

Eventim organizacija događaja i prodaja ulaznica - poslovna aplikacija nad relacijskom bazom podataka (Oracle APEX)

Cetina, Erik

Undergraduate thesis / Završni rad

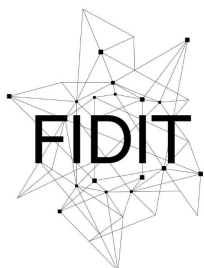
2022

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:321534>

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

Download date / Datum preuzimanja: **2024-05-19**



Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija

Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Sveučilišni preddiplomski studij informatike

Erik Cetina

Eventim organizacija događaja i
prodaja ulaznica - poslovna aplikacija
nad relacijskom bazom podataka
(Oracle APEX)

Završni rad

Mentor: doc. dr. sc. Danijela Jakšić

Rijeka, 2022.

Rijeka, 14.6.2022.

Zadatak za završni rad

Pristupnik: Erik Cetina


Naziv završnog rada: Eventim organizacija događaja i prodaja ulaznica - poslovna aplikacija nad relacijskom bazom podataka (Oracle APEX)

Naziv završnog rada na eng. jeziku: Eventim - business application with relational database (Oracle APEX)

Sadržaj zadatka: Baza podataka (BP) predstavlja kolekciju podataka, ograničenja i operacija koji reprezentiraju neke aspekte realnoga svijeta. Dakle, BP je model neke aplikacijske domene. Cilj rada je izgraditi aplikaciju za potrebe procesa Eventima. Aplikacija će biti izgrađena nad relacijskom BP pomoću alata Oracle APEX. Na početku će biti specificirani zahtjevi i potrebne značajke aplikacije. Sljedeći korak bit će dizajn baze podataka, odnosno izrada modela entiteta i veze. Transformacijom modela entiteta i veze konstruirat će se relacijski model. Na temelju relacijskog modela izgradit će se baza podataka. Nakon toga kreće izrada same aplikacije. Aplikacija će raditi nad relacijskom BP u navedenoj aplikacijskoj domeni (Eventim).

Mentor

doc. dr. sc. Danijela Jakšić



Voditelj za završne radove

doc. dr. sc. Miran Pobar



Zadatak preuzet: 14.6.2022.



(potpis pristupnika)

Table of Contents

1. Uvod.....	1
2. Modeliranje podataka	2
2.1. Dijagram entitet-veza	2
2.2. Relacijski model	6
3. Proces izrade	7
3.1. Zahtjev za radnim prostorom.....	7
3.2. Kreiranje nove aplikacije	8
3.3. Kreiranje tablica i veza	10
3.4. Kreiranje sekvenci za primarne ključeve	13
3.5. Kreiranje početne stranice	14
3.6. Dinamičke akcije	18
3.7. Prosljeđivanje vrijednosti na drugu stranicu.....	19
3.8. Forma.....	20
3.9. Interaktivni izvještaj	22
3.10. Uklanjanje prijave.....	23
3.11. Izrada administrativne sheme	25
4. Gotova aplikacija	26
5. Zaključak	28
6. Literatura	29
7. Popis priloga	30

Sažetak:

U ovome radu opisan je i prikazan proces izrade poslovne aplikacije nad relacijskom bazom podataka na temu Eventim, sustava za prodaju karata. Na početku izrade projekta bilo je potrebno izraditi dijagram entiteti/veze kako bi na temelju njega mogli izraditi bazu podataka. Za izradu aplikacije i same baze podataka korištena je Oracle APEX platforma. U radu je detaljno opisan postupak izrade aplikacije s popratnim slikama zaslona.

Ključne riječi: Oracle, Apex, baza podataka, web aplikacija, izvještaj, model, korisničko sučelje

1. Uvod

Prije osam godina, prvi je put uveden pojam low code ili niski kod za klasifikaciju razvojnih platformi koje su usmjerene na jednostavnost razvoja i lakoću korištenja. Te platforme su omogućile razvojnim programerima i korisnicima svih razina vještina kodiranje aplikacije, bez izravne potrebe za poznavanjem kodiranja (Kissflow, 2022.). Jedna od tih platformi je i Oracle APEX. Oracle Application Express ili samo Oracle APEX je razvojna platforma niskog koda koja se koristi za razvoj web aplikacija nad relacijskom bazom podataka (Oracle APEX, 2022.). APEX sadrži integrirano razvojno okruženje (IDE) koje se nalazi na web-u te iz tog razloga nema potrebe za instalacijom bilo kakvih alata ili software-a, što dodatno olakšava njegovo korištenje.

Relacijska baza podataka organizira podatke u retke i stupce te oni skupa čine tablicu. Obično su podaci strukturirani u tablicama koje se mogu povezati putem primarnog ili vanjskog ključa. Primarni i vanjski ključevi prikazuju različite odnose koji postoje između svih tablica unutar jedne baze (IBM dokumentacija, 2022.). Ti se odnosi obično ilustriraju kroz različite vrste podatkovnih modela, primjerice dijagramom entitet-veza te relacijskim modelom koji su opisani u nastavku rada. Također, relacijske baze podataka su izgrađene kako bi rarujele Structured Query Language (SQL), odnosno programski jezik koji se koristi za pohranu, manipulaciju i dohvat podataka.

U sklopu ovog rada izgrađena je aplikacija kojoj je glavna svrha prodaja ulaznica za razne događaje. Unutar same aplikacije korisnik može pretraživati ulaznice po raznim kriterijima. Nakon što korisnik pronađe željenu ulaznicu, odabere događaj i količinu ulaznica te nastavlja dalje s procesom kupovine. Administratorova uloga je dodavanje i brisanje događaja, ulaznica, lokacija i slično.

Na početku rada opisuje se modeliranje podataka, odnosno dijagram entitet-veza i relacijski model podataka, a zatim slijedi detaljan proces izrade aplikacije uz popratne slike zaslona. U konačnici se nalazi prikaz gotove aplikacije, zaključak, literatura i popis priloga.

2. Modeliranje podataka

Kako bi izbjegli nedoumice i potencijalne probleme tijekom razvoja baze podataka, koristimo se modeliranjem podataka, odnosno izgradnjom raznih dijagrama i modela koji nam omogućuju jasan pregled cijele strukture naše baze podataka. U svrhu izgradnje ove baze podataka i aplikacije koristili smo se dijagramom entitet-veza i relacijskim modelom baza podataka.

2.1. Dijagram entitet-veza

Modeliranje entitetima i vezama je grafički pristup dizajnu baze podataka. To je model podataka visoke razine koji definira elemente podataka i njihov odnos.






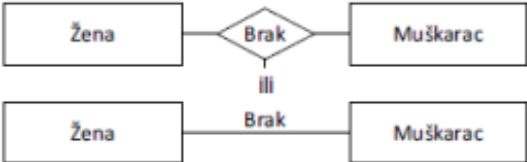




Entitet je stvar ili objekt u stvarnom svijetu koji se razlikuje od vanjske okoline. Na primjer, svaki zaposlenik jedne organizacije je zaseban entitet. Jaki tip entiteta ima svoj primarni ključ i neovisan je o drugim tipovima entiteta. Entitet označavamo sa pravokutnikom. Slabi tip entiteta je ovisan o drugom tipu objekta. U ovom primjeru aplikacije prikazano je kako je *stavka računa* slabi tip entiteta te je ovisna o jakom tipu entiteta, tj. *računu*.

Veze predstavljaju odnose koje postoje između entiteta. Brojnost tipa veze je broj koji govori koliko entiteta sudjeluje u vezi između ta dva entiteta. Grafički prikaz za vezu je ravna crta ili romb. Brojnost se označuje s dva uređena para koji sadrže gornju i donju granicu brojnosti te taj uređeni par smještamo na željenu vezu.

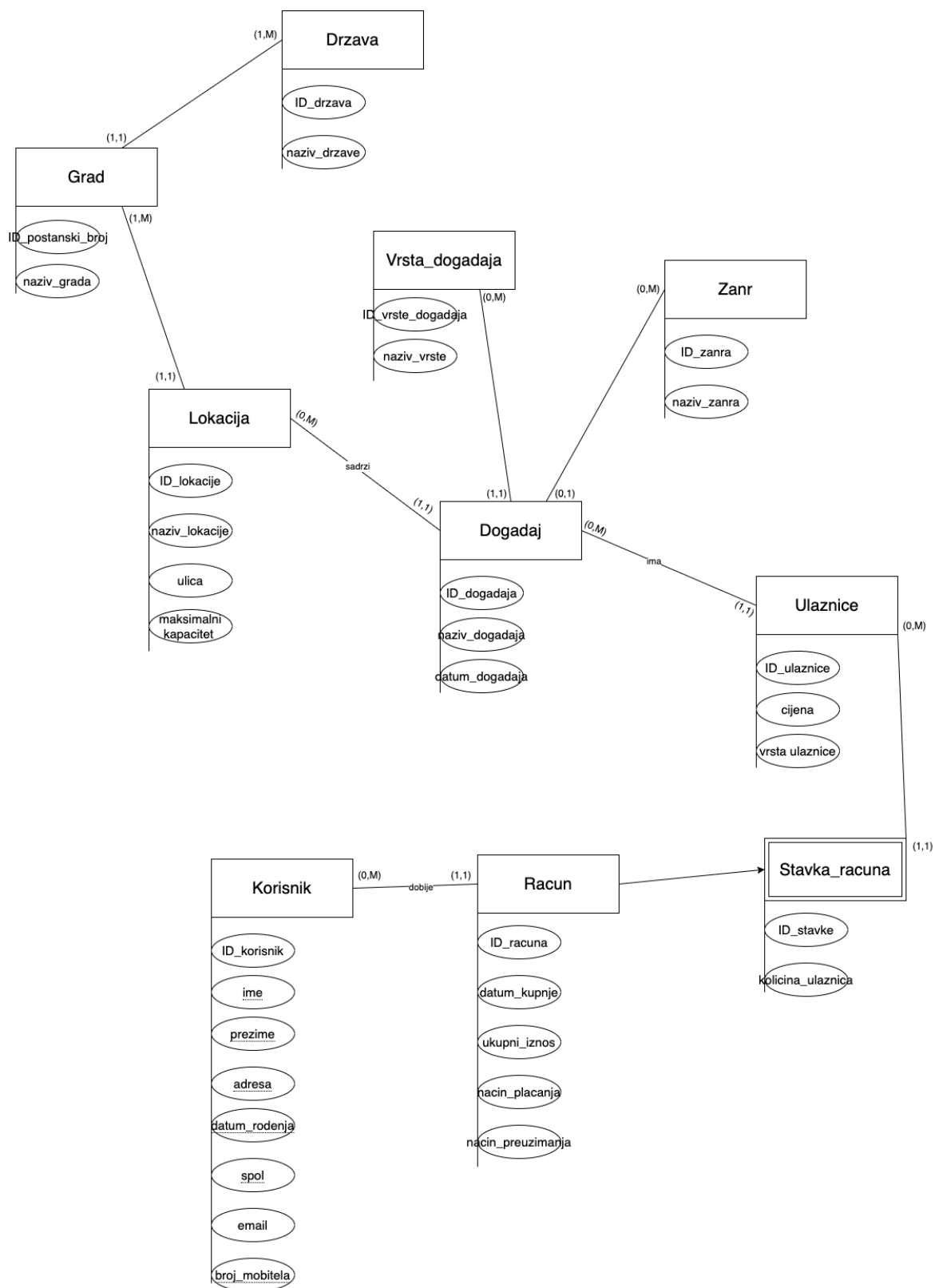
Atribut entiteta je svojstvo nekog entiteta, odnosno oznaka predmeta ili pojma. Označavamo ga s krugom.

