

Istraživanje naprednih mogućnosti platformi s malo koda za razvoj aplikacija

Jakupec, Karlo

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:959718>

Rights / Prava: [Attribution 3.0 Unported/Imenovanje 3.0](#)

Download date / Datum preuzimanja: **2025-02-13**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Karlo Jakupec

**ISTRAŽIVANJE NAPREDNIH
MOGUĆNOSTI PLATFORMI S MALO
KODA ZA RAZVOJ APLIKACIJA**

ZAVRŠNI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Karlo Jakupec

JMBAG: 0016151610

Studij: Informacijski i poslovni sustavi

**ISTRAŽIVANJE NAPREDNIH MOGUĆNOSTI PLATFORMI S
MALO KODA ZA RAZVOJ APLIKACIJA**

ZAVRŠNI RAD

Mentorica:

Izv. prof. dr. sc. Dijana Oreški

Varaždin, rujan 2024.

Karlo Jakupec

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Upoznat ćemo se sa „malo koda“ programiranjem i pokazati CSS i HTML na „malo koda „ platformi. Uzet ćemo postojeće poduzeće na temelju kojeg ćemo napraviti standardne stranice u Oracle Apexu. Grafički ćemo vidjeti koje poslovne procese ima naše odabrano poduzeće i kako to poduzeće posluje. Pomoću ERA modela kreirat ćemo tablice u aplikaciji i povezati te tablice. Na temelju tih tablica napraviti ćemo razne forme, izvještaje i grafikone koji će nam biti temelj za uređivanje stranica. CSS će nam pomoći da stranice uredimo i podignemo na malo veću razinu. Osim toga vidjet ćemo gdje se i na koji način unose CSS i HTML. Za kraj spomenut ćemo AI i vidjeti kako se on koristi. Također ćemo spomenuti gdje se sve koristi i pokazati neki primjer. Na kraju ovog rada trebali bih imati nešto bolji i ljepši izgled aplikacije i nešto više predznanja za neke malo složenije i modernije stvari u Oracle apexu.

Ključne riječi: Malo koda, Oracle apex, CSS, HTML, SQL, umjetna inteligencija

Sadržaj

1. Uvod.....	1
2. Platforme s malo koda.....	2
3. Informacije o poduzeću.....	3
4. PK MATRICA	4
4.1. Popis procesa.....	4
4.2. Prikaz PK matrice.....	6
5. ERA MODEL.....	8
6. Relacijski model.....	10
7. Primjena CSS-a	11
7.1. Customize i CSS.....	11
7.2. Inline CSS	20
7.3. CSS u textbox-u	24
6.4 HTML i CSS	27
8. Umjetna inteligencija u Oracle Apexu	30
9. Zaključak.....	33
10. Popis literature.....	34
11. Popis slika	35

1. Uvod

Živimo u dinamičnom vremenu gdje se sve brzo mijenja i vrlo je bitno da pratite trendove kako bi bili konkurentni na tržištu. Tako je i kod izrada aplikacija jer korisnici tih aplikacija žele da njihove stranice budu privlačnije i modernije od drugih. Što se tiče izrade web aplikacija, ako govorimo o nekim većim aplikacijama to je proces koji zahtjeva puno truda, znanja i ulaganja. Međutim ako govorimo o nekim manjim i intuitivnijim aplikacijama tada to može biti vrlo jednostavno.

Za kreiranje jednostavnijih aplikacija postoje razne platforme koje koriste malo koda programiranje, odnosno programiranje s malo koda. Jedna od takvih platformi je i Oracle Apex. Oracle Apex omogućuje izradu web i mobilnih aplikacija visoke kvalitete koje rješavaju stvarne probleme. Programeri će uz malo truda napraviti funkcionalnu aplikaciju, a korisnici će u brzom vremenu dobiti aplikaciju na korištenje.

U ovom radu pokušat ćemo prikazati kako aplikaciju podignuti na višu razinu. Želimo u izradu uložiti malo više truda i napraviti što moderniju aplikaciju. Koristit ćemo neke opcionalne stvari koje nisu bitne za svaku izradu. Točnije, koristit ćemo se CSS-om i HTML-om kako bi uredili izgled aplikacije.

Najprije ćemo odabrati neko stvarno poduzeće i upoznati se sa njihovim procesima i načinom djelovanja. Zatim ćemo sve to prikazati pomoću tablica i dijagrama. Nabrojat ćemo i opisat procese u tom odabranom poduzeću. Zatim slijedi PK matrica koja će nam prikazati koje sve dokumente ili klase čitaju, kreiraju i ažuriraju pojedini poslovni procesi. Ono što najbolje prikazuje povezanost svih klasa je ERA model koji će nam biti od velike važnosti prilikom kreiranja tablica u bazi i povezivanje tih tablica sa vanjskim i primarnim ključevima. Nakon svega toga prebacujemo se na Oracle Apex gdje prvo kreiramo tablice pa ih povezujemo i nakon toga kreiramo potrebne forme, izvještaje, grafikone i sl. Osim toga kreirat ćemo i neke nove i drugačije stranice kako bi dobili što bolju i funkcionalniju aplikaciju. Nakon što ubacimo CSS i HTML trebali bi dobiti aplikaciju koje će zasigurno koristiti svrhu i izgledati bolje nego one standardne.

