvoja, a pri tome smanjiti vrijeme i resurse potrebne za izradu kvalitetnih i skalabilnih aplikacija. Na kraju istraživanja navedene su smjernice za daljnje potencijalne izazove koji se mogu pojaviti tijekom razvoja aplikacija kao

što su integracija s ostalim sustavima , upravljanje stanjima aplikacije te rješavanje sigurnosnih problema uključujući implementaciju sigurnosnih mjera za zaštitu korisničkih podataka.

2. PREDSTAVLJANJE FLUTTERFLOW I SUPABASE TEHNOLOGIJA

Razumijevanje ovih tehnologija omogućit će čitateljima da donesu informativne odluke prilikom odabira alata za svoje razvojne projekte te da maksimalno iskoriste njihove mogućnosti. Kroz strukturirani pregled, analizirana je arhitektura komponente i funkcionalnosti obje tehnologije, s posebnim naglaskom na njihovu praktičnu primjenu u razvoju aplikacije. Posebna pažnja posvećena je Flutter okviru i Dart programskom jeziku, koji čine temelj FlutterFlow ekosistema. Razmotrene su prednosti i nedostaci FlutterFlowa pružajući uravnotežen pregled na platformu. Opisan je SupaBase pregled tehnologije uključujući njegov pristup autentikaciji i autorizaciji, pohranjivanje datoteka te SupaBase studio kao alat za upravljanje bazama podataka.

2.1. FlutterFlow – Osnovne karakteristike

FlutterFlow je platforma za razvoj mobilnih aplikacija, mrežnih aplikacija te aplikacija za stolna računala. Platforma je razvijena s ciljem pojednostavljenja procesa razvoja aplikacija za programere i dizajnere raznih razina iskustva. FlutterFlow omogućuje brzo stvaranje aplikacija, a moguće je čak i kreiranje aplikacija bez ikakvog iskustva s programiranjem. Glavna karakteristika je vizualni editor koji omogućuje brzo i jednostavno kreiranje aplikacija gdje korisnici mogu povlačiti i ispuštati elemente na ekran aplikacije, a platforma automatski generira potreban kôd. Uz to, FlutterFlow nudi napredne alate za otklanjanje grešaka koji omogućuju brzo identificiranje i rješavanje problema unutar aplikacije. FlutterFlow sadrži trgovinu u kojoj se mogu kupiti već gotovi predlošci aplikacija ili komponente za aplikaciju, a također sadrži veliki izbor predložaka i komponenti koje su besplatne te se mogu uređivati i koristiti u vlastitoj aplikaciji. Komponente su prilagodljive i mogu se lako konfigurirati kako bi odgovarale potrebama raznih aplikacija.

FlutterFlow ima značajku koja pomaže bržem stvaranju aplikacije, FlutterFlowAiGen. Pojednostavljeno govoreći to je umjetna inteligencija integrirana u FlutterFlow. Zadatak joj je da, nakon što se opiše izgled željene stranice, predložka ili komponente koju bi se željelo izraditi, umjetna inteligencija ponudi željeno rješenje, koje se može izmijeniti ili bez izmjena uključiti u aplikaciju. FlutterFlow omogućuje integraciju s različitim vanjskim uslugama, uključujući baze podataka i API-je što dodatno povećava fleksibilnosti i funkcionalnost aplikacija.

Platforma se redovito ažurira novim značajkama i poboljšanjima, temeljenim na povratnim informacijama korisnika, što osigurava da ostaje u koraku s najnovijim tehnološkim trendovima. Nadalje, nudi podršku za različite jezike i lokalizaciju, omogućujući programerima

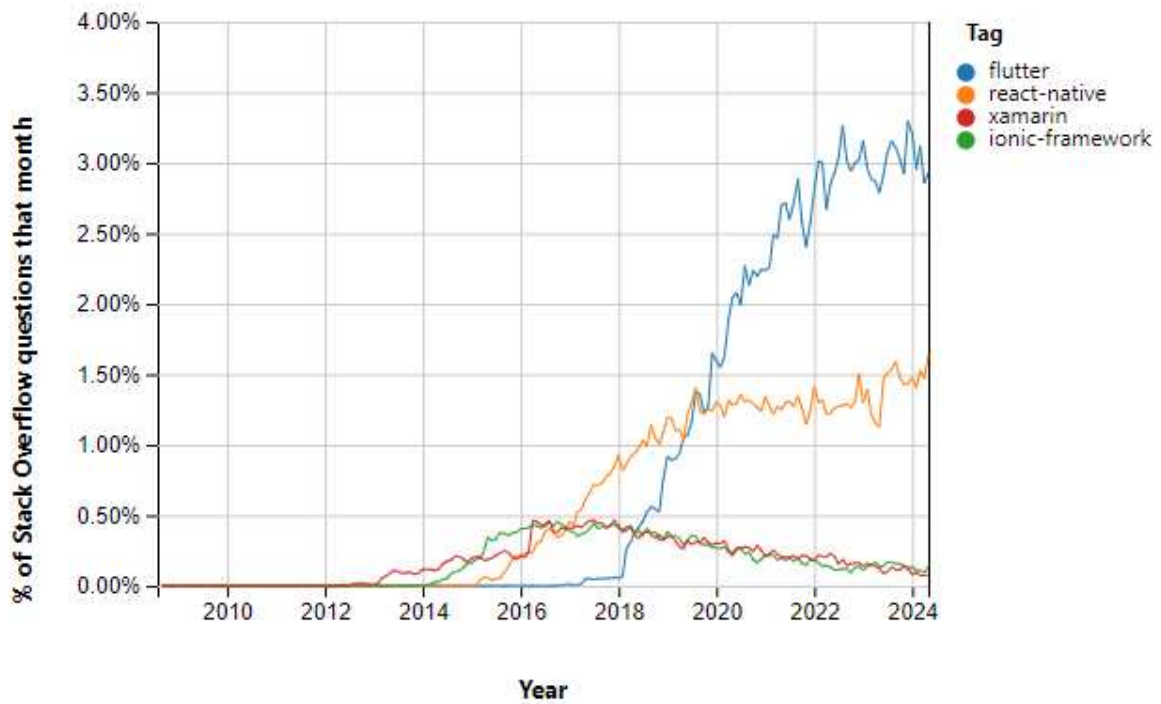
da kreiraju aplikacije koje su prilagođene globalnom tržištu. Uz podršku za razne platforme, FlutterFlow omogućuje jednostavno kreiranje responzivnih dizajna koji se lako prilagođavaju različitim veličinama ekrana i uređajima. Osim toga, FlutterFlow nudi detaljnu dokumentaciju i edukativne resurse koji pomažu korisnicima svih razina iskustava da maksimalno iskoriste platformu.

FlutterFlow je platforma s raznim alatima, uključujući integraciju baza podataka i generiranje kôda, ali može se reći da mu je glavna komponenta Flutter, koji će biti detaljnije objašnjen u sljedećem poglavlju.

2.1.1. Flutter

Flutter je programski okvir otvorenog kôda razvijen od strane Googlea, a glavni cilj mu je pojednostaviti proces razvoja aplikacija. Koristi programski jezik Dart kojeg je također razvio Google. Dart je moderan programski jezik dizajniran 2011. godine, jednostavan za učenje i koristi se za pisanje efikasnog kôda. Kroz razne biblioteke i dodatke Flutter omogućuje integraciju s različitim pozadinskim servisima.

Jedna od značajnih prednosti Fluttera je “hot reload”, odnosno brzo ažuriranje funkcionalnosti. Ova karakteristika omogućuje programerima da vide trenutne promjene u kôdu bez potrebe za ponovnim pokretanjem cjelokupne aplikacije. Flutter aplikacije se prevode u strojni kôd prilikom kompilacije, što rezultira bržim pokretanjem i manjom potrošnjom memorije u odnosu na interpretirane jezike. Osnovna građevna jedinica Fluttera je komponenta (engl. *widget*). U Flutteru postoje dvije osnovne vrste komponenti, a to su `StatelessWidget` i `StatefulWidget`. `StatelessWidget` je komponenta koja ne mijenja svoje stanje nakon što je kreirana, što znači da svi podaci koje koristi moraju biti unaprijed definirani i ne mogu se promijeniti tijekom vremena. `StatefulWidget`, za razliku od `StatelessWidget`a, može mijenjati svoje stanje tijekom vremena, što znači da može reagirati na događaje poput interakcije korisnika ili dolaznih podataka te ažurirati svoj prikaz u skladu s tim. Jedna od zanimljivih karakteristika Fluttera je podrška za animacije. Aplikacije s animacijama pružaju ugodno iskustvo prilikom njihovog korištenja, a vrlo često se animacijama pomaže korisnicima kako bi što lakše koristili aplikaciju. Flutter okvir može se usporediti s React-Nativeom zbog sličnog cilja, koji se očituje u tome da se isti kôd koristi za Android i za iOS platforme. U pogledu složenosti učenja, za React-Native potrebno je poznavanje JavaScripta i Reacta što može biti prepreka nekim početnicima. Slične platforme za izradu višeplatformskih aplikacija su Xamarin i Ionic, ali u odnosu na njih Flutter koristi Dart programski jezik koji podržava kompiliranje u strojni kôd što ima značajni utjecaj na performanse.