Agregacija ili dekompozicija je tip objekta u kojoj se veza između dva i više tipova objekta tretira kao novi tip objekta. Označavamo ga s romбом koji se nalazi unutar pravokutnika.

U nastavku možemo vidjeti prikaz grafičkih oblika koji se koriste za izradu dijagrama entitet-veza.

KONCEPT	SIMBOL	PRIMJER
TIP ENTITETA		
SLAB TIP ENTITETA		
TIP VEZE		
ATRIBUT		
AGREGACIJA		

Slika 1. Prikaz grafičkih oblika modela entitet-veza, Pavlič 2011.



Slika 2. Prikaz dijagrama entitet-veza korištenog za razvoj Eventim aplikacije

Dijagram se sastoji od 10 objekata. Jaki tipovi entiteta su: DRŽAVA, GRAD, LOKACIJA, VRSTA DOGAĐAJA, ŽANR, DOGAĐAJ, ULAZNICE, RAČUN, KORISNIK. Slabi tip entiteta je STAVKA RAČUNA.

Entitet DRŽAVA je povezana sa dva atributa: *ID_drzava* koja ujedno čini i primarni ključ, te *naziv_drzave*. Na dijagramu možemo vidjeti kako je povezana sa entitetom GRAD preko veze (1,M) na (1,1), što znači da u jednoj državi možemo imati više gradova, ali jedan grad može biti u samo jednoj državi.

Entitet GRAD također ima dva atributa. Iz razloga što svaki grad ima svoj poštanski broj, odnosno svaki poštanski broj je jedinstven, upravo smo *ID_postanski_broj* postavili kao primarni ključ za taj entitet. Drugi atribut je *naziv_grada*. Entitet GRAD povezan je sa entitetom LOKACIJA preko veze (1,M) na (1,1), odnosno jedan grad može sadržavati više lokacija ali jedna lokacija može biti samo u jednom gradu.

Entitet LOKACIJA ima 4 atributa: *ID_lokacija*, *naziv_lokacija*, *ulica* te *maksimalni kapacitet*. *ID_lokacija* smo postavili kao primarni ključ. LOKACIJA je povezana sa entitetom DOGAĐAJ preko veze (0,M)(1,1). Jedna lokacija može imati više događaja ali jedan događaj može biti samo na jednoj lokaciji.

Entitet DOGAĐAJ ima 3 atributa: *ID_dogadaj*, *naziv_dogadaja* te *datum_dogadaja*. *ID_dogadaj* je primarni ključ. Povezan je sa entitetima VRSTA DOGAĐAJA vezom (1,1)(0,M), ŽANR vezom (0,1)(0,M) te ULAZNICE vezom (0,M)(1,1).

Entitet VRSTA DOGAĐAJA ima 2 atributa: *ID_vrsta_dogadaja* i *naziv_vrste*. *ID_dogadaja* je primarni ključ te je entitet povezan sa entitetom DOGAĐAJ prijašnje spomenutim vezama.

Entitet ŽANR isto kao i VRSTA DOGAĐAJA ima 2 atributa: *ID_zanr* te *naziv_zanra*. *ID_zanr* nam igra ulogu primarnog ključa. Entitet je povezan sa DOGAĐAJ preko prijašnje spomenutih veza.

Entitet ULAZNICE sadrže 3 atributa: *ID_ulaznica*, *cijena* te *vrsta_ulaznice*. *ID_ulaznica* nam je primarni ključ. Entitet ULAZNICE povezan je sa slabim tipom entiteta STAVKA RAČUNA preko veze (0,M)(1,1).

Slabi tip entiteta sadrži 2 atributa: *ID_stavka* te *kolicina_ulaznica*. Primarni ključ je *ID_stavka*. Entitet RAČUN je povezan na entitet STAVKA RAČUNA posebnim tipom veze između jakog i slabog tipa entiteta.

Entitet RAČUN sadrži 5 atributa: *ID_racun*, to nam je ujedno i primarni ključ, *datum_kupnje*, *ukupni_iznos*, *nacin_placanja* te *nacin_preuzimanja*. Povezan je sa entitetom KORISNIK preko veze (1,1)(0,M).

Entitet KORISNIK sastoji se od 8 atributa: *ID_korisnik* (primarni ključ), *ime*, *prezime*, *adresa*, *datum_rodenja*, *spol*, *email* te *broj_mobitela*.

2.2. Relacijski model

Na temelju dijagrama entitet-veza nacrtan je relacijski model. Za razliku od dijagrama entitet-veza, relacijski model prikazuje i vanjske ključeve. Primarni ključevi su podcrtani punom crtom, dok su vanjski ključevi podcrtani isprekidanom crtom.

DRZAVA (ID_drzava, naziv_drzave)

GRAD (ID_postanski_broj, naziv_grada, FK_ID_drzava)

LOKACIJA (ID_lokacija, naziv_lokacije, ulica, maksimalni_kapacitet, FK_ID_postanski_broj)

DOGADAJ (ID_dogadaj, naziv_dogadaja, datum_dogadaja, FK_ID_lokacija, FK_ID_vrsta_dogadaja, FK_ID_zanr)

VRSTA_DOGADAJA (ID_vrsta_dogadaja, naziv_vrste)

ZANR (ID_zanr, naziv_zanra)

ULAZNICE (ID_ulaznica, cijena, vrsta_ulaznice, FK_ID_dogadaj)

STAVKA_RACUNA (ID_stavka, kolicina_ulaznica, FK_ID_ulaznica)

RACUN (ID_racun, datum_kupnje, ukupni_iznos, nacin_placanja, nacin_preuzimanja, FK_ID_stavka, FK_ID_korisnik)

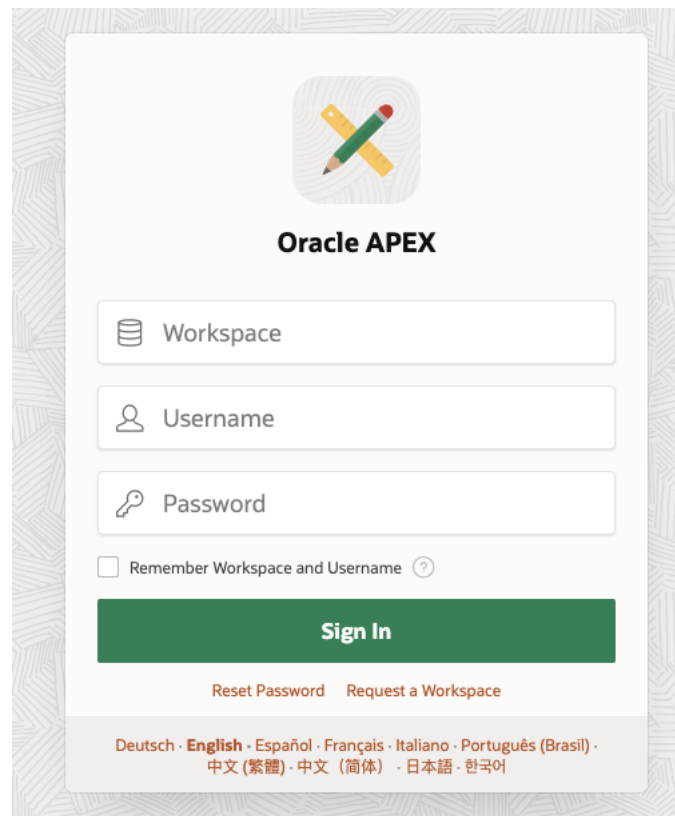
KORISNIK (ID_korisnik, ime, prezime, adresa, datum_rodenja, spol, email, broj_mobitela)

3. Proces izrade

U nastavku rada slijedi prikaz izgradnje web aplikacije Eventim u sustavu Oracle APEX s popratnim slikama zaslona.

3.1. Zahtjev za radnim prostorom

Ukoliko želimo koristiti APEX, prvi korak je zatražiti pristup radnom prostoru (*engl. Workspace*). To učinimo tako da kliknemo na *Request a Workspace*(Slika 3.).



Oracle APEX

Workspace

Username

Password

☐ Remember Workspace and Username ?

Sign In

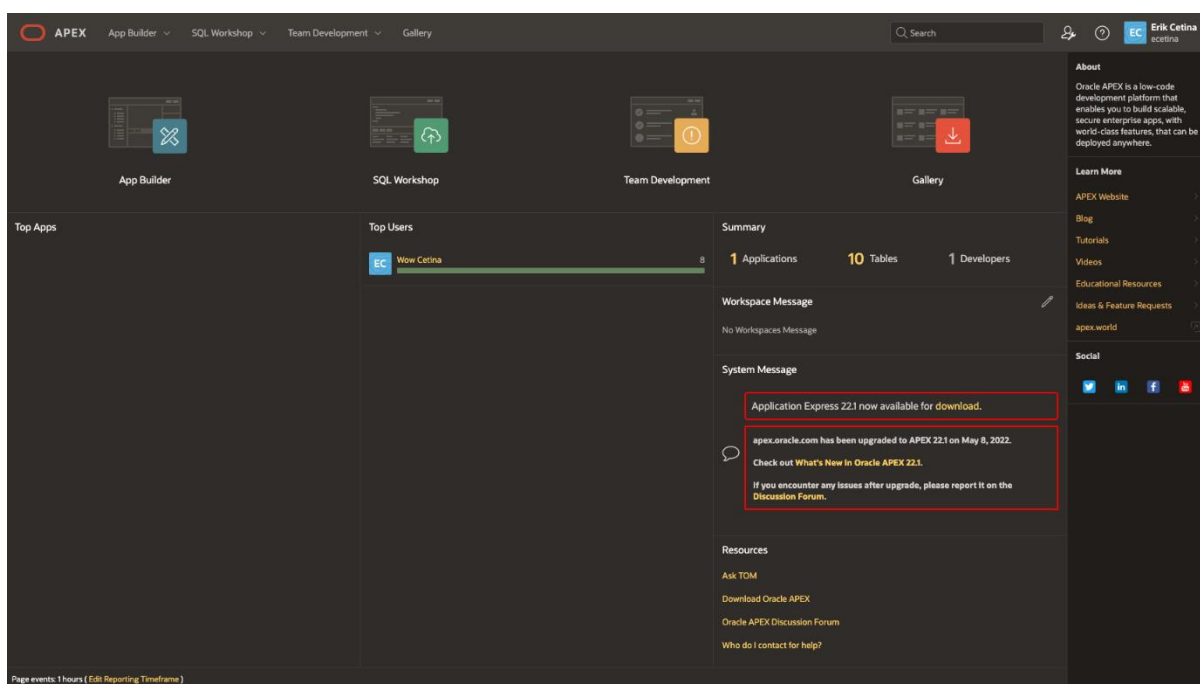
[Reset Password](#) [Request a Workspace](#)