2. Platforme s malo koda

Ovaj rad temelji se na platformi *Oracle Apex* koja spada u platforme s malo koda. Malo koda programiranje sve se više razvija i sve više se koristi. Za vrijeme korona krize, malo koda programiranje je bilo jako traženo. Firme su se susretale sa novim izazovima i morale su reagirati brzo da bi ostale konkurente na tržištu. Malo koda programiranje znatno ubrzava i olakšava izradu aplikacija. Umjesto pisanja gomilu linija koda te raznih funkcija ovdje se „drag and drop“ načinom rade aplikacije. Nije potrebno nikakvo predznanje da bi se mogle napraviti funkcionalne i intuitivne aplikacije. Stoga je malo koda programiranje stvorilo novu vrstu programera koju nazivaju „programeri građani“. To su programeri koji nemaju nikakvo preveliko znanje o programiranju a izrađuju aplikacije. Iz ovoga možemo zaključiti da danas može svatko postati programer, ali „programer građanin“ i da svatko može napraviti aplikaciju. Neke od prednosti malo koda programiranja su brzina, implementacija jednim klikom i jednostavnost. Upravo to opisuje i *Oracle Apex* u kojem radimo aplikaciju za ovaj rad. Međutim ipak malo koda programiranje zahtijeva neko predznanje jer postoje dijelovi gdje treba ručno unositi kod tako da nije baš sve „drag and drop“. Smatra se da malo koda programiranje je programiranje u omjeru od 80 prema 20. Što znači da se 80% radi „drag and drop“ načinom a ostalih 20 se radi ručno. Platforme s malo koda svojim načinom kreiranja aplikacija znatno smanjuju troškove firme, što je još jedna od prednosti. Zbog brže izrade aplikacije, programeri imaju više vremena za posvetiti se nekim drugim stvarima. (Starčević, 2021)

3. Informacije o poduzeću

Aplikaciju za ovaj rad ćemo raditi na primjeru stvarnog poduzeća. Poduzeće koje smo odabrali je Jan Spider iz Pitomače. Tvrtka Jan Spider bavi se uzgojem i preradom ljekovitog i aromatičnog bilja za čajeve. Jan Spider samo je jedna od nekoliko firmi iz grupacije koja se bavi kompletnom proizvodnjom čajeva. Osim za vlastite potrebe, bilje prerađuju i za vanjsko tržište. Jan Spider u grupaciji je sa tvrtkom Đolta koja se bavi uzgojem bilja i sa tvrtkom Herbarium koja se bavi pakiranjem bilja. Stoga će naša aplikacija imati razne forme poput tipova bilja i kupci bilja te razne otpremnice, izvještaji i radni nalozi koji proizlaze iz uzgoja, prerade ili pakiranja.

4. PK MATRICA

PK matrica nam daje vizualni prikaz koje klase kreiraju, čitaju i ažuriraju pojedini procesi u firmi. Stoga ćemo u poglavlju 4.1. prikazati koje sve procese ima naša odabrana firma. Svakom tom procesu ćemo u PK matrici dodijeliti klasu koju taj proces kreira, čita i ažurira. Tako ćemo na jednom mjestu dobiti prikaz svih poslovnih procesa i klasa koje nastaju u tim procesima.

4.1. Popis procesa

Podsustav	Naziv procesa	Opis procesa
Nabava	Planirati nabavu	Poslovođa nabave s direktorom planira potrebe poduzeća
	Pripremiti natječaj	Napisati natječaj za odabir kooperanata koji će sudjelovati u proizvodnji bilja
	Sklopiti ugovor za nabavu	Odabrati kooperante i potpisati ugovore
Proizvodnja	Planirati i pripremiti	Isplanirati plan berbe i sadnje te pripremiti zemlju, strojeve i radnike
	Saditi	Posaditi ljekovito bilje na obrađenu zemlju
	Održavati	Zalijevati, špricanje protiv bolesti, kultivirati
	Žeti	Prikupljanje, kombajniranje i branje
Skladište	Napraviti raspored skladišta	Napraviti plan gdje će koje bilje biti spremjeno
	Puniti sušare	Prazne sušare napuniti iskombajniranim i prikupljenim biljem
	Održavati sušare	Paziti na temperaturu sušara i na kvalitetu sušenja
	Prazniti sušare	Isprazniti pune sušare
	Zaprimiti prerađeno i zapakirano bilje	Prikupiti bilje od kooperanata i svoje osušeno bilje