Slika 1. Usporedba Fluttera s React-Native, Xamarin i Ionic-frameworkom (Stack Overflow)

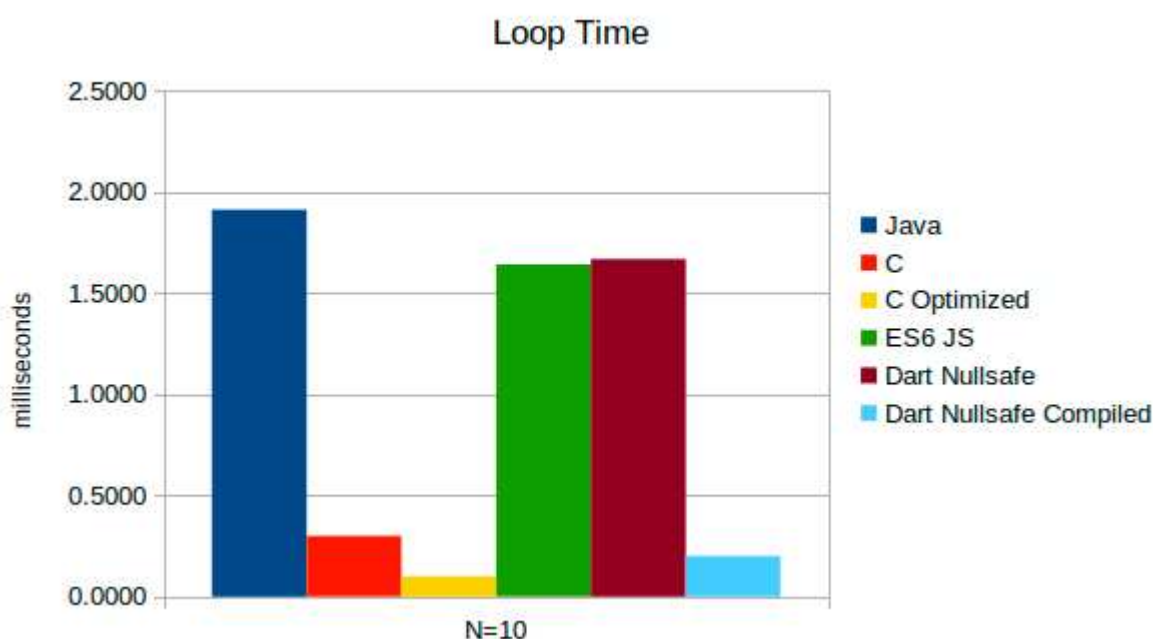
S obzirom na to da je baziran na komponentama, Flutter olakšava razvoj modularnih aplikacija koje su jednostavnije za održavanje i nadogradnju u dužem periodu. Ističe se po tome što pruža mogućnost izrade iznimno brzih animacija koje doprinose poboljšanju korisničkog iskustva i privlačnosti aplikacija. Grafikon (Slika 1) prikazuje postotak pitanja na Stack Overflowu po mjesecima za četiri razvojna okvira Flutter, React-Native, Xamarin i Ionic-framework. Podaci obuhvaćaju period od 2010. do 2024. godine. Grafikon omogućuje praćenje popularnosti navedenih okvira kroz vrijeme u zajednici programera.

2.1.2. Dart programski jezik

Jedna od ključnih karakteristika Dart je jednostavnost. Jezik je dizajniran s ciljem da bude čitljiv, razumljiv i pristupačan što ga čini prikladnim izborom za programere različitih razina iskustva. Svojom sintaksom Dart je sličan sintaksi JavaScripta, Jave ili C#-a, što olakšava prelazak s drugih programskih jezika na Dart. Statički je tipizirani jezik koji podržava dinamičku tipizaciju te pruža veću fleksibilnost u određenim situacijama. Objektno je orijentiran što ga čini pogodnim za izradu složenih i modularnih aplikacija. Može se kompilirati u strojni kôd (engl. *ahead of time compilation*) što poboljšava performanse aplikacija. Službeni repozitorij za Dart i Flutter pakete je Pub.dev, a omogućuje pronalaženje, instalaciju i dijeljenje

paketa s drugima.

Dart je moderan, sveobuhvatan i univerzalan programski jezik koji nudi mnoštvo prednosti programerima. Njegova brzina, jednostavnost korištenja i široka rasprostranjenost platforme čine ga dobrim izborom za izradu različitih aplikacija. Izvršavanje kôda direktno u mrežnom pregledniku moguće je uz pomoć online alata DartPad. Ovaj alat je razvijen s ciljem olakšavanja učenja i testiranja Dart programskog jezika bez potrebe za instalacijom lokalnog razvojnog okruženja. Slika 2. Vrijeme izvršavanja petlje (N=10) (nequalsonelifestyle) prikazuje vrijeme koje je potrebno za izvršavanje manjeg broja iteracija. Vrijeme petlje je mjereno u milisekundama. Ovi podaci mogu biti korisni pri donošenju odluka o izboru programskog jezika za projekte gdje je ključna brzina izvršavanja.



Slika 2. Vrijeme izvršavanja petlje (N=10) (nequalsonelifestyle)

2.2. FlutterFlow – Prednosti i nedostaci

FlutterFlow nudi brojne prednosti, ali ima i nekoliko nedostataka koje je važno uzeti u obzir prilikom korištenja ove tehnologije:

Prednosti FlutterFlowa su:

- jednostavan je za korištenje čak i za korisnike koji nemaju iskustva u razvoju aplikacija
- besplatan je za korištenje za osobne i nekomercijalne projekte
- omogućuje vizualno programiranje, što olakšava rad osobama bez velikog iskustva u programiranju
- alati za povlačenje i ispuštanje smanjuju vrijeme razvoja aplikacija
- nudi mogućnost integracije s različitim servisima putem akcija, čime se proširuje funkcionalnost aplikacije
- omogućuje praćenje performansi aplikacije i optimizaciju kôda kako bi se postigla veća brzina i responzivnost
- ima aktivnu zajednicu s brojnim resursima, podrškom i uputama.

Nedostaci FlutterFlowa su:

- korištenje samog vizualnog pristupa može ograničiti fleksibilnost za izradu složenijih elemenata, što zahtjeva određeno znanje za programsko prilagođavanje složenijih elemenata
- iako ima besplatni plan, za naprednije značajke i veće timove programera potrebno je platiti
- za njegovo korištenje potrebna je internetska veza budući da je mrežno-baziran.

2.3. SupaBase - Pregled tehnologije

SupaBase je započeo kao projekt s ciljem pružanja otvorene alternative Firebase platformi.

Nastao je 2020. godine te je brzo stekao popularnost među programerima, zbog svoje svestranosti i jednostavnosti korištenja. Ključne značajke SupaBasea uključuju:

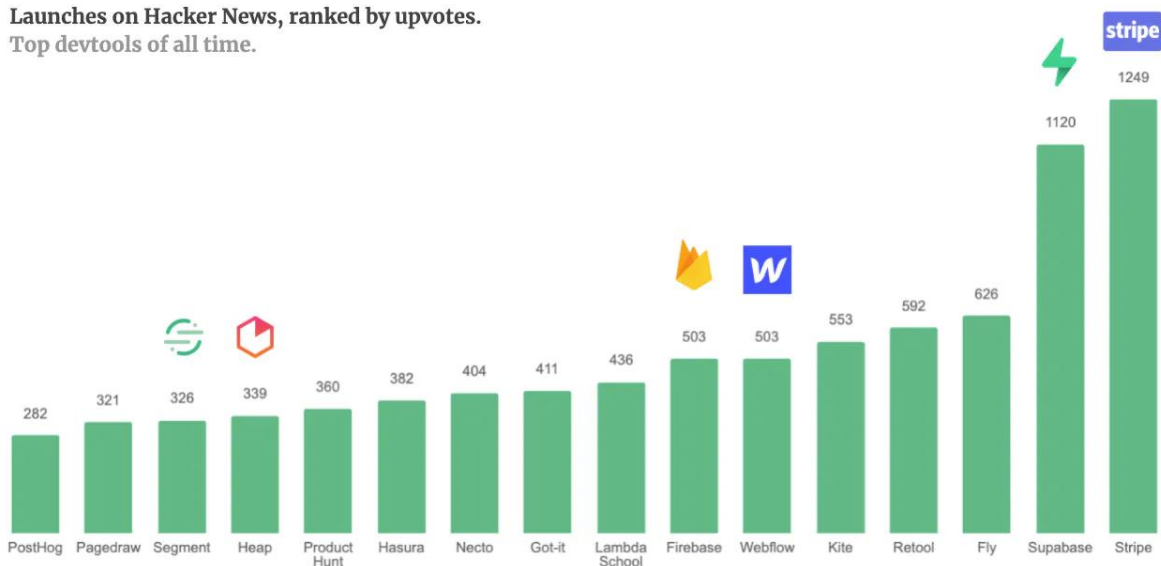
- baziran je na SQL bazi podataka (PostgreSQL), za razliku od Firebasea koji je baziran na NoSQL bazi podataka
- PostgreSQL baza podataka omogućava korištenje moćnih značajki SQL-a uz dodatak modernih alata za razvoj aplikacije
- nudi fleksibilne opcije za autentikaciju korisnika, uključujući e-mail i lozinku, Google GitHub i druge popularne servise
- omogućuje pohranu različitih tipova datoteka, uključujući slike, video i zvuk, te

podržava URL potpisivanje za siguran pristup datotekama

- podržava komunikaciju u stvarnom vremenu preko WebSocketsa, što omogućava aplikacijama ažuriranje korisničkog sučelja u stvarnom vremenu
- posjeduje SupaBase vektor, otvorenu bazu za vektorske podatke, idealnu za razvijanje aplikacija sa strojnim učenjem i pretragom
- ima dobro opisanu dokumentaciju i podržava migracije s različitih baza podataka, uključujući Firebase, Postgres, MySQL, Microsoft SQL Server i druge
- otvorenog je kôda, što korisnicima omogućuje pristup izvornom kôdu i mogućnost doprinosa njegovom razvoju.