Deutsch · **English** · Español · Français · Italiano · Português (Brasil) ·
中文 (繁體) · 中文 (简体) · 日本語 · 한국어

Slika 3. APEX zahtjev za radnim prostorom

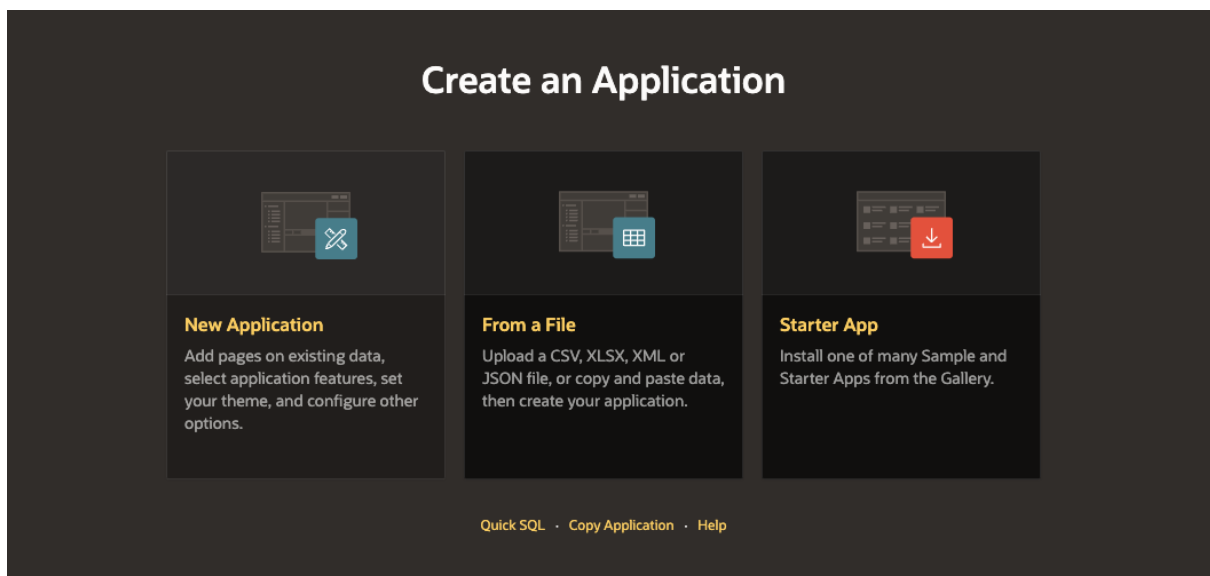
Nakon toga, zahtjev popunimo s potrebnim podacima te putem elektroničke pošte dobijemo potvrdu nakon koje smo spremni početi s izradom aplikacije.

3.2. Kreiranje nove aplikacije



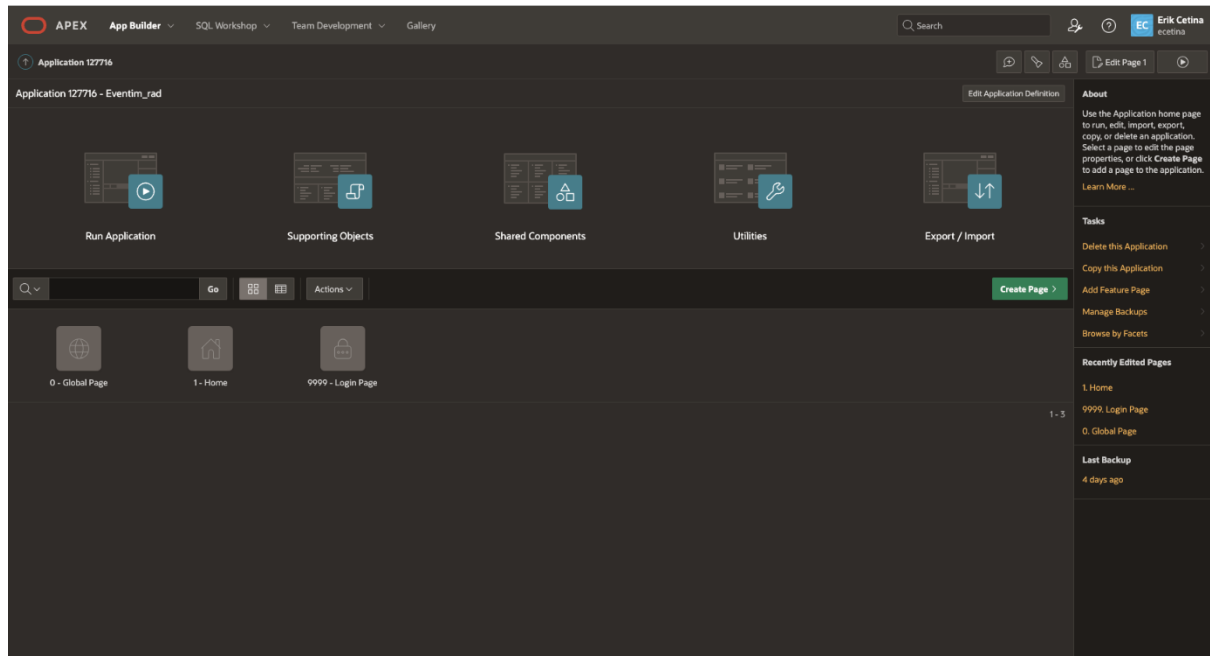
Slika 4. Početna stranica Oracle APEX-a

Nakon prijave vidimo početnu stranicu našeg sustava (Slika 4.). Želimo napraviti novu aplikaciju, stoga kliknemo na *App Builder* te na *Create*. To nas dovede do *Create an Application* prozora gdje možemo odabrati kako želimo započeti s izradom. Budući da na raspolaganju nemamo nikakvu datoteku kojom bi započeli našu aplikaciju, odabiremo *New Application* (Slika 5.).



Slika 5. Nova aplikacija

Nakon što pritisnemo *New Application* upisujemo ime aplikacije. Ukoliko znamo unaprijed da nam trebaju neke stranice, to možemo učiniti odmah. U ovoj aplikaciji su sve stranice dodane naknadno. Dakle, kliknemo *Create Application* i APEX kreira našu aplikaciju te nas odvede na glavno sučelje (Slika 6.) preko kojeg dalje pristupamo svim potrebnim alatima.

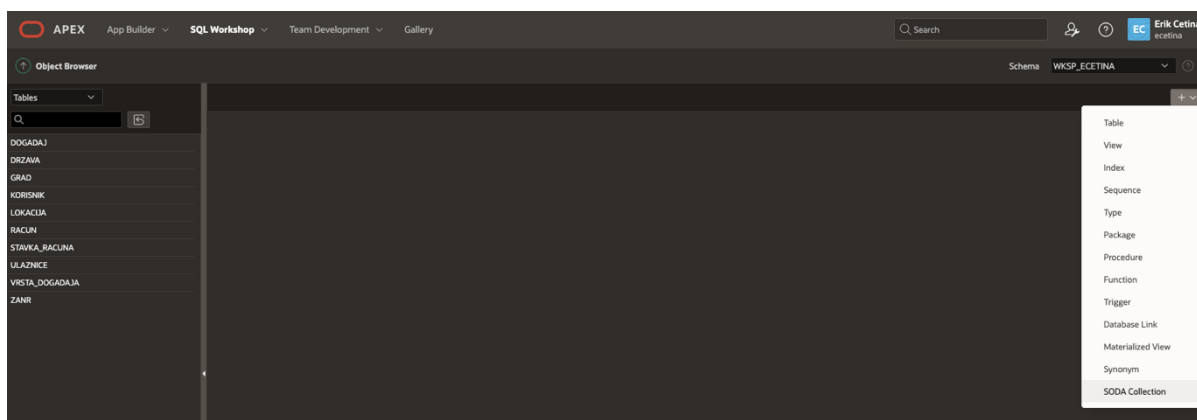


Slika 6. Glavno sučelje aplikacije

3.3. Kreiranje tablica i veza

Prije kretanja u proces izrade stranice, za pronalaženje potrebnu logike, prvo moramo napraviti bazu podataka.

Za izradu nove tablice potrebno je kliknuti na *SQL Workshop* na gornjoj navigacijskoj traci (Slika 7.) te nakon toga biramo hoćemo li kliknuti na *Object browser* te napraviti tablice putem prilagođenog sučelja ili ćemo odabrati *SQL commands* te tablice napraviti upisivanjem SQL koda. U ovom radu naglasak je na APEX platformi tako da su tablice kreirane preko APEX-ovog sučelja. Dakle, kliknemo na *Object browser* te nakon toga na malu ikonicu u obliku znaka plus. Ponudi nam se padajući izbornik sa popisom vrijednosti koje možemo odabrati.



Slika 7. Nova tablica

Odaberemo *Table*, upišemo ime tablice i imena i tipove stupaca koje želimo u našoj novoj tablici (Slika 8.). Prva tablica koju smo napravili je tablica *Korisnik* (Slika 9.). Kao što smo već prije napomenuli, tablica *Korisnik* sadrži 8 atributa, odnosno stupaca. Dakle, upišemo prvi stupac, *ID_korisnik*, tip stavimo u *Number* jer želimo da naš primarni ključ bude neka brojeva jedinstvena sekvenca pomoću koje ćemo kasnije moći prepoznati pojedine retke. Namjestimo preciznost na 10 što znači da *ID_korisnik* neće moći imati više od 10 znamenki, te skalu na 0, ne želimo da naš ID bude decimalni broj. Obavezno označimo *Not Null* opciju jer svaki novi redak mora imati svoj jedinstveni broj. Dodajemo sljedeći stupac, *Ime*, te njegov tip stavljamo na *Varchar2*, skalu postavljamo na 20 te također označujemo *Not Null*. Želimo da kod svakog novog korisnika njegovo ime bude prisutno u tablici *Korisnik*. Nastavljamo dalje dok ne unesemo sve stupce za tablicu *Korisnik*.