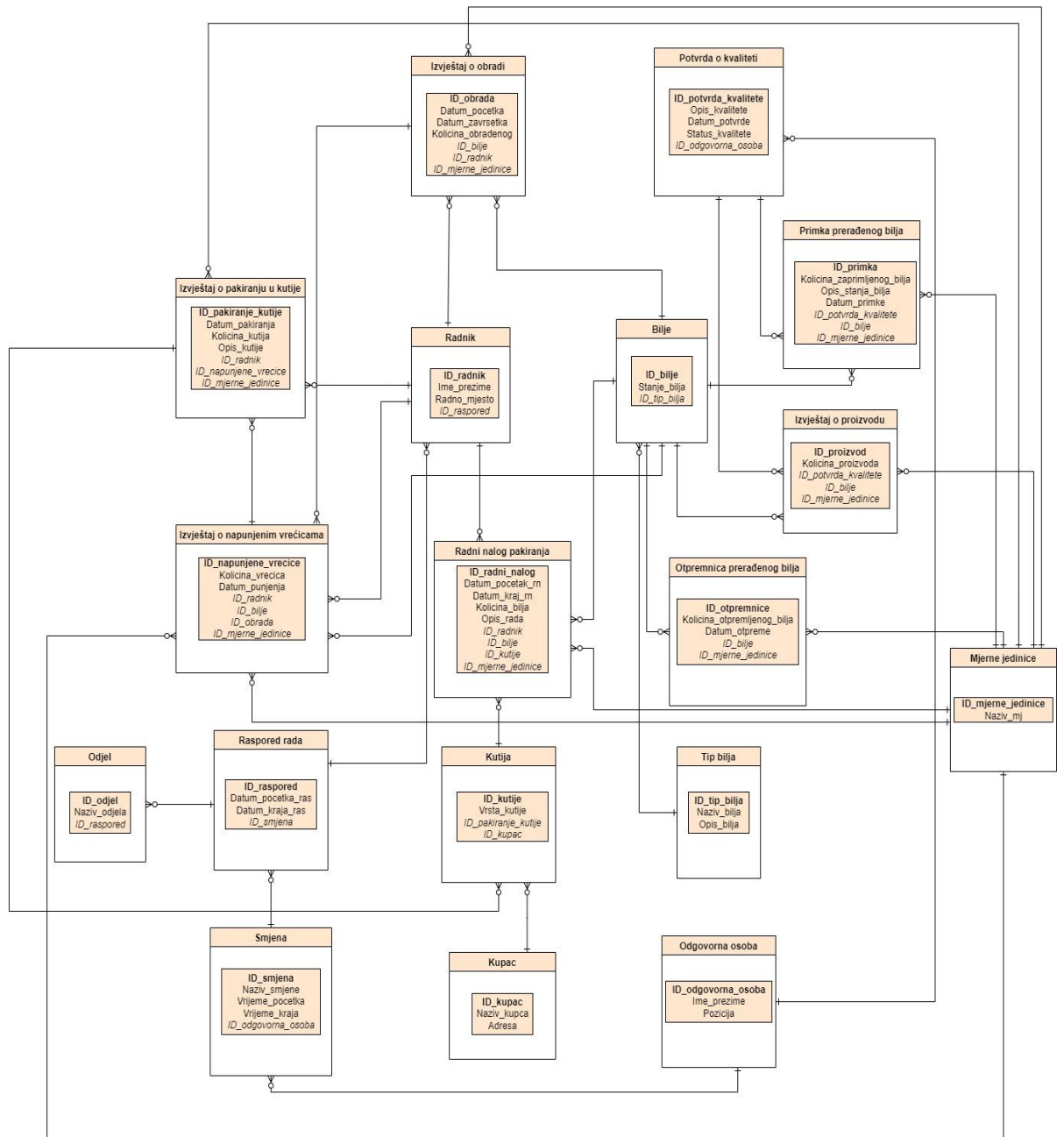
Slika 1: 1. dio popisa procesa

Prerada	Zaprimiti osušeno bilje	Voditi bilješke koliko je prikupljenog bilja
	Provjeriti kvalitetu	Provjeriti kvalitetu bilja kroz sustav provjere
	Rasporediti po kvaliteti	Odvojiti po razinama kvalitete
	Obraditi	Pustiti bilje u pogon
	Pakirati prerađeno bilje	Spremanje u jumbo vreće i spremanje u skladište
Pakiranje	Zaprimiti prerađeno bilje	Zaprimanje jumbo vreća sa prerađenim biljem iz skladišta
	Razvrstati bilje po proizvodnim linijama	Vaditi bilje iz jumbo vreća i slaganje istog na proizvodne linije ovisno o vrsti bilja
	Puniti vrećice	Puniti filter vrećice s biljem koje smo stavili na proizvodne linije
	Pakirati u kutije	Pakirati filter vrećice u kutije označene prema vrsti i kupcu
Prodaja	Tražiti ponude	Istražiti ponude proizvođača čajeva
	Sklopiti ugovor za prodaju	Potpisati ugovore s odabranim proizvođačima čajeva
	Unaprijediti prodaju	Istražiti načine za poboljšanje kvalitete i povećanje broja zadovoljnih kupaca
Računovodstvo	Obračunati troškove proizvodnje i nabave	Preuzeti račune od poslovođa pa ih razvrstati te zbrojiti
	Platiti troškove	Platiti obračunate troškove proizvodnje, nabave i ostalih aktivnosti poduzeća
	Izraditi financijska izvješća	Prikupiti izvješća o proizvedenim i prodanim proizvodima te troškovima i izračunati razliku na godišnjoj razini
	Izraditi račune	Ispisati račune za kupce i kooperante na temelju izrađenih ugovora i izdatnica

Slika 2: 2. dio popisa procesa

Po dijagonali vidimo da svaki proces ima svoju klasu koju kreira, čita i ažurira, no međutim postoje i druge klase koje moraju procesi čitati da bi mogli obavljati posao. Svakom bojom nam je označen drugi proces a oznake u tablicama C, R i U se odnose na *Create, Read i Upload*.

5. ERA MODEL



Slika 4: ERA model

ERA model se koristi za prikaz svih entiteta nekog odabranog sustava kao i njihove međusobne povezanosti te atributa spomenutih entiteta. Prikazani ERA model sastoji se od 18 tablica, odnosno entiteta te prikazuje podsustav odabranog i obrađenog poduzeća. U ovome slučaju radi se o poduzeću Spider Grupe, a odabrani podsustav je podsustav Pakiranje. Taj podsustav se u PK matrici nalazi na petom mjestu i označen je narančastom bojom.