Jedna od ključnih prednosti SupaBasea leži u tome što je temeljen na PostgreSQL bazi podataka, koja je jedan od najpouzdanijih sustava za upravljanje relacijskim bazama podataka. PostgreSQL nudi visok stupanj sigurnosti i besplatno je dostupan, što ga čini popularnim među programerima. Osim toga, PostgreSQL je objektno-relacijski sustav baza podataka (ORDBMS) koji nudi širok raspon naprednih značajki i funkcionalnosti te je podržan od strane velike i aktivne zajednice programera i korisnika. Za pomoć pri radu SupaBase ima alat koji se zove SupaBase asistent. Asistent je umjetna inteligencija koja pojednostavljuje upravljanje bazama podataka i interakciju s njima. SQL editor nudi pomoćne funkcionalnosti poput automatskog dovršavanja kôda, prikaza pogrešaka te ispravljanja pogrešaka. Slika 3 prikazuje najpopularnije alate na Hacker Newsu prema broju glasova. Iz grafikona je vidljivo da SupaBase po broju glasova ima daleko veću popularnost od Firebasea.

Launches on Hacker News, ranked by upvotes.
Top devtools of all time.



Slika 3. Top razvojni alati (Hacker News)

2.3.1. SupaBase autentikacija i autorizacija

Autentikacija i autorizacija su ključne komponente svake sigurne i funkcionalne aplikacije, a SupaBase pruža snažne alate za implementaciju ovih sigurnosnih mehanizama. Pod autentikacijom podrazumijevamo proces provjere identiteta, a autorizacija podrazumijeva razinu pristupa koju osoba ima nakon što je autorizirana. Za autentikaciju SupaBase omogućuje programerima da brzo i jednostavno postave sustav autentikacije za svoju aplikaciju. Bez obzira radi li se o pristupu s e-mailom i lozinkom, autentikaciju putem društvenih mreža ili integraciji s vanjskim OAuth poslužiteljima, SupaBase podržava navedene opcije. Autentikacija u SupaBaseu se temelji na JWT (engl. *JSON Web tokens*) standardu što omogućuje sigurno slanje informacija kao JSON objekta. Nakon što su korisnici autenticirani, SupaBase pruža mehanizme za kontrolu pristupa podacima, što je ključno za održavanje sigurnosti informacija. SupaBase nudi opciju kontrole pristupa na temelju uloga (engl. *role-based access control*), što omogućuje administratorima da dodijele određene uloge korisnicima.

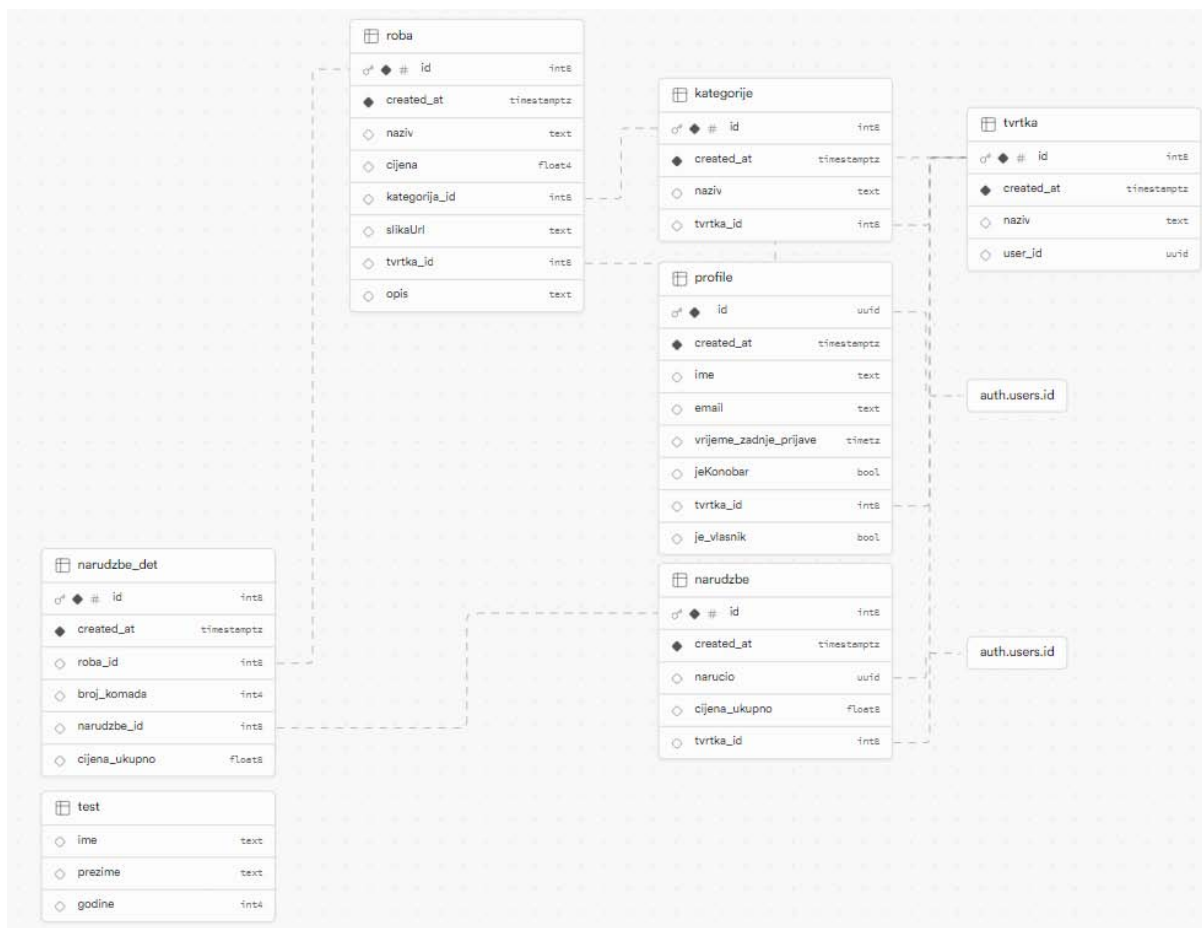
2.3.2. SupaBase pohranjivanje datoteka

SupaBase pruža jednostavan i učinkovit način za pohranu datoteka u oblak, omogućujući lako učitavanje, preuzimanje i upravljanje datotekama. Za pohranu datoteka SupaBase koristi takozvane kante (engl. *bucket*), koje služe za bolju organizaciju kod pohranjivanja datoteka. Korištenjem kanti može se kategorizirati i razvrstati datoteke prema različitim kriterijima što olakšava njihovo upravljanje i pristup. Svaka kanta ima postavke pristupa i dozvola, što omogućuje kontroliranje tko može vidjeti i upravljati datotekama unutar određene kante. SupaBase također obuhvaća i Image optimizer, što je funkcionalnost koja služi za optimizaciju i komprimiranje datoteka pri prijenosu podataka čemu se smanjuje opterećenje i povećava brzina pristupa. SupaBase također pruža jednostavne API-je za učitavanje i preuzimanje datoteka.

2.3.3. SupaBase studio pregled i značajke

SupaBase studio je vizualni alat za upravljanje bazama podataka koji olakšava rad sa SupaBaseom. Sastoji se od jednostavnih sučelja za izvršavanje SQL upita, pregledavanje tablica i podataka te upravljanje korisnicima i dozvolama. Jednostavan je za korištenje pa ga tako i početnici mogu bez problema koristiti. Omogućuje izmjenu, brisanje i dodavanje podataka u tablice, kao i prikazivanje svih tablica u bazi podataka s mogućnošću pretraživanja i filtriranja podataka. Omogućuje dodavanje različitih izvještaja s njihovim vizualnim pregledima, kako bi se na temelju podataka moglo efikasnije upravljati bazom podataka. U sam SupaBase studio ugrađeno je mnoštvo različitih alata pa je tako jedan od značajnih alata i

SupaBase AI za brže generiranje željenih rezultata poput kreiranja tablica ili pogleda. Schema Visualizer je još jedna značajka koja omogućuje vizualni pregled tablica s njihovim vezama i tipovima podataka. Slika 4 prikazuje tablice s popisom atributa, a linije između njih označavaju kako su tablice povezane ili u kakvom su odnosu jedna prema drugoj. Ovakav dijagram poznat je kao dijagram entiteta i odnosa te se koristi za vizualno prikazivanje podataka.



Slika 4. Schema Visualizer

3. PRISTUP PROBLEMU APLIKACIJE ZA CJENIKE

U modernom poslovnom okruženju, aplikacije za upravljanje cjenicama postaju ključni alat za tvrtke koje žele učinkovitije pratiti i ažurirati cijene proizvoda. Ovaj dio rada opisuje osnovno razumijevanje razvoja aplikacije sa FlutterFlowom i SupaBaseom, postavljanje projekta u Flutterflowu te samu integraciju sa Supabase bazom podataka. Opisano je kako Supabase moderna backend platforma može biti integrirana s FlutterFlowom, pružajući robusnu infrastrukturu za pohranu i manipulaciju podacima o cjenicama.