Create Table

Table Name:

☐ Preserve Case

Column Name	Type	Precision	Scale	Not Null	Identity	Move
ID_korisnik	NUMBER	10	0	<input checked="" type="checkbox"/>	- None -	^ v
ime	VARCHAR2		20	<input checked="" type="checkbox"/>		^ v
prezime	VARCHAR2		30	<input checked="" type="checkbox"/>		^ v
adresa	VARCHAR2		50	<input checked="" type="checkbox"/>		^ v
datum_rodenja	DATE			<input checked="" type="checkbox"/>		^ v
spol	VARCHAR2		1	<input type="checkbox"/>		^ v
email	VARCHAR2		100	<input checked="" type="checkbox"/>		^ v
broj_mobitela	NUMBER	15	0	<input checked="" type="checkbox"/>	- None -	^ v
	- Select Datatype -					^ v

Slika 8. Nova tablica – Korisnik

Kliknemo *Next* te postavimo primarni ključ na stupac ID_korisnik. Ako želimo dodati vanjski ključ, to možemo učiniti odmah ili kasnije. Tablica Korisnik nema vanjskih ključeva pa možemo preskočiti ovaj dio. Na kraju stisnemo *Create Table*, tablica se stvori te je vidljiva pod *Object browser-om*.

KORISNIK

Column Name	Data Type	Nullable	Default	Primary Key
ID_KORISNIK	NUMBER(10,0)	No	-	1
IME	VARCHAR2(30)	No	-	-
PREZIME	VARCHAR2(50)	No	-	-
ADRESA	VARCHAR2(100)	Yes	-	-
DATUM_RODENJA	DATE	No	-	-
SPOL	VARCHAR2(10)	Yes	-	-
EMAIL	VARCHAR2(50)	No	-	-
BROJ_MOBITELA	NUMBER(20,0)	No	-	-

[Download](#) | [Print](#)

Slika 9. Tablica Korisnik

Prva tablica je gotova pa idemo na sljedeću. Postupak je isti, kliknemo na plusić, iz padajuće liste odaberemo tablicu, damo joj ime te dodamo stupce. Ako kod izrade tablice želimo dodati i vanjski ključ, onda u toj tablici moramo napraviti i jedan stupac za vanjski ključ. Kod tablice Račun imamo vanjski ključ koji gleda na tablicu Stavka Računa. Dakle, u tablicu Račun ćemo dodati stupac kojeg ćemo nazvati FK_ID_stavka_racuna. Njega postavljamo kao vanjski ključ koji će gledati na stupac ID_stavka_racuna iz tablice Stavka Računa(Slika 10.). Brisanje postavljamo na *Cascade*, odnosno kod brisanja retka iz tablice Račun želimo da se pobriše i pripadni redak iz tablice Stavka Računa. Ponavljamo ovaj postupak dok ne izgradimo svih 10 tablica iz naše baze podataka.

Add Constraint

Schema:
WKSP_ECETINA

Table:
RACUN

Constraint Name:
RACUN_FK

☐ Preserve Case

Constraint Type:
Foreign Key

☐ Disallow Delete
☒ Cascade Delete
☐ Set Null on Delete

Foreign Key Column(s)

ID_RACUN
DATUM_KUPNJE
UKUPNI_LZNOS
NACIN_PLACANJA
NACIN_PREUZIMANJA
FK_ID_KORISNIK

FK_ID_STAVKA_RACUNA

>>
>
<
<<

Reference Table Name

STAVKA_RACUNA

Reference Table Column(s)

FK_ID_ULAZNICA
KOLICINA_ULAZNICA

ID_STAVKA_RACUNA

>>
>
<
<<

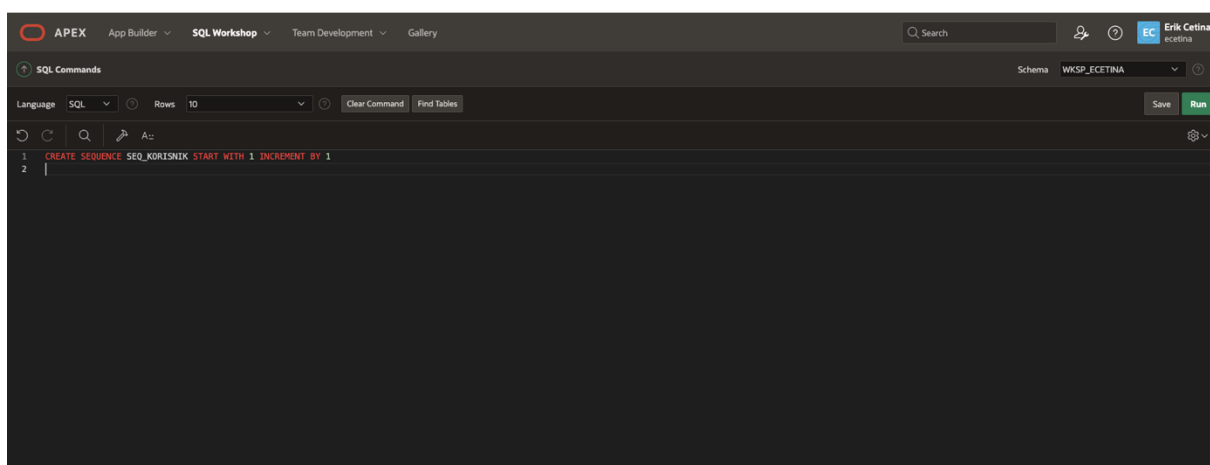
Slika 10. Kreiranje vanjskog ključa

12

3.4. Kreiranje sekvenci za primarne ključeve

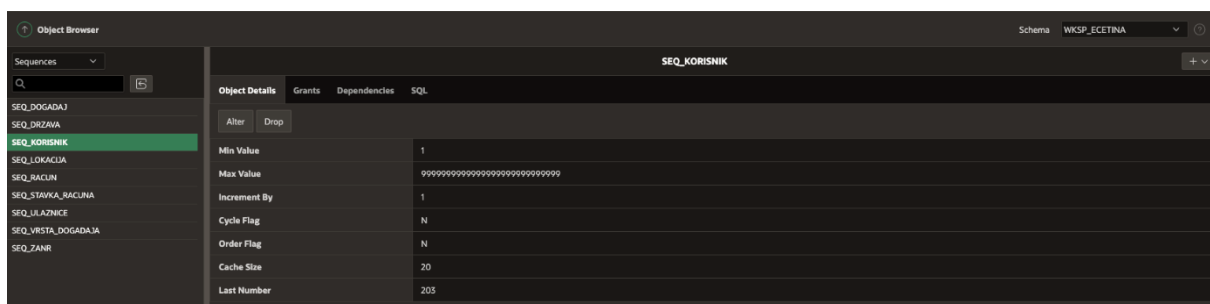
Nakon što napravimo tablice, želimo napraviti samoinkrementirajući broj koji će se inkrementirati na svakom novom retku. Tu sekvencu želimo povezati sa našim primarnim ključem. Primjerice, kod izrade novog korisnika treba se upisati vrijednost primarnog ključa. Kako ta vrijednost mora biti jedinstvena, korisniku bi bilo nespretno upisivati svaki put ručno tu vrijednost iz razloga što ne bi napamet znao koji je broj sljedeći na redu. Iz tog razloga se napravi samoinkrementirajuća vrijednost koja će se na svakom novom korisniku automatski upisati u stupac ID_korisnik. Dakle, prvi korisnik će imati ID = 1, drugi korisnik će imati ID = 2, itd.

Da bi stvorili sekvencu moramo upisati jednu liniju koda u *SQL Commands* konzoli. Uđemo u *SQL Commands* te upišemo komandu prikazanu na slici 11.



Slika 11. Izrada sekvence

Kliknemo na *Run* te pokrenemo kod. Ukoliko je kod točan, konzola izbaciti da je sekvenca uspješno kreirana. Naše sekvence možemo pregledati pod *Object browser*-om tako što umjesto tablica odaberemo prikaz sekvenci (Slika 12.). O tome kako sekvencu spojiti na željeni stupac slijedi u nastavku.

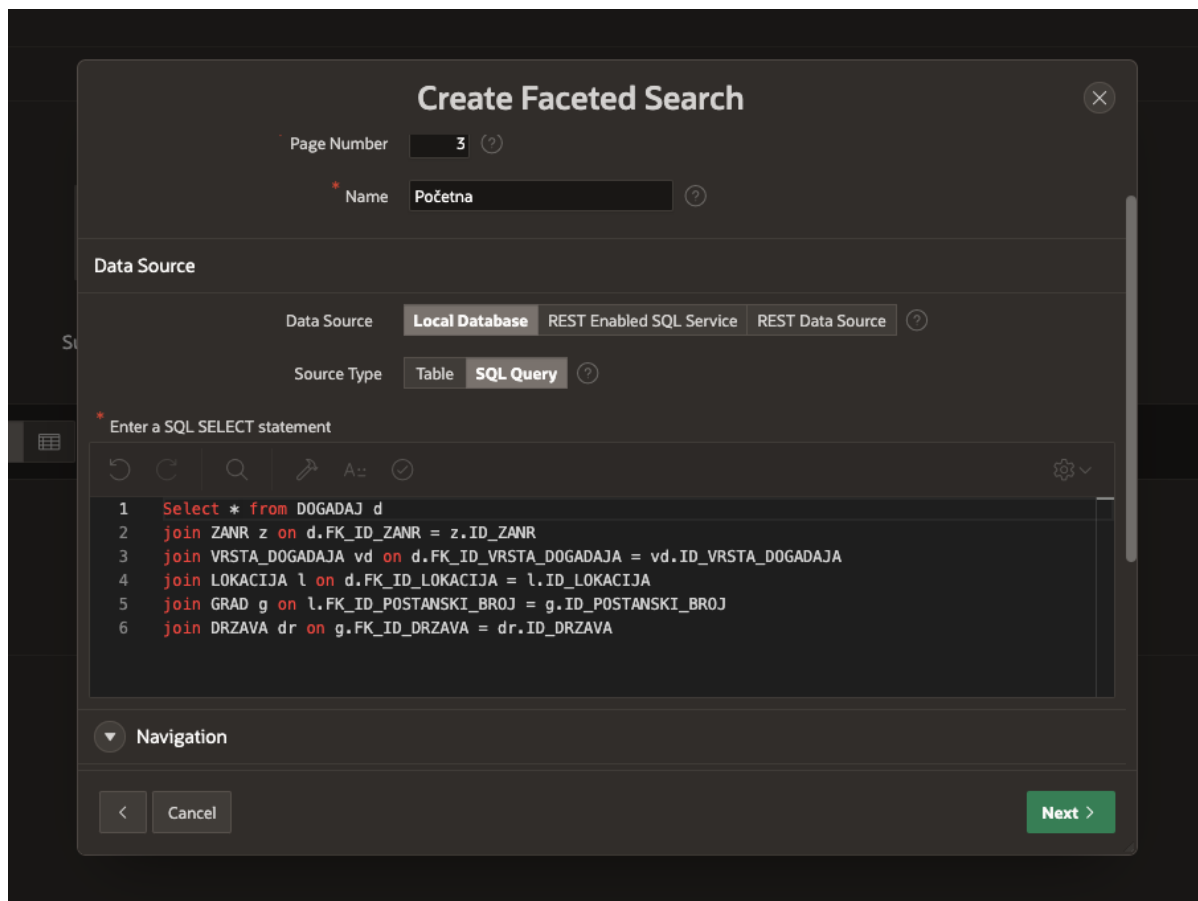


Slika 12. Prikaz sekvenci unutar sučeljah

3.5. Kreiranje početne stranice

Glavna svrha Eventim aplikacije je kupnja ulaznica. Iz tog razloga želimo da korisnik odmah na početnoj stranici može vršiti pretragu nad događajima. Zbog toga, naša će početna stranica biti sastavljena od popisa svih događaja te mnoštvom filtera pomoću kojih korisnik može što brže doći do ciljanog događaja te kupiti karte za isti.