Kad pogledamo na ERA model možemo zaključiti da nekoliko entiteta ima više veza u usporedbi na druge entitete te su ti entiteti najvažniji u ovome modelu. U ovome slučaju radi se o entitetima Radnik i Bilje.

Entitet Radnik veže se na gotovo sve dokumente budući da je on važan radi odrade posla ili nekog potpisa. Entitet Bilje veže se na sve dokumente vezane uz proizvod koji se proizvodi u obrađenom poduzeću odnosno prerađeno bilje koje se pakira u filter vrećice za čaj.

Postoje mnogi dokumenti koji služe za normalno djelovanje poduzeća. Poduzeće je veliko i radi u tri smjene što znači da je kompleksno i da je potreban velik broj dokumenata kako bi se bilježio rad te izvještaji kako bi bilježili koliko je čega obrađeno da bi imali jasne evidencije.

Jedan od najjednostavnijih dokumenata je Tip bilja. On sadrži tri atributa: ID_tip_bilja, Naziv_bilja i Opis_bilja. Dokument koji je najviše u uporabi je Radni nalog pakiranja. Budući da se on izdaje svaki dan nekoliko puta, radi različitih smjena ili različitih zadataka. Sljedeći dokument je Primka prerađenog bilja koja nam služi da vodimo bilješke kada primimo prerađeno bilje i pripremamo ga za pakiranje. Potvrda o kvaliteti je dokument koji nam služi kako bi mogli zapisati razlike između bilja i njegove kvalitete. Odnosno služi kao potvrda koliko je određena količina bilja te samo bilje. Izvještaj o napunjenim vrećicama nam služi kako bi kontrolirali daljnje pakiranje te ga koristimo u Izvještaju o pakiranju u kutije. Izvještaj o pakiranju u kutije koristimo kod Kutije budući da poduzeće pakira za različite kupce pa kako bi svaki kupac dobio proizvod u odabranoj ambalaži. Dokumenti povezani s radnikom su nam Raspored rada, Odjel i Smjena koji nam pomažu u organizaciji radnika kako bi znali koji dan rade u kojoj smjeni i kako bi organizirali koji broj radnika nam u određenom period treba u određenom odjelu. Otpremnica prerađenog bilja je dokument koji šaljemo zajedno s biljem kako bi imali evidenciju koliko smo bilja obradili i koliko bilja je idući podsustav trebao zaprimiti.

Mjerne jedinice su entitet koji nam pomaže pri zapisu količina bilja u različitim dokumentima.