3.1. Razumijevanje potrebe razvoja aplikacija s FlutterFlowom i SupaBaseom

Razvoj aplikacija predstavlja kompleksan proces koji zahtijeva duboko razumijevanje potreba korisnika. FlutterFlow pruža platformu za vizualni razvoj koji omogućuje prilagodbu sučelja prema specifičnim zahtjevima.

Razvoj aplikacija s tradicionalnim tehnologijama poput React Nativea može biti složen i zahtjevan, te integracija backend funkcionalnosti s frontendom može biti složena. Supabase s druge strane pruža backend rješenje koje je ključno za upravljanje podacima. Integracija Supabasea omogućuje efikasno pohranjivanje i dohvaćanje informacija o proizvodima, cjenicama te drugim relevantnim podacima. Ova funkcionalnost ključna je za osiguravanje točnih i ažuriranih informacija o cjenicama.

3.2. Postavljanje Projekta u FlutterFlowu

Prvi korak u procesu je registracija na FlutterFlow platformu. Proces registracije je jednostavan, može se registrirati putem e-maila, ili koristiti vlastiti Google, Apple, Microsoft ili GitHub račun za registraciju.

Nakon registracije postoji opcija kreiranja novog projekta. Projekt može biti kreiran kao prazan projekt ili uz pomoć nekih unaprijed zadanih predložaka. Izborom praznog projekta postavljaju se sve zadane postavke na projektu poput zadane teme, zadane navigacijske trake, početne stranice te ostale značajke. Moguće je dodati dopuštenja za aplikaciju ako je potrebno, poput dopuštenja za korištenje kamere, mikrofona, lokacije i drugih dopuštenja. Od značajnijih stvari koje se mogu postaviti su sredstva aplikacije (engl. *App Assets*), a tu se mogu odabrati i slika koja će biti postavljena za ikonu aplikacije, uvodna slika u aplikaciju (engl. *Initial splash image*), te slika koja će se pojaviti u slučaju neke pogreške (engl. *Error placeholder image*). FlutterFlow omogućuje izradu višejezične aplikacije pa tako programer može izabrati neke od jezika koje su ponuđeni, a ako željeni jezici nisu navedeni, postoji mogućnost da se ručno unesu prijevodi za tekstove. Ako aplikacija zahtijeva integraciju plaćanja, FlutterFlow podržava nekoliko sustava za plaćanje. Trenutno su to Stripe, Braintree, RevenueCat i RazorPay. Od

ostalih alata koje je možda poželjno postaviti kod postavljanja projekta su Algolija, Google Analytics, Google Maps, AdMob, Mux Livestream ili OneSignal. Integracija s Githubom nije dostupna u besplatnoj verziji, za ovu opciju potrebno je nadograditi osnovni plan na standard ili pro verziju. Korištenjem GitHub integracije može se pohraniti projekt u GitHub repozitorij i iskoristiti sve prednosti koje GitHub nudi.

3.3. Integracija SupaBase baze podataka

Ova integracija omogućuje optimalno iskorištavanje prednosti oba alata, pružajući efikasno rješenje koje zadovoljava zahtjeve korisnika.

Trenutno FlutterFlow podržava Firebase, SupaBase i SQLite baze podataka. Za povezivanje FlutterFlow projekta sa SupaBase bazama podataka u FlutterFlowu, pod postavke i integracije u integracijama postoji SupaBase koju je potrebno izabrati te uključiti. Za dodavanje su potrebni API URL i Anon Key koje se nalaze na SupaBase računu pod opcijama projekta. API URL je osnovna adresa putem koje aplikacija komunicira sa SupaBase bazom podataka.

Kada se kreira novi projekt u SupaBase konzoli dodjeljuje se jedinstveni URL koji predstavlja krajnju točku za sve API pozive vezane za taj projekt. Anon ključ je javni API ključ, a omogućuje anonimni pristup SupaBase bazi podataka. Anon ključ se obično koristi za operacije koje ne zahtijevaju autentikaciju korisnika, kao što je na primjer čitanje javno dostupnih podataka. Ako je potrebna autentikacija nakon integracije SupaBasea, u FlutterFlow opcijama se može jednostavno postaviti stranica za prijavu i stranica koja će se prikazati nakon uspješne prijave korisnika.

Jednostavnost upotrebe i bogate funkcije čine SupaBase odličnom opcijom za razvoj mobilnih i mrežnih aplikacija s FlutterFlowom.

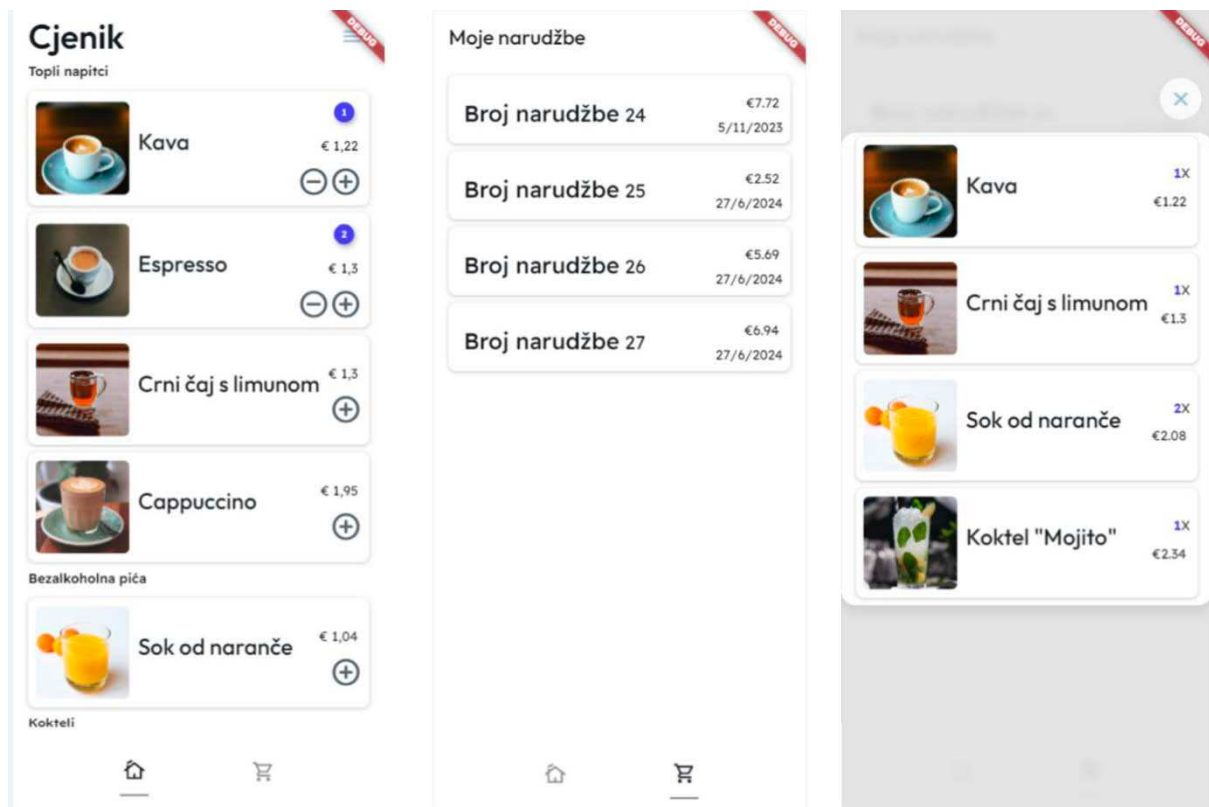
4. PRAKTIČNI RAD – IZRADA APLIKACIJE ZA CJENIKE

Ovo poglavlje predstavlja praktični dio rada, detaljno opisujući proces izrade aplikacije za cjenike koristeći FlutterFlow i SupaBase tehnologije. Kroz strukturirani prikaz prikazano je kako se mogućnosti ovih platformi primjenjuju u stvarnom razvojnom procesu. Definirani su ciljevi i zahtjevi aplikacije, postavljajući jasne smjernice za funkcionalnosti koje aplikacija obuhvaća. Ova faza je ključna za usmjeravanje razvoje i osiguravanje da krajnji proizvod odgovara potrebama korisnika. Korišten je Storyboard za vizualiziranje strukture aplikacije, pokazujući kako su različiti ekrani i komponente integrirani. Posebna pažnja posvećena je procesu pridruživanja akcija elementima sučelja, što je ključno za stvaranje interaktivne i dinamične aplikacije. Vizualni efekti koji su značajni za kvalitetu korisničkog iskustva opisani su u dijelu animacije u FlutterFlowu. Objašnjen je proces kreiranja tablica i pregleda, te sigurnost podataka koristeći SupaBaseove napredne značajke poput Row Level Security.