Ulazimo u glavno sučelje te stisnemo na tipku *Create Page*. Ta tipka će se koristiti i ubuduće svaki puta kada želimo napraviti novu stranicu. Klikom na tipku *Create Page*, APEX nam izbaci popis svih mogućih unaprijed izrađenih predložaka. Odaberemo fasetiranu pretragu (*engl. Faceted search*) (Slika 13.). Taj način pretrage nam omogućuje filtriranje podataka po odabranim stupcima. Zatim upisujemo ime stranice, u našem slučaju upisujemo Početna. Odmah nas pita na temelju koje tablice želimo napraviti fasetiranu pretragu. Zbog brzine i praktičnosti, želimo korisniku omogućiti filtriranje na temelju što više stupaca. To znači da bi trebali odabrati više tablica odjednom, no to nije moguće. Postoje dva rješenja; jedna je izrada pogleda (*engl. View*) nad tablicom (u našem slučaju tablicama). Dakle, umjesto same tablice odaberemo pogled. Drugo rješenje je napisati SQL upit, spojiti tablice (*engl. Join*) koje nam trebaju te taj upit iskoristiti kao izvor podataka.



The screenshot shows the 'Create Faceted Search' dialog box in APEX. The 'Page Number' is set to 3. The 'Name' field contains 'Početna'. Under 'Data Source', 'Local Database' is selected. Under 'Source Type', 'SQL Query' is selected. The SQL statement entered is:

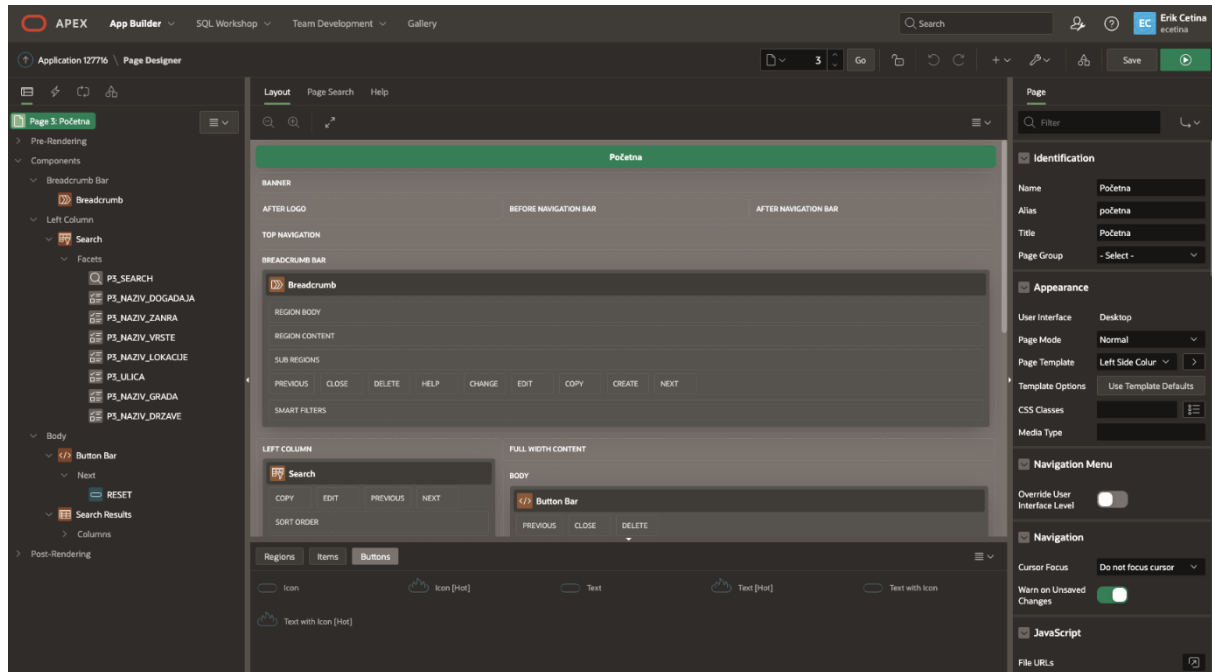
```
1 Select * from DOGAĐAJ d
2 join ZANR z on d.FK_ID_ZANR = z.ID_ZANR
3 join VRSTA_DOGADAJA vd on d.FK_ID_VRSTA_DOGADAJA = vd.ID_VRSTA_DOGADAJA
4 join LOKACIJA l on d.FK_ID_LOKACIJA = l.ID_LOKACIJA
5 join GRAD g on l.FK_ID_POSTANSKI_BROJ = g.ID_POSTANSKI_BROJ
6 join DRZAVA dr on g.FK_ID_DRZAVA = dr.ID_DRZAVA
```

The 'Next' button is highlighted in green.

Slika 13. Fasetirana pretraga

Odlučili smo se za SQL upit u izvoru. Napišemo upit, validiramo ga klikom na kvačicu, te stisnemo *Next*. APEX nam izgradi stranicu i dovede nas u *Page Designer* sučelje (Slika 14.).

Prije nego pokrenemo našu web aplikaciju i vidimo kako izgleda, želimo dodati podatke unutar tablica kako bi nad njima mogli vršiti testiranja. To možemo napraviti u *Object browseru* tako što kliknemo na željenu tablicu i pod *Data* stavkom odaberemo *Insert Row*. Još uvijek nismo povezali sekvence, tako da ID-eve popunjavamo ručno. U svaku tablicu ubacimo podatke po želji te se vratimo na *Page Designer* sučelje.



Slika 14. Page Designer sučelje

Klikom na gornji desni zeleni gumb pokrećemo našu aplikaciju sa stranice na kojoj se trenutno nalazimo.

Eventim_rad

Početna

Search...

Total Row Count 5

Reset

Id Događaj	Naziv Događaja	Datum Događaja	Fk Id Lokacija	Fk Id Vrsta Događaja	Fk Id Zanr	Id Zanr	Naziv Zanra	Id Vrsta Događaja	Naziv Vrste
1	Iron Maiden	1/1/2021	1	1	1	1	Metal	1	Glazba
2	Vatrogasci	5/23/2022	3	1	2	2	Folk	1	Glazba
3	Madonna	3/29/2022	2	1	3	3	Pop	1	Glazba
4	Boris Brejcha	7/29/2022	4	1	4	4	Techno	1	Glazba
6	Real - Barcelona	10/14/2022	5	2	28	28	Nogomet	2	Sport

Release 1.0

App 127716 Page 3 Session Debug Quick Edit Customize

Slika 15. Pokrenuta fasetirana pretraga

Ukoliko kliknemo na neki od ponuđenih filtera, aplikacija odmah filtrira samo željene retke te ostale ukloni sa prikaza. Možemo ukloniti sve filtere ili dodavati još filtera(Slika 15. i Slika 16.).

Eventim_rad

Početna

Search...

Total Row Count 1

Reset

Id Događaj	Naziv Događaja	Datum Događaja	Fk Id Lokacija	Fk Id Vrsta Događaja	Fk Id Zanr	Id Zanr	Naziv Zanra	Id Vrsta Događaja	Naziv Vrste
6	Real - Barcelona	10/14/2022	5	2	28	28	Nogomet	2	Sport

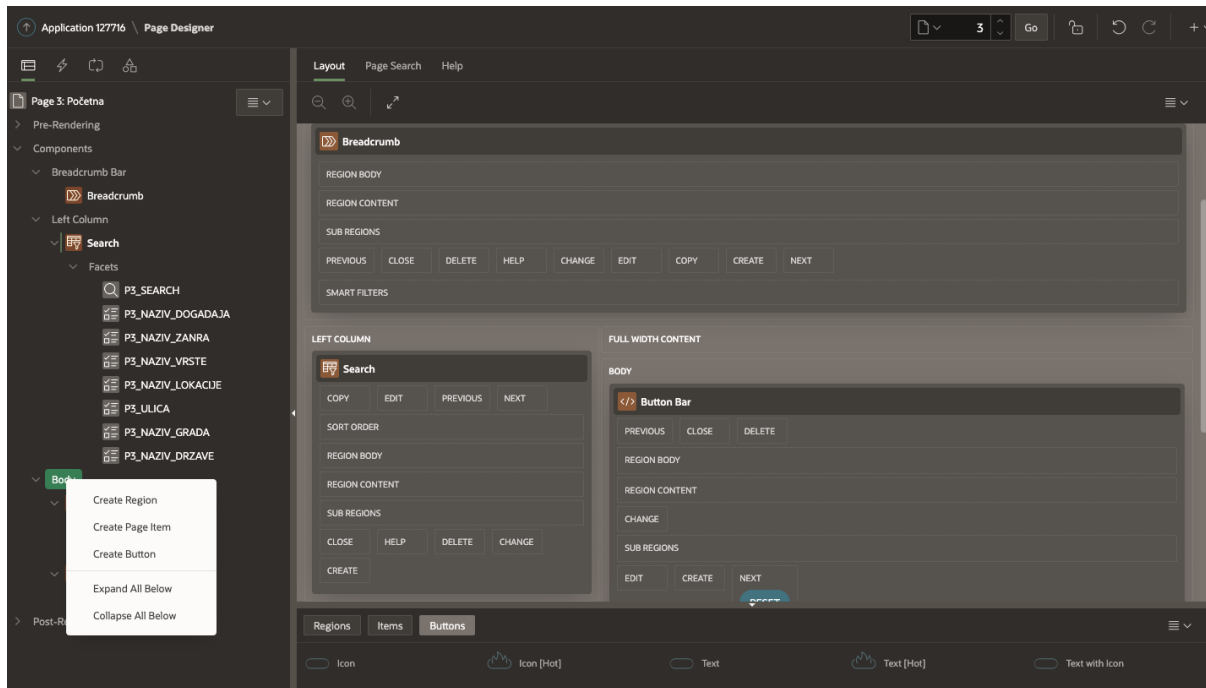
Release 1.0

App 127716 Page 3 Session Debug Quick Edit Customize

Slika 16. Filtrirana fasetirana pretraga

Sada kada imamo fasetiranu pretragu, želimo dodati novi prozor na našu web stranicu koji će služiti da korisnika odvede na željenu stranicu događaja. Na stranici događaja, korisnik će vidjeti sve dostupne karte za taj događaj i njihovu cijenu, zatim će moći odabrati broj ulaznica i vrstu ulaznica koje želi kupiti.