6. Relacijski model

Relacijska shema ERA modela sa slike 19 je sljedeća:

Izveštaj o obradi (ID_obrada, Datum_pocetka, Datum_zavrsetka, Kolicina_obradenog, ID_bilje, ID_radnik ID_mjerne_jedinice)

Potvrda o kvaliteti (ID_potvrda_kvalitete, Opis_kvalitete, Datum_potvrde, Status_kvalitete, ID_odgovorna_osoba)

Izveštaj o pakiranju u kutije (ID_pakiranje_kutije, Datum_pakiranja, Kolicina_kutija, Opis_kutije, ID_radnik, ID_mjerne_jedinice, ID_napunjene_vrecice)

Radnik (ID_radnik, Ime_prezime, Radno_mjesto, ID_raspored)

Bilje (ID_bilje, Stanje_bilja, Vrsta_bilja, ID_tip_bilja)

Primka prerađenog bilja (ID_primka, Kolicina_zaprimljenog_bilja, ID_bilje, Opis_stanja_bilja, Datum_primke, ID_potvrda_kvalitete, ID_mjerne_jedinice)

Izveštaj o proizvodu (ID_proizvod, Kolicina_proizvoda, ID_potvrda_kvalitete, ID_bilje, ID_mjerne_jedinice)

Izveštaj o napunjenim vrećicama (ID_napunjene_vrecice, Kolicina_vrecica, Datum_punjenja, ID_radnika, ID_bilje ID_mjerne_jedinice, ID_obrada)

Radni nalog pakiranja (ID_radni_nalog, Datum_pocetak_rn, Datum_kraj_rn, Kolicina_bilja, Opis_rada, ID_radnik, ID_bilje, ID_kutije ID_mjerne_jedinice)

Otpremnica prerađenog bilja (ID_otpremnice, Kolicina_otpremljenog_bilja, Datum_otpreme, ID_bilje, ID_mjerne_jedinice)

Odjel (ID_odjel, Naziv_odjela, ID_raspored)

Raspored rada (ID_raspored, Datum_pocetka_ras, Datum_kraja_ras, ID_smjena)

Smjena (ID_smjena, Naziv_smjene, Vrijeme_pocetka, Vrijeme_kraja, ID_odgovorna_osoba)

Kutija (ID_kutije, vrsta_kutije, ID_pakiranje_u_kutije, ID_kupca)

Tip bilja (ID_tip_bilja, Naziv_bilja, Opis_bilja)

Kupac (ID_kupca, Naziv_kupca, Adresa)

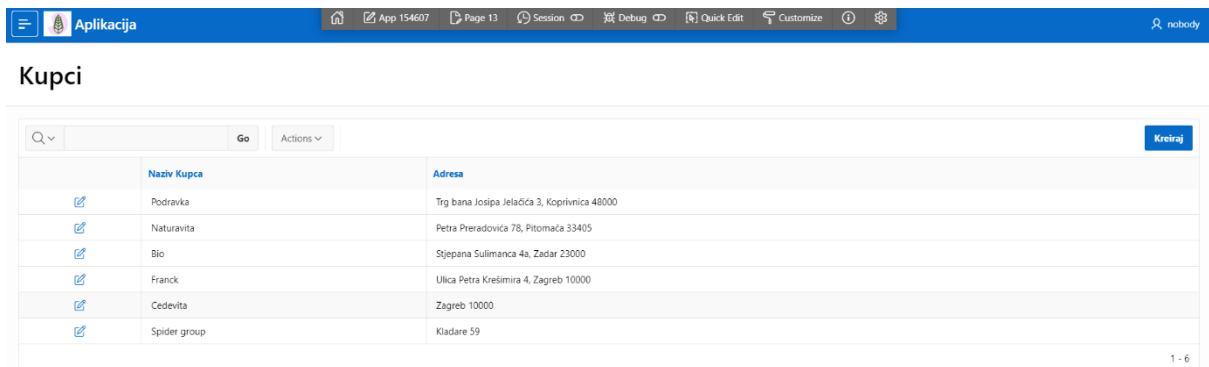
Odgovorna osoba (ID_odgovorna_osoba, Ime_prezime, Pozicija)

Mjerne jedinice (ID_mjerne_jedinice, Naziv_mj)

7. Primjena CSS-a

7.1. Customize i CSS

Sve promjene sa CSS-om ćemo raditi na istoj stranici kako bi lakše i bolje vidjeli promjene koje su se dogodile. Stoga ćemo prvo prikazati izgled stranice koja je uvijek na početku ista i to je standardni izgled svake stranice.

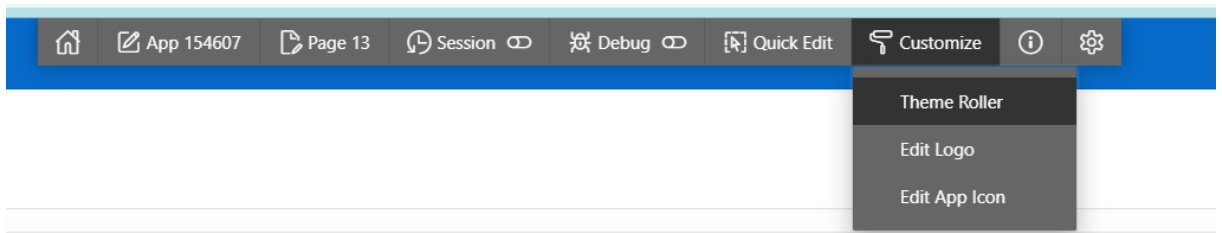


The screenshot shows the Oracle Apex application interface. At the top, there is a blue header bar with the text "Aplikacija" and a search icon. Below the header, the title "Kupci" is displayed. The main content area contains a table with two columns: "Naziv Kupca" and "Adresa". The table lists six customers with their names and addresses. A search bar is located at the top left of the table, and a "Kreiraj" button is at the top right. The table footer shows "1 - 6".