4.1. Definiranje ciljeva i zahtjeva

Ciljevi i zahtjevi aplikacije definirani su kako bi se osigurala optimalna funkcionalnost i korisničko iskustvo. Glavni ciljevi uključuju:

- omogućiti korisnicima stvaranje, uređivanje i brisanje cjenika putem sučelja
- osigurati privlačan dizajn sučelja kako bi korisnici lako koristili aplikaciju
- omogućiti korisnicima registraciju i prijavu putem sigurnog sustava autentikacije
- implementirati funkcionalnosti poput kreiranja, čitanja, ažuriranja i brisanja podataka
- prikazati detaljan pregled proizvoda sa slikom i opisom
- osigurati pravilno prikazivanje aplikacije na različitim uređajima
- podijeliti aplikaciju na tri uloge: vlasnik, konobar i gost, te ovisno o ulozi korisnicima omogućiti određene preglede i funkcionalnosti
- na početnoj stranici cjenika omogućiti pregledavanje robe s cijenama
- omogućiti otvaranje detalja robe kada korisnik odabere artikl na početnoj stranici.

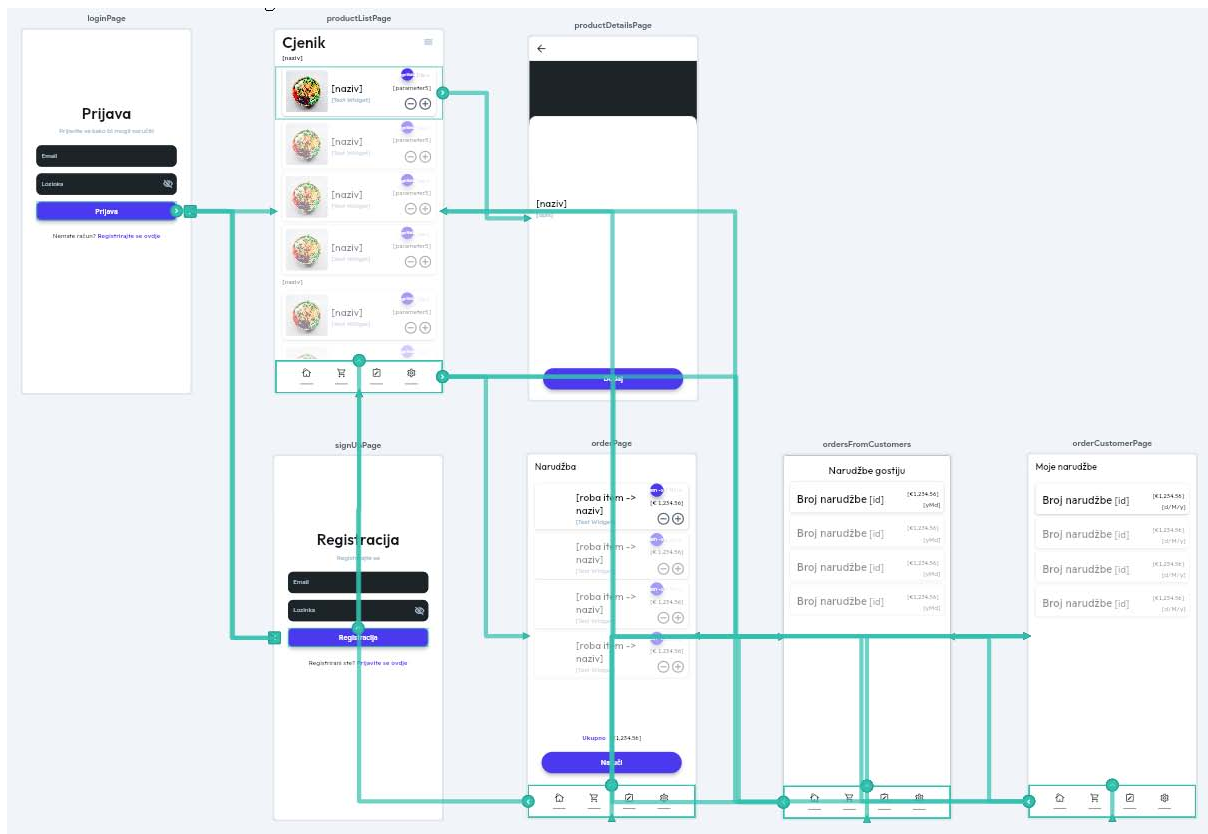


Slika 5. Prikaz ekrana aplikacije: početni, narudžbe i detalji narudžbe

Slika 5 prikazuje tri različita ekrana aplikacije Cjenik. Na prvom ekranu prikazana je lista pića dostupnih za narudžbu. Svako piće je prikazano sa slikom, nazivom i cijenom, a pića su grupirana po kategorijama. Drugi ekran s naslovom Moje narudžbe prikazuje listu narudžbi s cijenama, datumom kada je narudžba napravljena i ukupnom cijenom narudžbe. Treći ekran prikazuje detalje narudžbe uključujući količinu i pojedinačne cijene.

4.2. Detaljni pregled funkcionalnosti i arhitekture aplikacije

FlutterFlow Storyboard je vizualni alat koji omogućuje uvid u aplikaciju, planiranje toka i navigaciju kroz Flutter aplikaciju, te omogućuje pregled svih ekrana aplikacije na jednom mjestu (Slika 6)



Slika 6. Pregled strukture aplikacije u storyboardu

Aplikacija sadrži nekoliko osnovnih ekrana koji korisnicima omogućuju različite funkcionalnosti. Ovi ekrani uključuju:

1. Prijava – Prijava već registriranog korisnika
2. Početna – Pregled robe s cijenom
3. Detalji Robe – Prikaz naziva robe sa slikom i detaljima
4. Registracija – Registracija korisnika
5. Pregled robe – Ekran na kojem korisnik može pregledati narudžbu, promijeniti narudžbu i naručiti
6. Narudžba – Artikli koje je korisnik prethodno dodao za narudžbu
7. Povijest Narudžbe – Prikaz naručenih artikala.

4.2.1. Funkcionalnosti aplikacije

Aplikacija je osmišljena kako bi omogućila vlasnicima i osoblju kafića učinkovito upravljanje cjenicama i ponudom proizvoda, te kako bi olakšala korisnicima pregled i pristup informacijama o cijenama. Temelji na tri uloge: gost, konobar i vlasnik, od kojih svaka ima specifične funkcije i preglede.

Funkcionalnosti za gosta:

- Početna stranica: Gost ima pregled svih dostupnih artikala s cijenama. Klikom na ikonu plus, gost može dodati artikle u narudžbu. Nadalje, gosti mogu pregledavati detalje odabranog artikla klikom na njegovu sliku.
- Narudžba: Na stranici narudžba gost može pregledavati artikle dodane u narudžbu, mijenjati količine i vidjeti ukupnu cijenu.
- Izbornik: Iz izbornika, gost ima pristup postavkama i povijesti narudžbi. Klikom na povijest narudžbi, otvara se dijalog s detaljima prethodnih narudžbi.

Funkcionalnosti za konobara:

- Početna stranica: Konobar ima pregled artikala, ali nema mogućnosti dodavanja artikla u narudžbu. Ima mogućnost izmjene artikla.
- Dodavanje robe i kategorije: Konobar može dodavati nove artikle i kategorije proizvoda putem izbornika.

Funkcionalnosti za vlasnika:

- Vlasnik ima iste funkcionalnosti kao i konobar, no za razliku od konobara vlasnik može dodjeljivati korisnicima uloge gosta odnosno konobara.

4.3. Razvoj aplikacije za cjenike koristeći FlutterFlow i Supabase

Prilikom razvoja programeri imaju na raspolaganju značajku *privuci i ispusti* komponente (eng. *widžete*). Kako je komponenta osnovna jedinica Fluttera tako postoji veliki izbor predefiniраниh komponenti. Neke od osnovnih komponenti su:

- Tekst (Text) - koristi se za prikazivanje teksta na stranici.
- Red (Row) - organizira komponente horizontalno, slično kao redak u tablici.
- Kolona (Column) - organizira komponente vertikalno, kao stupci u tablici.
- Stog (Stack) - omogućuje preklapanje više komponenti na istom mjestu.
- Kontejner (Container) - definira okvir za okvir za druge komponente i omogućuje postavljanje stilova i pozadine.

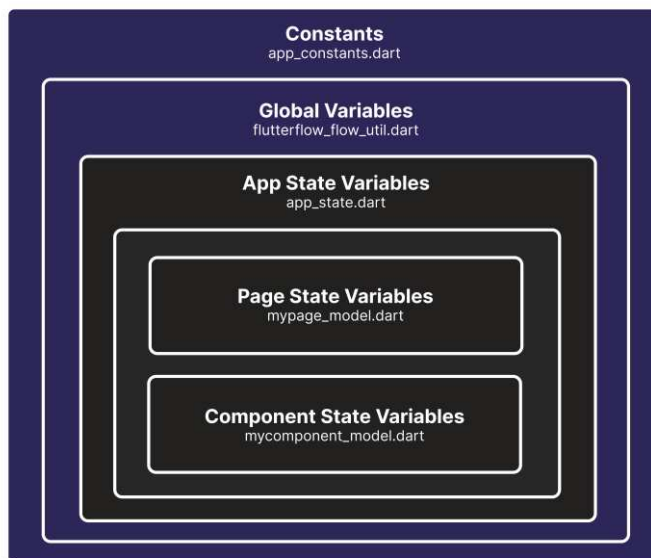
Komponente se međusobno kombiniraju kako bi stvorili stablo komponenti. Ono prikazuje koliko komponenti ima na stranici i kako su povezane (odnos roditelj-dijete). Stablo

komponenti pomaže razumjeti strukturu aplikacije te kako su stvari vizualno posložene.