Da bi to ostvarili, lebdimo sa kursorom iznad našeg tijela (*engl. Body*) web stranice te desnim klikom otvaramo mali izbornik gdje možemo odabrati *Create Page Item*(Slika 17.).



Slika 17. Stvori stavku stranice

Tip mora biti *Select List* jer želimo da se korisniku ponudi padajući izbornik sa svim događajima. Za izvor podataka koristimo SQL upit gdje povlačimo sve nazive događaja te tako dobijemo traženu listu.

<div> <div>✓ Izaberi</div> <div> <div>Boris Brejcha</div> <div>Iron Maiden</div> <div>Madonna</div> <div>Real - Barcelona</div> <div>Vatrogasci</div> </div> </div>						Grad	Drzava
Madonna	3/29/2022	Pop	Glazba	HNK Zagreb	30000	Rijeka	Hrvatska
Real - Barcelona	10/14/2022	Nogomet	Sport	Crna Gora Lokacija	1000	Zagreb	Hrvatska
Boris Brejcha	7/29/2022	Techno	Glazba	Beograd stadion	10000	Podgorica	Crna Gora
Vatrogasci	5/23/2022	Folk	Glazba	Sarajevo Arena	100000	Beograd	Srbija
						Sarajevo	Bosna

Slika 18. Select list

U ovom trenutku(Slika 18.), iako kliknemo na neki od događaja, ništa se ne dogodi jer još uvijek nismo definirali što se događa klikom na neke od ovih vrijednosti.

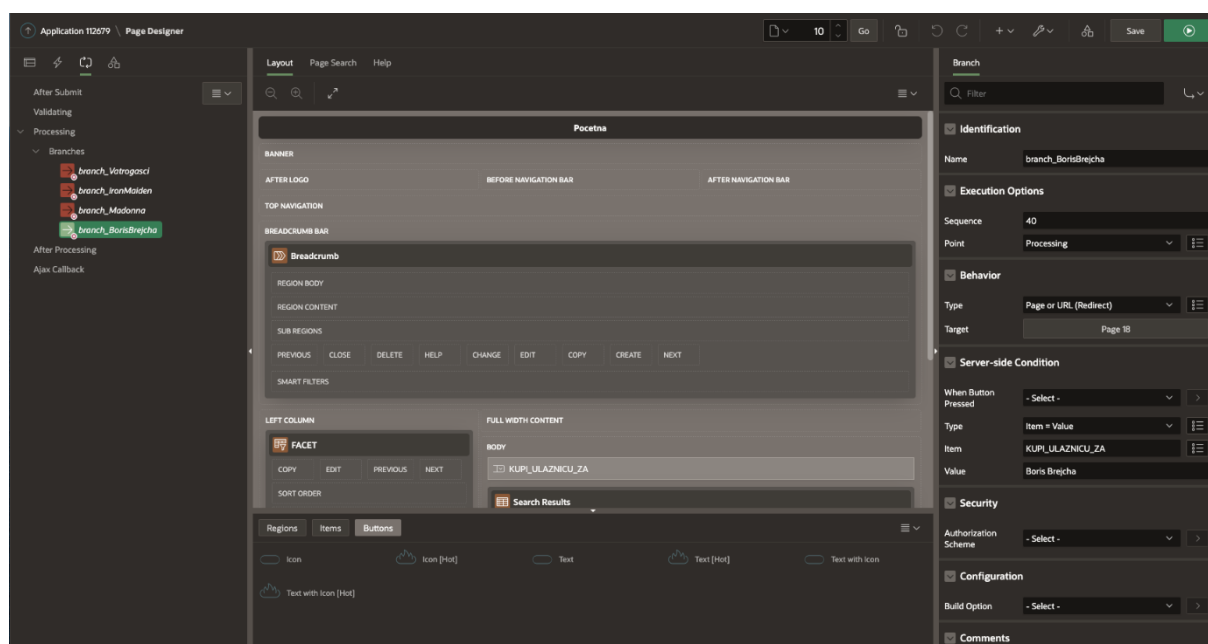
3.6. Dinamičke akcije

U APEX-u, *Dynamic Actions* programerima pružaju način da deklarativno definiraju ponašanje na strani klijenta bez potrebe za poznavanjem JavaScripta. Pomoću jednostavnog čarobnjaka, programeri mogu odabrati stavku stranice, uvjet, unijeti vrijednost i odabrati radnju, npr. prikaži, sakrij, omogući i onemogući (*engl. Show, Hide, Enable, Disable*).

U ovoj aplikaciji dodajemo dinamičku akciju na tu stavku stranice gdje smo napravili listu vrijednosti događaja. Dinamičku akciju postavimo na slučaj promjene (*engl. Change*), stoga, kada promijenimo vrijednost te stavke, aktivira nam se dinamička akcija koja šalje stranicu dalje (*engl. Submit Page*).

Da bi znali koju stavku liste moramo poslati na koju stranicu, radimo granu (*engl. Branch*) koja će preusmjeriti našu aplikaciju na željenu stranicu.

APEX grane omogućuju nam implementaciju i kontrolu tijeka naše APEX aplikacije uz korištenje deklarativnih ili ručno kodiranih opcija. Odjeljak „Grane“ dio je stupca Obrada stranice (*engl. Page Processing*). Napravimo granu koja će preusmjeriti našu aplikaciju na željenu stranicu (Slika 19.).

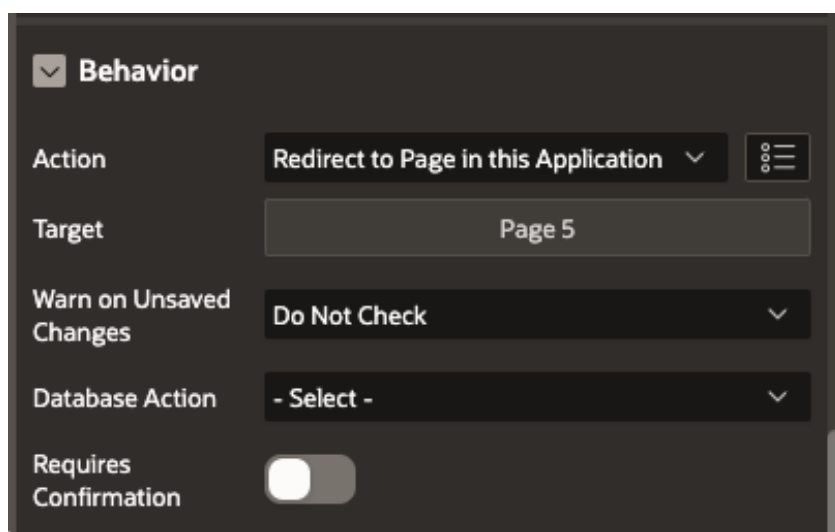


Slika 19. APEX grane (*engl. Branches*)

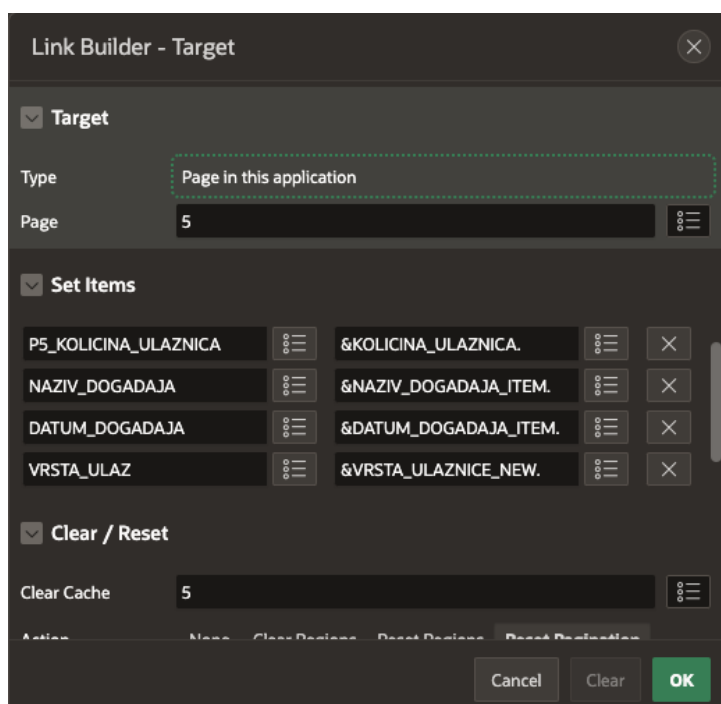
3.7. Proslijeđivanje vrijednosti na drugu stranicu

Trenutno se nalazimo na stranici jednog događaja. Želimo da korisnik odluči o količini i vrsti ulaznica koje želi kupiti. Da bi mu to omogućili, dodajemo dvije nove stavke koje će sadržavati opcije o vrsti i količini ulaznica. Također, želimo da se te dvije vrijednosti proslijede dalje na stranicu izrade računa. Dakle, ukoliko korisnik odabere da želi 5 ulaznica, kod izrade računa polje s količinom ulaznica će već biti popunjeno.

U ovoj aplikaciji napravljena je tipka “Kupi” pa je ideja je da korisnik klikne na tipku koja će ga odvesti na stranicu izrade računa. Na tipku “Kupi”, pod ponašanje (*engl. Behaviour*) (Slika 20.) stavljen je proslijeđivanje na stranicu unutar aplikacije i pod postavi stavke (*engl. Set Items*) (Slika 21.) smo proslijedili željene vrijednosti.