	Naziv Kupca	Adresa
	Podravka	Trg bana Josipa Jelačića 3, Koprivnica 48000
	Naturavita	Petra Preradovića 78, Pitomača 33405
	Bio	Stjepana Sulimanca 4a, Zadar 23000
	Franck	Ulica Petra Krešimira 4, Zagreb 10000
	Cedevita	Zagreb 10000
	Spider group	Kladare 59

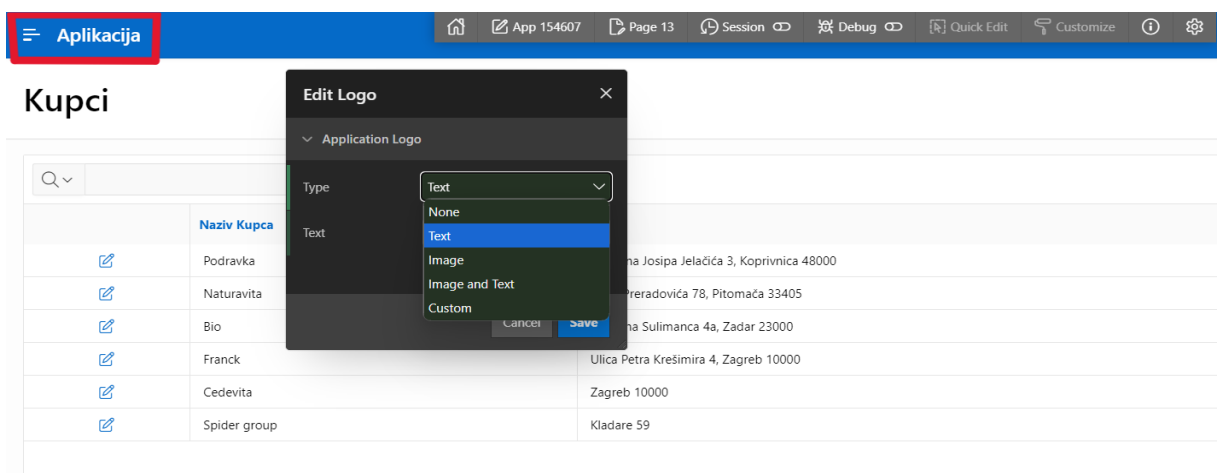
Slika 5: Početna stranica kupci

Jedan od načina kako možemo urediti stranicu na Oracle Apexu je *Customize* opcija. Svaka stranica u Oracle Apexu ima alatnu traku na kojoj se nalaze prečice do određenih postavki. Također se na toj alatnoj traci nalazi i opcija *Customize* koju kad kliknemo dobijemo 3 opcije: *Theme Roller*, *Edit Logo* i *Edit App Icon*. *Customize* opcija služi za brzo i jednostavno uređivanje izgleda stranice. Točnije biramo temu ili sami slažemo postavljamo boje pozadine teksta i sl. Tema koju odaberemo u *Customizeu* ona se primjenjuje na sve stranice u aplikaciji, ne samo na onoj na kojoj se trenutno nalazimo.



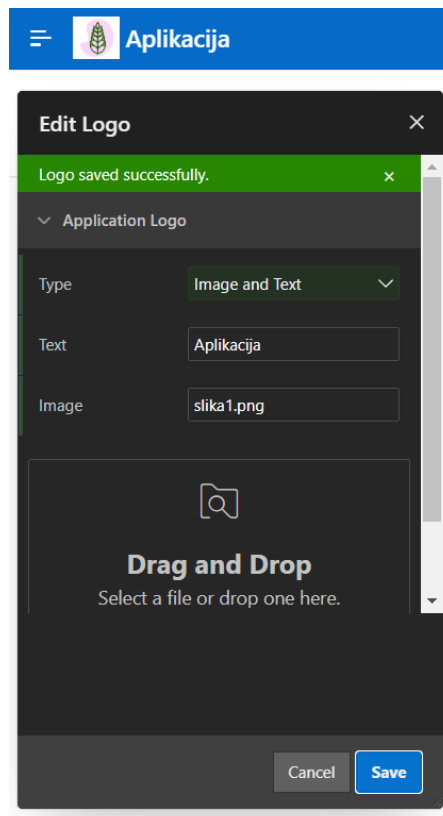
Slika 6: Opcija za uređivanje boja

Prođimo prvo kroz *Edit Logo* opciju. Kad kliknemo na *Edit Logo* tada biramo *type* našeg loga u *Navigation Baru*.