Programer se može poslužiti i s ugrađenim alatom umjetne inteligencije (FlutterFlowAI Gen) u kojem može opisati izgled stranica pa će mu alat generirati sadržaj na temelju tog opisa.

4.3.1. FlutterFlow upravljanje stanjem i prilagođene funkcije

Upravljanje stanjem u FlutterFlowu implementirano je na više razina, ovisno o opsegu i potrebama aplikacije.



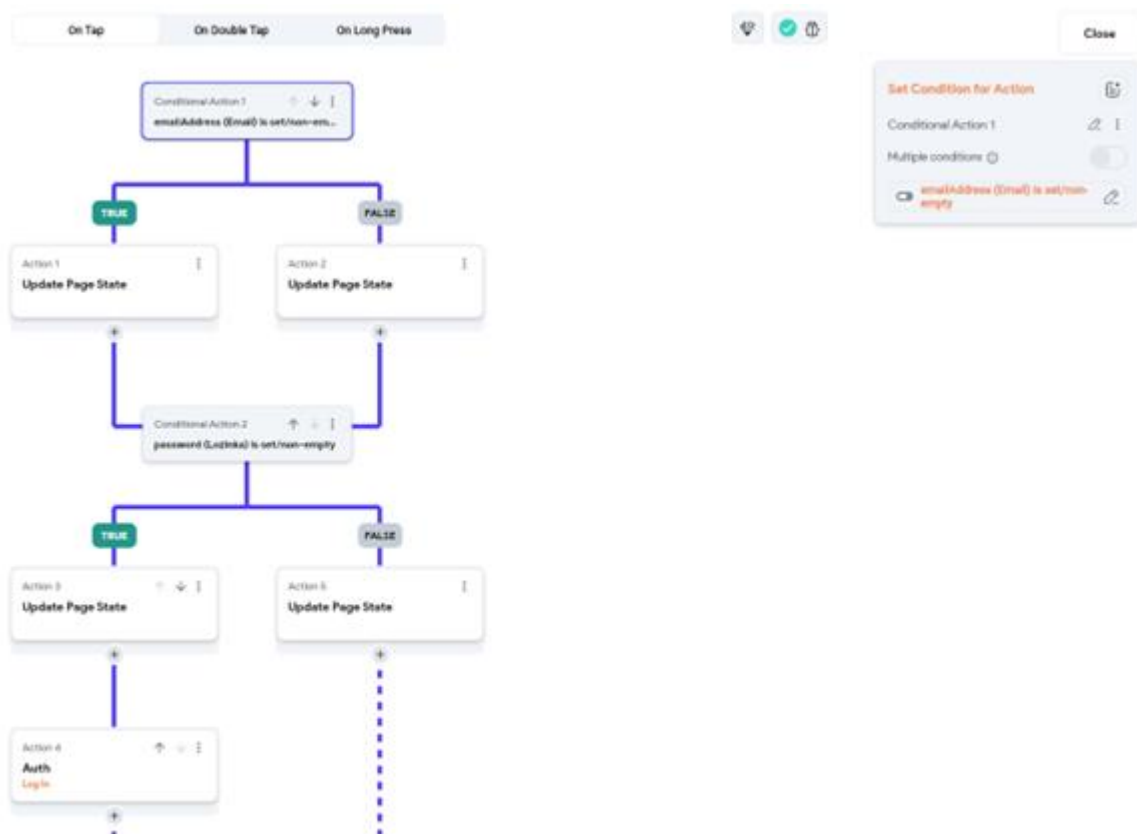
Slika 8. Provider prikaz varijabli i konstanti (FlutterFlow Docs)

U radu je korišten paket Provider (Slika 8) kako bi se upravljalo stanjem komponente, stanjem stranice i stanjem aplikacije. Provider je popularan paket za upravljanje stanjem i dijeljenjem podataka unutar aplikacije. Omogućuje dijeljenje i ažuriranje podataka u različitim komponentama U aplikaciji cjenik kao primjer za App State varijablu, dodana je varijabla userRole koja je potrebna za određivanje uloga u aplikaciji pa se tako novo registriranom korisniku dodjeljuje uloga gost. Ovisno o dodijeljenoj ulozi korisnik ima određena dopuštenja i određene funkcionalnosti koje su značajke određene uloge. Mehanizam za upravljanje globalnim stanjem u FlutterFlow aplikaciji je FFAppState. To je struktura koja omogućuje da se podaci lako dijele te da im se pristupa bilo gdje u aplikaciji. Kada se vrijednosti u FFAppState promjene, odgovarajući dijelovi automatski se ažuriraju što omogućuje dinamičke promjene te bolje korisničko iskustvo. Prilagođene funkcije u FlutterFlowu se koriste ako je potrebno implementirati složenije funkcionalnosti aplikacije. Pišu se u Dart programskom jeziku, a ako korisnik nema iskustva s Dart programskim jezikom u FlutterFlow editoru također postoji AI

Code Assistant kojem je potrebno dati opis željene funkcionalnosti funkcije kako bi generirao Dart funkciju koja se može odmah testirati. Ako generirana funkcija nije zadovoljavajuća, može se dati detaljniji opis i ponoviti generiranje.

4.3.2. Pridruživanje akcije elementima

Na komponentama postoji mogućnost dodavanja akcije. Akcije se razlikuju ovisno o kojem se elementu radi, a postoje i već predefinirane akcije koje je moguće iskoristiti. Ako predefinirane akcije ne zadovoljavaju korisnikove potrebe, moguće je definiranje prilagođenih akcija. Također još postoje i prilagođene funkcije i prilagođene komponente koje korisnik može definirati ako za tim ima potrebe. Za dodavanje i uređivanje akcija korišten je action flow editor, pomoćni alat za bržu izradu akcija aplikacije. Akcije ovise o elementima na koje se pridodaje akcija, pa tako za element tipka imamo izbor akcije na dodir, dvostruki dodir ili dugi pritisak. U editoru akcija se može dodavati, odnosno ubacivati akcija po potrebi te ako je potrebno akcija se može privremeno stopirati (engl. *disable action*), a također postoji mogućnost kopiranja, izrezivanja i brisanja akcija. U akcijskom editoru mogu se dodavati i petlje i uvjeti kako bi se kreirale složenije logike unutar aplikacije. Ovo omogućuje dinamičko upravljanje tokom akcija, čime se povećava fleksibilnost i prilagodljivost aplikacije. Korištenje action flow editora znatno ubrzava proces izrade složenih aplikacijskih tokova, omogućujući korisnicima da brzo iteriraju i testiraju različite scenarije unutar svojih aplikacija. Na Slici 9 možemo vidjeti grafički prikaz uvjeta koji ispituje je li vrijednost jednaka, ovisno o tome izvršava li se jedna ili druga strana. Također se može vidjeti koje se sve vrste akcija mogu dodati u editoru.



Slika 9. Pridruživanje akcije elementima

Ako programer nema puno znanja o programiranju, unutar aplikacije dostupan je alat Code Copilot (AI Code Assistant) kojem je potrebno dati opis što bi navedena funkcija trebala raditi te će na temelju opisa copilot generirati funkciju.

4.3.3. Animacije u FlutterFlowu

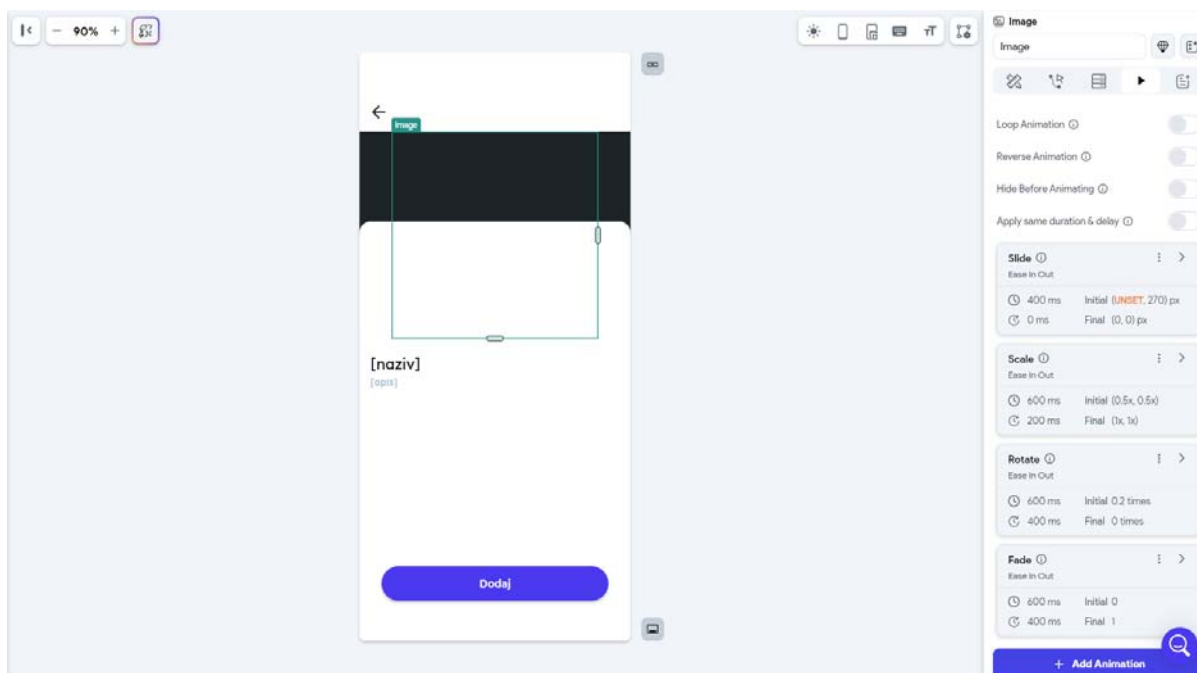
Aplikacije mogu raditi bez animacija, ali nadopunjene animacijama čine korisnička sučelja privlačnijima i u posljednje vrijeme su neizostavan i ključni dio izrade dizajna. FlutterFlow nudi vizualno okruženje koje omogućuje jednostavnu izradu animacija (Slika 10).