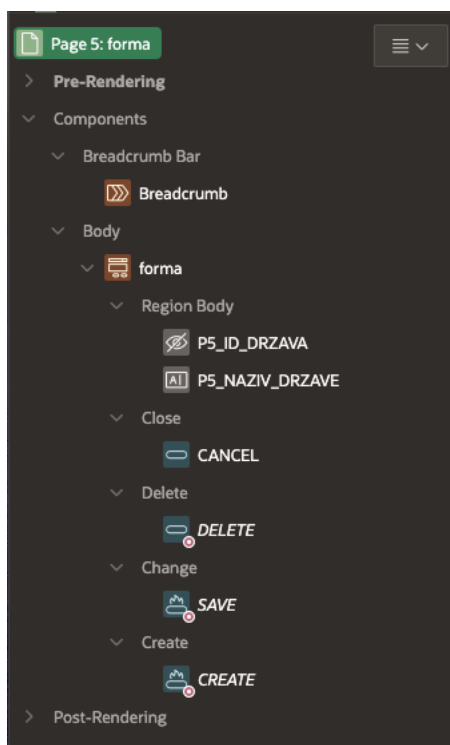
Slika 20. "Ponašanje" stavke



Slika 21. Postavi stavke

3.8. Forma

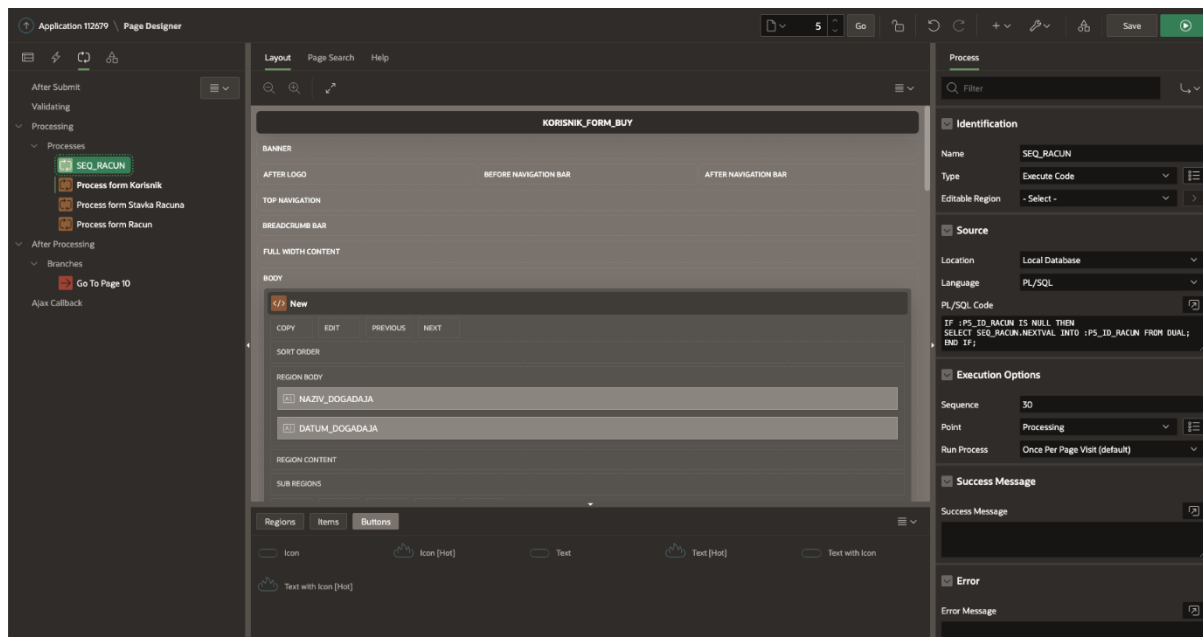
Kada se korisnik odluči za ulaznicu, njenu količinu i vrstu, klikne na tipku “Kupi” te ga ona odvede na novu stranicu. Na novoj stranici želimo da korisnik upiše svoje podatke kako bi se u našoj bazi izgenerirao novi korisnik te kako bi se napravio novi račun. To podrazumijeva unos podataka u našu bazu. U APEX-u imamo već predodređene predloške koji nam znatno olakšavaju posao tako da za napraviti novu formu moramo napraviti novu stranicu. To učinimo tako da odaberemo predložak *Form*, dodjelimo proizvoljno ime stranici, odaberemo tablicu na temelju koje radimo *Form* te kliknemo *Next*.



Slika 22. APEX Forma

Vidimo da na stranici imamo novu formu(Slika 22.). Ukoliko u ovom trenutku pokrenemo aplikaciju, moći ćemo dodati, ažurirati ili obrisati neke podatke iz naše odabrane tablice. Unutar ove aplikacije napravljena je kombinacija više formi na jednoj stranici. Razlog tome je što želimo da korisnik ažurira više tablica odjednom.

Kada kreiramo formu iz predloška, APEX automatski dodaje procese u pozadini koji se pobrinu da se naša forma ispravno inicijalizira te da se ispravno procesira. Ako dodajemo više formi na jednu stranicu, pogotovo ako su međusobno povezane vanjskim ključevima, moramo paziti kojim redom će se odvijati procesiranje. Sve se to može namjestiti u lijevom odjeljku sučelja(Slika 23.). Također, u istom odjeljku možemo spojiti sekvencu na odabrani stupac.



Slika 23. Redoslijed formi i sekvenca

3.9. Interaktivni izvještaj

U ovoj aplikaciji administrativni dio sučelja sastoji se od formi i interaktivnih izvješća. Interaktivna izvješća pružaju moćne značajke za prilagodbu izvješća kao što su pretraživanje, filtriranje, sortiranje, grupiranje, agregacije, izračuni, grafikoni i više. Želimo da admin može dodavati nove lokacije, nove događaje, nove ulaznice, itd. To možemo riješiti preko APEX-ovih gotovih predložaka. Kreiramo novu stranicu tipa „Interaktivno izvješće“ (*engl. Interactive Report*). Kod kreiranje interaktivnog izvješća primjećujemo kako APEX odmah ponudi opciju kreiranja forme koja će biti povezana sa novim interaktivnim izvješćem (Slika 24.). Isto tako, postoji opcija dodjeljivanja stranica grupama unutar navigacije. Želimo da ovaj dio aplikacije bude pod upravljanjem administratora tako da unutar roditeljskog unosa navigacijskog izbornika možemo dodijeliti vrijednost Administrator. Napravimo interaktivna izvješća i forme za sve tablice kojima želimo upravljati.

Create Interactive Report

Page Definition

- Page Number: 4
- Name: Korisnik_report
- Page Mode: Normal (Modal Dialog)
- Include Form Page: ☒
- Form Page Number: 6
- Form Page Name: Korisnik_form

Data Source

- Data Source: Local Database (REST Enabled SQL Service, REST Data Source)
- Source Type: Table (SQL Query)
- Table / View Owner: WKSP_ECETINA
- Table / View Name: KORISNIK

Navigation

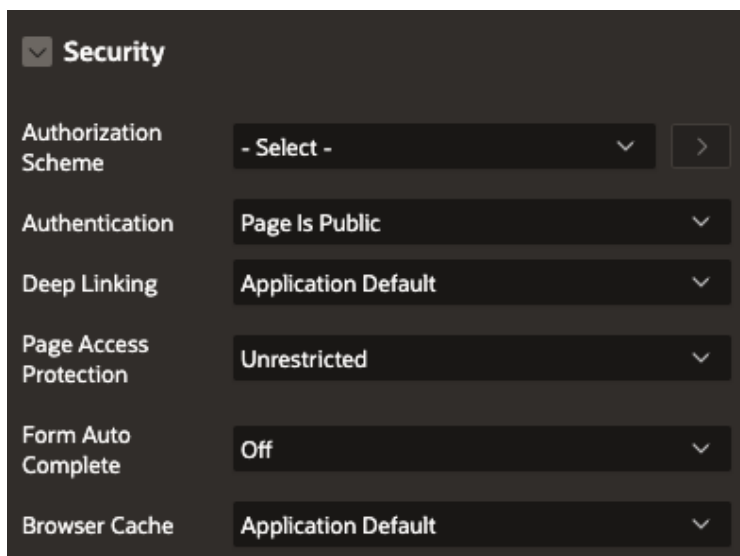
- Breadcrumb: ☒
- Breadcrumb Parent Entry:
- Navigation: ☒
- Navigation Preference: Create a new entry
- Parent Navigation Menu Entry: Administrator
- Icon: fa-table

< Cancel Next >

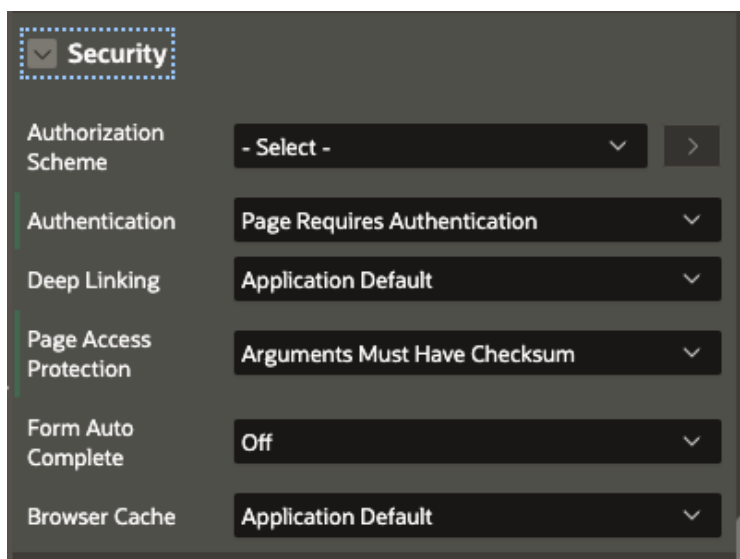
Slika 24. Kreiranje interaktivnog izvještaja

3.10. Uklanjanje prijave

Ideja aplikacije je da se korisnik koristi stranicom kao gost, te da tek kod kupnje upisuje svoje privatne podatke koji se onda ubacuju u tablicu Korisnik. Da bi to omogućili, na svakoj stranici za koju želimo da bude javna (*engl. Public*), pod “sigurnost” (*engl. Security*) autentifikaciju postavimo na “Stranica je javna” (*engl. Page is Public*) (Slika 25.). Time smo korisniku omogućili pristup stranici bez prijave ili registracije. Stranice koje hoćemo da i dalje koriste autentifikaciju pustimo na “stranica zahtjeva autentifikaciju” (*engl. Page Requires Authentication*) (Slika 26.).



Slika 26. Primjer javne stranice



Slika 25. Primjer stranice zaštićene autentifikacijom

U našoj aplikaciji početnu stranicu, stranice događaja te stranicu kupnje ostavljamo na javnim sigurnosnim postavkama dok naša interaktivna izvješća osiguramo nužnom autentifikacijom.