Slika 7: Uređivanje ikone naslova

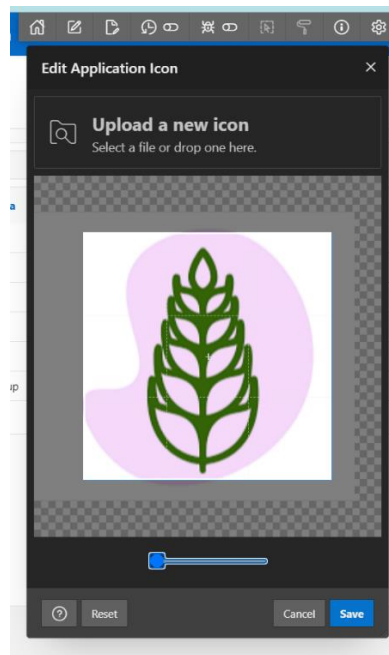
Navigation Bar je ova plava traka koja se prostire uzduž po vrhu aplikacije i možemo vidjeti u crvenom kvadratu gdje piše *Aplikacija*, to je dio koji uređujemo u *Edit Logo* postavkama. Na slici 7 možemo vidjeti koje sve opcije imamo, a to su redom: *text*, *image*, *image and text* i *custom*. Mi ćemo odabrati opciju *Image and Text* kako bi gore u *Navigation Baru* uz tekst imali i neku ikonicu.



Slika 8: Slika i tekst kod uređivanja ikone

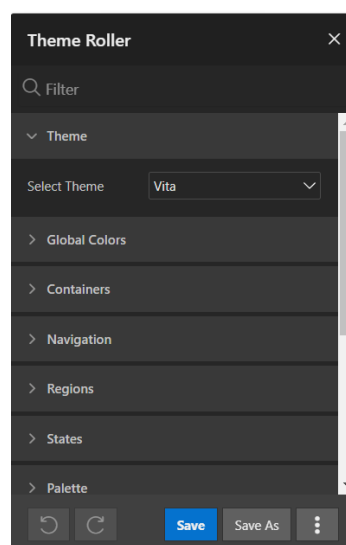
Na slici 8 možemo vidjeti kako izgleda kada odaberemo *Image and Text*. U *Drag and Drop* prozor smo ubacili našu sliku, a kod *Text* opcije smo napisali *Aplikacija*. Na samom vrhu slike u *Navigation Baru* možemo odmah vidjeti kako to izgleda.

Sljedeća opcija u *Customizeu* je *Edit App Icon* i na slici 9 možemo vidjeti kako to izgleda kad otvorimo tu opciju. Ukratko, ova opcija nam služio za sređivanje i *upload* ikone.



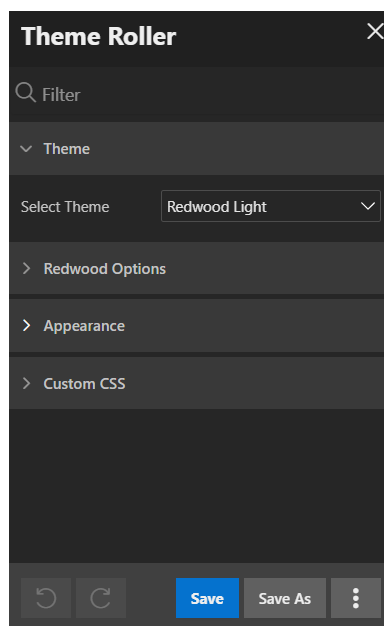
Slika 9: Uređivanje aplikacijske ikone

Zadnja i najkorisnija opcija u *Customizeu* je *Theme Roller*. Kad pritisnemo tu opciju otvaraju nam se razni padajući izbornici koje uglavnom služe za odabir boja. Izbor je velik, a padajući izbornici su sljedeći: *Theme*, *Global Colors*, *Containers*, *Navigation*, *Regions*, *States*, *Palette*, *Buttons*, *Forms*, *Interactive Reports*, *Layout*, *Custom CSS*. Na slici 10 možemo vidjeti kako to izgleda i ovdje ne vidimo sve opcije koje smo nabrojali ali ako scrollamo prema dolje onda ćemo ih vidjeti.