U FlutterFlowu se već nalaze predefinirane animacije poput klizanja, pulsiranja i rotiranja.

Trajanje animacije i ostali parametri mogu se precizno prilagoditi, a animacije se mogu kombinirati kako bi se dobili složeniji vizualni efekti. Za naprednije korisnike FlutterFlow nudi mogućnost pisanja kôda za animacije što daje potpunu kontrolu i omogućuje stvaranje jedinstvenih efekata koji nisu dostupni kroz vizualno okruženje.

Neki od primjera korištenja animacija su korištenje animacija za prijelaz između različitih stranica u aplikaciji, animiranje elemenata korisničkog sučelja, animacije kod interakcije korisnika kada korisnici dodirnu ili prelistavaju elemente aplikacije i slično. Trajanje, vrsta

animacije i ostale opcije mogu se prilagoditi u editoru za animacije. Nakon izmjena, moguće je pokrenuti pregled pojedinačne animacije ili pregled svih povezanih animacija. Animacije su moćan alat i koriste se za poboljšanje korisničkog sučelja u aplikaciji.



Slika 10. Izbornik opcija animacija

4.3.4. Testiranje i optimizacija FlutterFlow aplikacije

Testiranje i optimizacija su ključni aspekti razvoja bilo koje aplikacije pa tako i Flutter aplikacije.

FlutterFlow ima opciju koja može dati prijedloge za poboljšanje aplikacije kao npr. definiranje maksimalne veličine slike kod učitavanja slika. Tako se sprječava smanjenje performansi i usporavanje rada aplikacije koje može nastati ako su učitane slike veće od potrebnih.

Također, pruža prijedloge za uklanjanje bespotrebnih upita na bazu podataka te ostalih značajki koje bi mogle usporiti aplikaciju.

Neka od najčešćih testiranja uključuju:

- Jedinični testovi: koriste se za testiranje pojedinačnih komponenti aplikacije
- UI testovi: koriste se za testiranje korisničkog sučelja aplikacije
- Funkcionalni testovi: koriste se za testiranje funkcionalnosti aplikacije
- Testovi performansi: koriste se za mjerenje performansi aplikacije.

U samom FlutterFlow editoru moguće je prebaciti se na različitu veličinu ekrana te bez pokretanja vidjeti responzivnost određenih elemenata na određenom zaslonu aplikacije.

Testiranje i optimizacija FlutterFlow aplikacija su važan dio procesa razvoja. Koristeći različite

tehnike testiranja i optimizacije, može se osigurati da aplikacija radi besprijekorno i što je brže moguće.

4.4. SupaBase za upravljanje podacima

SupaBase pruža RESTful API za interakciju s bazom podataka i upravljanje autentikacijom u aplikaciji. RESTful API je automatski generiran na temelju baze podataka što znači da nije potrebno pisati dodatni kôd za definiranje operacije nad podacima. SupaBase RESTful API pruža visoku razinu sigurnosti kao što su HTTPS enkripcija i integracija s popularnim alatima za upravljanje pravima pristupa.. Jedan od ključnih aspekata u sigurnosti u SupaBaseu je Row Level Security (RLS). RLS omogućava kontrolu pristupa podacima na razini redova u tablici, osiguravajući da korisnici mogu vidjeti i mijenjati samo one podatke za koje imaju dozvolu. SupaBase s Row Level Security predstavlja mehanizam sigurnosti za izradu aplikacija koje zadovoljavaju visoke standarde za sigurnost i privatnost.

4.5. SupaBase kreiranje tablica i pogleda

Kod razvijanja aplikacija općenito pa tako i kod aplikacije za cjenike, među prvim koracima u stvaranju baze podataka je definiranje struktura tablica. Tablice se u SupaBaseu mogu kreirati koristeći korisničko sučelje u editoru tablica ili koristeći SQL editor. Ukoliko se odabere SQL editor postoji mogućnost korištenja SupaBase AI, koji pomaže kod bržeg kreiranja tablica i pogleda na osnovu opisa potrebne tablice ili pogleda.

Tablice potrebne za aplikaciju kreirane su u javnoj shemi za pohranu podataka kako bi im se moglo pristupiti putem API-ja.

Pogledi su virtualne tablice koje predstavljaju rezultat SQL upita nad jednom ili više tablica.

U kôdu 1 prikazan je kôd definicije tablice roba. Definicija tablice može se u svakom trenutku pogledati u SupaBase editoru. Definicija se otvara u načinu samo za čitanje. Tablica je kreirana kroz vizualni editor, dok je nakon uspješnog kreiranja tablice sam SupaBase generirao definiciju tablice.

Kreirana tablica sastoji se od sljedećih atributa:

1. id: Primarni ključ identiteta, tipa int8
2. created_at: Tipa timestamptz, koristi se za spremanje vremena dodavanja novog retka u tablicu, uključujući vremensku zonu
3. naziv: Polje koje sadrži naziv robe
4. cijena; Decimalni tip podatka koji predstavlja cijenu robe
5. opis: Tekstualni opis robe
6. kategorija_id: Strani ključ koji povezuje tablicu s kategorijama

7. tvrtka_id: Strani ključ koji povezuje tablicu s tvrtkom
8. slikaUrl: Tekstualni tip podatka za pohranu URL-a do slike robe. Ovo polje je definirano sa zadanom vrijednošću ako slika nije definirana.

```
create table
public.roba (
    id bigint generated by default as identity,
    created_at timestamp with time zone not null default now(),
    naziv text null,
    cijena real null,
    kategorija_id bigint null,
    "slikaUrl" text null default
'https://faddzyvmpmlstxpnqwsj.supabase.co/storage/v1/object/public/roba/emptyGlass.jpg'::text,
    tvrtka_id bigint null,
    opis text null default ''::text,
    constraint roba_pkey primary key (id),
    constraint roba_kategorija_id_fkey foreign key (kategorija_id) references kategorije (id),
    constraint roba_tvrtka_id_fkey foreign key (tvrtka_id) references tvrtka (id)
) tablespace pg_default;
```

Kôd 1. Definicija tablice roba

U situaciji gdje nije bilo potrebno nešto mijenjati, a potrebni su podaci iz više tablica korišteni su pogledi za dohvat podataka.

```
create view
  public.narudzbe_det_view as
select
  nd.id,
  nd.narudzbe_id,
  nd.broj_komada,
  nd.cijena_ukupno,
  r.tvrtka_id,
  r.naziv,
  r."slikaUrl"
from
  narudzbe_det nd
join roba r on r.id = nd.roba_id;
```

Kôd 2. Definicija pogleda narudzbe_det_view

Pogledi u SupaBaseu mogu uključivati složene operacije kao što su pridruživanja različitih tablica i agregacije, omogućujući nam da efikasno izvlačimo informacije iz više izvora podataka. Idealni su za situacije u kojima trebamo konzistentan i strukturiran pristup podacima, jer je omogućeno da definiramo logičke operacije koje se automatski primjenjuju pri svakom upitu na bazu podataka. Kreiranje pogleda može se obaviti korištenjem vizualnog editora ili korištenjem SQL editora, uz mogućnost korištenja SupaBase AI za ubrzanje procesa definiranja pogleda. Kôd 2 prikazuje definiciju pogleda narudzbe_det_view koji se sastoji od dvije tablice.

5. ZAKLJUČAK

Kroz ovaj praktični rad prikazana je izrada aplikacije za cjenike koristeći FlutterFlow za dizajniranje korisničkog sučelja i SupaBase za backend razvoj. Ova kombinacija tehnologija omogućila je brzu i efikasnu izradu aplikacije s naprednim funkcionalnostima, a autentikacija je ostvarena korištenjem SupaBase platforme.

SupaBase, kao backend platforma otvorenog kôda, pokazala se kao izvrsna alternativa za Firebase. Pruža jednostavnu integraciju s aplikacijama razvijenim u FlutterFlowu, omogućavajući brzo postavljanje i upravljanje bazama podataka bez potrebe za složenim konfiguracijama. Korisnici mogu koristiti SupaBase za stvaranje aplikacija u stvarnom vremenu zahvaljujući podršci za WebSocket protokol, što omogućava trenutno ažuriranje podataka u aplikaciji.

FlutterFlow nadograđuje Flutter okvir, olakšavajući razvoj grafičkog sučelja a SupaBase dodaje široki spektar funkcionalnosti, kao što su spremanje podataka i datoteka, te autentikaciju. SupaBase nudi ugrađene funkcije za sigurnost i autorizaciju, što pomaže u zaštiti korisničkih podataka i osigurava da samo ovlašteni korisnici mogu pristupiti određenim podacima i datotekama.