3.11. Izrada administrativne sheme

U APEX-u postoji predložak imenom „Kontrola pristupa“ (*engl. Access Control*) (Slika 27.). Kontrola pristupa našoj aplikaciji dodaje ulogu čitatelja, suradnika i administratora te odgovarajuću shemu autorizacije. Kreiramo novu stranicu, odaberemo kontrolu pristupa te odaberemo kreaciju nove administrativne stranice te kreaciju novog unosa navigacijskog izbornika. Ta opcija će nam u navigacijskom izborniku napraviti novu stakvu u koju možemo spremiti sve stavke koje želimo da budu pod administrativnim privilegijama.

Create Access Control Pages

This wizard creates pages to manage an access control list. Use the Application Access Control shared component to associate application roles with application users. This wizard also adds a **reader**, **contributor** and **administrator** role and corresponding authorization scheme to your application. Apply these authorization schemes to pages and page components to manage access by user and role.

* Starting Page Number: 10020

Page Group: - Select Page Group -

Build Option: Feature: Access Control

* Administration Page Preference: ☒ Create a new page ☐ Identify an existing page

* Administration Page Number: 10100

* Administration Page Name: Administration

Administration Page Navigation Preference: ☒ Create a new navigation menu entry ☐ Do not associate this page with a navigation menu entry ☐ Identify an existing navigation menu entry for this page

* New Navigation Menu Entry: Administration

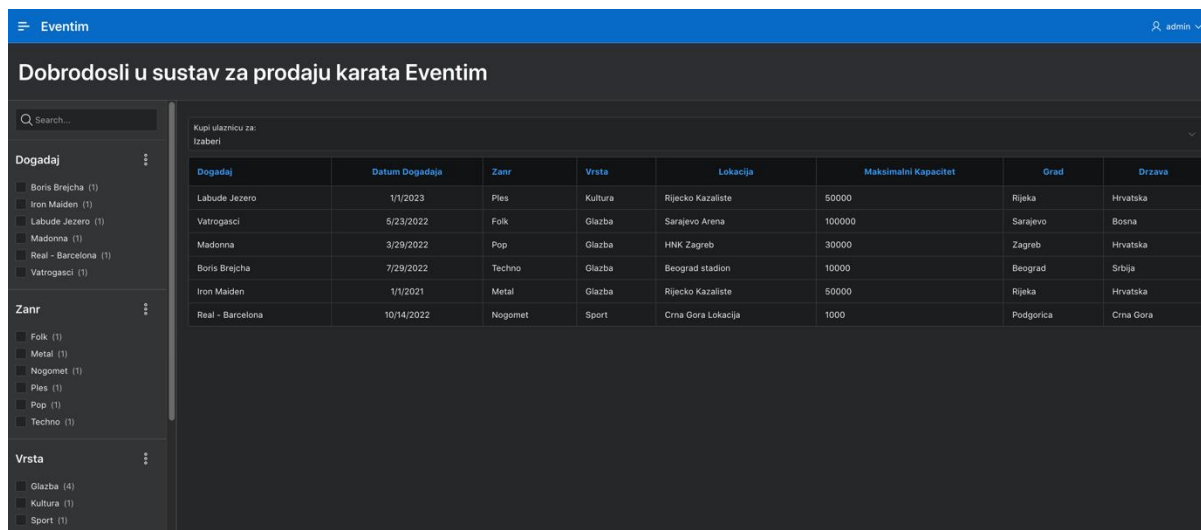
Parent Navigation Menu Entry: - No parent selected -
Home
Početna
forma
drzava_interactive_report
Administration
... (Prijava)

< Cancel Next

Slika 27. Kontrola pristupa

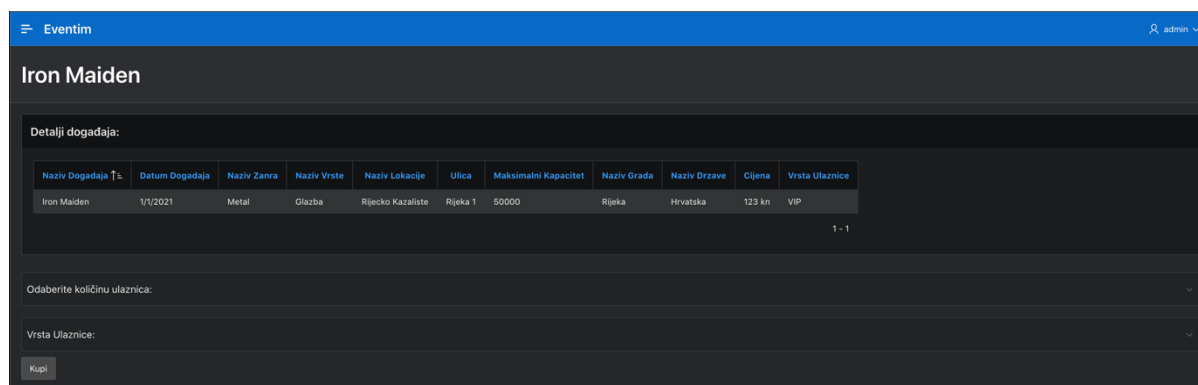
4. Gotova aplikacija

Krajnji rezultat izrade aplikacije je prikazan u nastavku. Početna stranica aplikacije s kojom se susreće korisnik prikazana je na Slici 28. Na lijevoj strani prikazana je fasetirana pretraga. U sredini se nalazi pregled svih događaja te se iznad pregleda može odabrati događaj za kojeg se želi kupiti karta.



Slika 28. Početna stranica

Nakon što se korisnik odluči za kartu, klikom odabere događaj te ga aplikacija odvede na stranicu događaja (Slika 29.).



Slika 29. Stranica događaja

Ispod tablice detalja događaja, korisnik može odabrati količinu i vrstu ulaznica koje mu odgovaraju. Klikom na tipku *Kupi* aplikacija vodi korisnika na stranicu kreiranja računa.

The screenshot shows the 'New' form in the Eventim application. The form is divided into two main sections: 'New' and 'Korisnik'. The 'New' section contains fields for 'Naziv Događaja' (Event Name) with the value 'Iron Maiden' and 'Datum Događaja' (Event Date) with the value '1/1/2021'. The 'Korisnik' section contains fields for 'Ime' (First Name), 'Prezime' (Last Name), 'Adresa' (Address), 'Datum Rodjenja' (Date of Birth), 'Spol' (Gender), 'Email', and 'Broj Mobilnoga' (Mobile Number). The 'Datum Rodjenja' field has a calendar icon. The bottom of the form shows a status bar with various icons and the text 'App 112679', 'Page 5', 'Session', 'Debug', 'Quick Edit', 'Customize', and 'admin'.

Slika 30. Kreiranje računa - prvi zaslon

The screenshot shows the 'Ulaznice' form in the Eventim application. The form is divided into three main sections: 'Ulaznice', 'Stavka Racuna', and 'Racun'. The 'Ulaznice' section contains fields for 'Cijena' (Price) with the value '123' and 'Vrsta Ulaznice' (Ticket Type) with the value 'VIP'. The 'Stavka Racuna' section contains a field for 'Kolicina Ulaznica' (Quantity of Tickets) with the value '2'. The 'Racun' section contains fields for 'Datum Kupnje' (Purchase Date) with the value '9/19/2022', 'Ukupni iznos' (Total Amount) with the value '246', 'Nacin Placanja' (Payment Method), and 'Nacin Preuzimanja' (Pickup Method). The bottom of the form shows a status bar with various icons and the text 'App 112679', 'Page 5', 'Session', 'Debug', 'Quick Edit', 'Customize', and 'admin'. There is a 'Poništi' (Cancel) button on the left and a 'Kupi' (Buy) button on the right.

Slika 31. Kreiranja računa - drugi zaslon

Na stranici kreiranja računa (Slika 30. i Slika 31.) korisnik upisuje svoje osobne podatke, način plaćanja i način preuzimanja. Klikom na tipku *Kupi* korisnik kupuje ulaznice i njegovi podaci se upisuju u bazu podataka.

5. Zaključak

Uz pomoć početnih modela i dijagrama izrađena je aplikacija koja predstavlja simulaciju puno kompleksnije web aplikacije Eventim, sustava za prodaju karata. Korisnik bez prijave ili registracije može navigirati stranicom te tražiti i kupiti ulaznice za razne događaje. Administrator ima mogućnosti dodavanja, brisanja ili ažuriranja svih podataka nad svim tablicama.

Tokom cijelog postupka izgradnje aplikacije upoznavali smo se s načinom korištenja Oracle APEX platforme. Pokazali smo da Oracle APEX ima mnoštvo pogodnosti i prednosti, kao na primjer sučelje prilagođeno korisniku te low code način rada. Uz navedene pogodnosti, možemo pretpostaviti da je prosječnom programeru nakon upoznavanja s platformom jednostavno izgraditi web aplikaciju s nekoliko osnovnih značajki. To ne znači da Oracle APEX nije u mogućnosti dovesti aplikaciju do razine funkcionalnosti nekih poznatijih razvojnih platformi već da je dobro dizajniran i prilagođen korisniku.

Kao i svaki software, Oracle APEX osim prednosti ima i nedostatke. Jedan od glavnih nedostataka APEX-a je odsutnost sustava za verzioniranje, dakle, ako na aplikaciji radi veliki tim programera, može se javiti problem ukoliko svi u isto vrijeme počnu raditi promjene. Neke od konfiguracija opcija nisu odmah vidljive te bi bolja organizacija radnog prostora uvelike koristila procesu razvoja. Također, u nekoliko navrata se dogodilo da bi se promjene na aplikaciji spremile, no one se ne bi reflektirale na samu aplikaciju prilikom pokretanja. Nakon par takvih slučajeva, korisnik bi se sigurno na svakoj promjeni zapitao je li aplikacija zaista pohranila nove promjene ili ne.

Kao zaključak izrade aplikacije, možemo reći da je Oracle APEX jedan moćan alat kojime se mogu koristiti i korisnici bez prijašnjeg iskustva. Zajednica raste svakodnevno te ukoliko nastave s razvojem platforme, smatram da bi mogla postati sve učestalija kod razvoja web aplikacija.

6. Literatura

1. IBM dokumentacija
<https://www.ibm.com/cloud/learn/relational-databases>
Zadnji pristup: 13.9.2022
2. Kissflow
<https://kissflow.com/low-code/history-of-low-code-development-platforms/>
Zadnji pristup: 10.9.2022
3. Oracle APEX
https://en.wikipedia.org/wiki/Oracle_Application_Express
Zadnji pristup: 13.9.2022

7. Popis priloga

Oracle APEX link za pristup aplikaciji:

<https://apex.oracle.com/pls/apex/r/ecetina/eventim/pocetna>

Username: admin

Password: admin123