Aplikacija za cjenike predstavlja samo jedan od mnogih primjera gdje se ove tehnologije mogu uspješno primijeniti za razvoj modernih aplikacija. FlutterFlow i SupaBase su idealni za razvoj aplikacija za više platformi pri čemu se isti kôd može koristiti za različite platforme. Također olakšavaju implementaciju ključnih funkcionalnosti aplikacije, kao što su registracija i prijava korisnika, kreiranje, čitanje i brisanje podataka.

Primjena FlutterFlowa i SupaBasea u ovom projektu pokazala je kako se koristeći ove alate može postići visoka razina produktivnosti i kvalitete u razvoju aplikacija. Brzina razvoja i fleksibilnost u prilagodbi korisničkog sučelja kao i moćne backend funkcionalnosti čine ove tehnologije izuzetno korisnim za široki spektar projekata, od malih aplikacija do kompleksnih sustava.

Razvoj aplikacije pomoću FlutterFlowa pokazao se jednostavan zato što se u editoru sve pokušava prikazati vizualno, a to znatno ubrzava razvoj same aplikacije. FlutterFlowAiGen kao napredna značajka FlutterFlowa, koja koristi umjetnu inteligenciju, pokazala se izuzetno korisna za generiranje komponenti, elemenata i funkcija. SupaBase u svom virtualnom editoru također olakšava kreiranje, editiranje i brisanje tablica i pogleda. Storyboard jedan od SupaBase korisnih alata, vizualno prikazuje tablice što značajno pomaže razumijevanju strukture i odnosa podataka. SupaBase asistent je također koristan alat jer olakšava upravljanje bazama podataka i interakciju s njima. Tijekom razvoja, korištenje vizualnih editora i AI funkcionalnosti

pomoglo je bržem savladavanju izazova te osiguralo visoku kvalitetu krajnje aplikacije.

Rad na ovom projektu potvrdio je da su FlutterFlow i SupaBase izvanredan izbor za razvoj aplikacija koje zahtijevaju visoku funkcionalnost i intuitivno sučelje. Ove tehnologije omogućuju razvoj aplikacija koje mogu zadovoljiti potrebe suvremenih korisnika, pružajući im pouzdane i učinkovite alate za obavljanje svakodnevnih zadataka. Stoga se može zaključiti da su FlutterFlow i SupaBase ne samo praktične, već i vrlo moćne platforme za razvoj modernih, višeplatformskih aplikacija.

LITERATURA

1. AAI@EduHr. *Riječnik pojmova*. <https://www.aaiedu.hr/o-sustavu/sto-je-aaieduhr/rjecnik-pojmova> (pristupljeno 28. 01. 2024.).
2. Agency, LowCode. »FlutterFlow vs Flutter.«
3. AltexSoft. *Pros and Cons of Flutter App Development*. <https://www.altexsoft.com/blog/pros-and-cons-of-flutter-app-development/> (pristupljeno 19. 04. 2024.).
4. Benchmark, DeltaBlue. *Studying Dart Performance with the DeltaBlue Benchmark*. <https://www.nequalsonelifestyle.com/2021/12/12/dart-benchmarking-with-delta-blue/> (pristupljeno 11. 06. 2024.).
5. Blog, Supabase. *Flutter Tutorial: Building a Chat App*. <https://supabase.com/blog/flutter-tutorial-building-a-chat-app> (pristupa 03. 03. 2024.).
6. Codemagic. *Dart vs JavaScript*. <https://blog.codemagic.io/dart-vs-javascript/> (pristupljeno 22. 05. 2024.).
7. docs, FlutterFlow [https://1863580799-files.gitbook.io/~files/v0/b/gitbook-x-prod.appspot.com/o/spaces%2F-MhFNOxEwcl8ED58MUC_%2Fuploads%2Fx6eKBMDDA8stepFdChU6j%2Fstate%20management%20demo%20\(2\).png?alt=media&token=bcef7e0d-1634-436c-9c6f-197eb566a930](https://1863580799-files.gitbook.io/~files/v0/b/gitbook-x-prod.appspot.com/o/spaces%2F-MhFNOxEwcl8ED58MUC_%2Fuploads%2Fx6eKBMDDA8stepFdChU6j%2Fstate%20management%20demo%20(2).png?alt=media&token=bcef7e0d-1634-436c-9c6f-197eb566a930) (pristupljeno 23. 05. 2024.).
8. Flutter. »Flutter README.« [github.com](https://github.com/flutter/flutter?tab=readme-ov-file). <https://github.com/flutter/flutter?tab=readme-ov-file> (pristupa 05. 01. 2023.).
9. FlutterFlow. *FlutterFlow Documentation*. <https://docs.flutterflow.io/> (pristupljeno 23. 11. 2023.).
10. GeeksforGeeks. *Flutter Projects with Source Code*. <https://www.geeksforgeeks.org/flutter-projects-with-source-code/> (pristupljeno 17. 04. 2024.).
11. —. *Flutter Stateful vs Stateless Widgets*. <https://www.geeksforgeeks.org/flutter-stateful-vs-stateless-widgets/> (pristupljeno 23. 05. 2024.).
12. Google. *Dart*. <https://dart.dev/> (pristupljeno 06. 12. 2023.).
13. —. *Dart Guides*. <https://dart.dev/guides> (pristupljeno 05. 12. 2023.).
14. —. *Flutter Documentation*. <https://flutter.dev/> (pristupljeno 07. 12. 2023.).
15. —. *Flutter Documentation*. <https://docs.flutter.dev/> (pristupljeno 24. 11. 2023.).
16. Hackr.io. *React Native vs Flutter*. <https://hackr.io/blog/react-native-vs-flutter> (pristupljeno 15. 02. 2024.).

17. IO, NonStop. *What is SupaBase*. . <https://blog.nonstopio.com/what-is-supabase-15ac54bfa849> (pristupljeno 11. 02. 2024.).
18. Kinsta. *What is PostgreSQL?* . <https://kinsta.com/knowledgebase/what-is-postgresql/> (pristupljeno 04. 02. 2024.).
19. Owerflow, Stack. *Tag trends*. . <https://insights.stackoverflow.com/trends?tags=flutter%2Creact-native%2Cionic-framework%2Cxamarin> (pristupa 23. 05. 2024.).
20. Palm, Zee. *FlutterFlow: What It Brings to the Table*. . <https://medium.com/zee-palm/flutterflow-what-it-brings-to-the-table-13a61a1fe7d9> (pristupljeno 10. 02. 2024.).
21. PostgreSQL. *PostgreSQL Global Development Group*. . <https://www.postgresql.org/> (pristupljeno 07. 01. 2024.).
22. *React Native Documentation*. . <https://reactnative.dev/> (pristupa 08. 01. 2024.).
23. Schwarzmuller, Maximilian. *Flutter & Dart - The Complete Guide*. . <https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/> (pristupljeno 26. 01. 2024.).
24. *SupaBase GitHub Repository*. . <https://github.com/supabase/supabase> (pristupa 18. 01. 2024.).
25. SupaBase. *Integrations: FlutterFlow*. . <https://supabase.com/partners/integrations/flutterflow> (pristupa 14. 02. 2024.).
26. —. *SupaBase Documentation*. . <https://supabase.com/docs> (pristupa 23. 11. 2023.).
27. Techkno, Third Rock. *FlutterFlow vs Flutter*. . <https://www.thirdrocktechkno.com/blog/flutterflow-vs-flutter/> (pristupljeno 21. 05. 2024.).
28. Trio. *Flutter vs React Native*. . <https://trio.dev/flutter-vs-react-native/> (pristupa 15. 02. 2024.).
29. Wikipedia. *Dart (programming language)*. . [https://en.wikipedia.org/wiki/Dart_\(programming_language\)](https://en.wikipedia.org/wiki/Dart_(programming_language)) (pristupljeno 05. 12. 2023.).

SAŽETAK

Ovaj završni rad predstavlja tehnologije FlutterFlow i SupaBase kroz primjere izrade aplikacije za cjenike. FlutterFlow je alat koji omogućuje brzi i vizualni pristup dizajniranja aplikacija.

SupaBase s druge strane pruža backend rješenja uključujući pohranu podataka, autentikaciju i upravljanje datotekama. Kroz ovaj rad prikazan je detaljan proces razvoja aplikacije, počevši od dizajniranja korisničkog sučelja u FlutterFlowu do implementacije backend funkcionalnosti s pomoću Supabasea.

Aplikacija za cjenike razvijena u ovom projektu demonstrira kako kombinacija ovih tehnologija može omogućiti brzi razvoj, visoku produktivnost i kvalitetu aplikacije. Na kraju rad prikazuje da su FlutterFlow i SupaBase moćni alati za razvoj modernih, višeplatformskih aplikacija koje zadovoljavaju potrebe suvremenih korisnika.

Ključne riječi: FlutterFlow, Flutter, SupaBase, aplikacija za cjenike, višeplatformske aplikacije

SUMMARY

This paper presents the technologies FlutterFlow and SupaBase through examples of creating a price list application. FlutterFlow is a tool that provide a fast and visual approach to designing applications.

SupaBase, on the other hand, provides backend solutions including data storage , authentication, and file management. This work demonstrates how the combination of these technologies can enable rapid development, high productivity, and quality of the application. In conclusion, the work shows that FlutterFlow and SupaBase are powerful tools for developing modern, cross-platform applications that meet the needs of contemporary users.

Keywords: FlutterFlow, Flutter, SupaBase, price list application, cross-platform applications