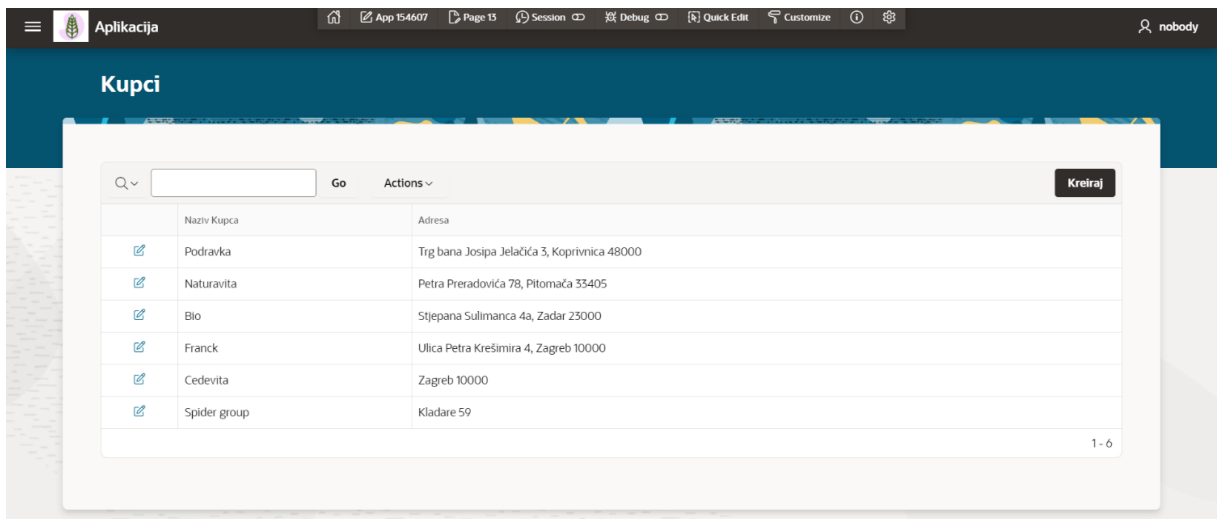
Slika 10: Teme

Sama aplikacija nudi nam 5 teme koje možemo odabrati ispod padajućeg izbornika *Theme*. Odabirom bilo koje od tih 5 dobivamo drugačiji izgled aplikacije odnosno raspored stvari je isti samo su boje drugačije pa tako imamo sljedeće teme: *Redwood Light*, *Vita*, *Vita - Dark*, *Vita - Red* i *Vita - Slate*. *Redwood Light* opcija malo je drugačija nego ostale. Ako odaberemo *Redwood Light* tada nećemo imati sve ona padajuća izbornika kao na slici 10 nego samo njih 4. Od tih 4 ostaje nam onaj glavni odnosno padajući izbornik *Theme* a preostala 3 su: *Redwood Options*, *Appearance* i *Custom CSS*.

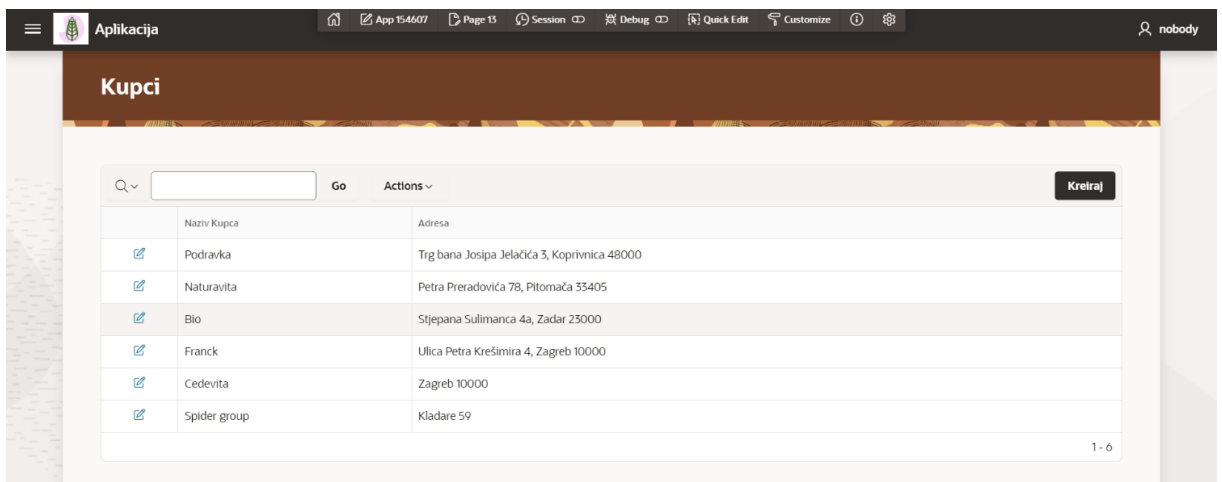


Slika 11: Tema crveno drvo

Na slici 11 možemo vidjeti kako izgledaju padajući izbornici kada odaberemo *Redwood Light*, a sada ćemo prikazati dvije slike kako može izgledati aplikacija ako malo posložimo postavke kod *Redwood Options* i *Appearance*.



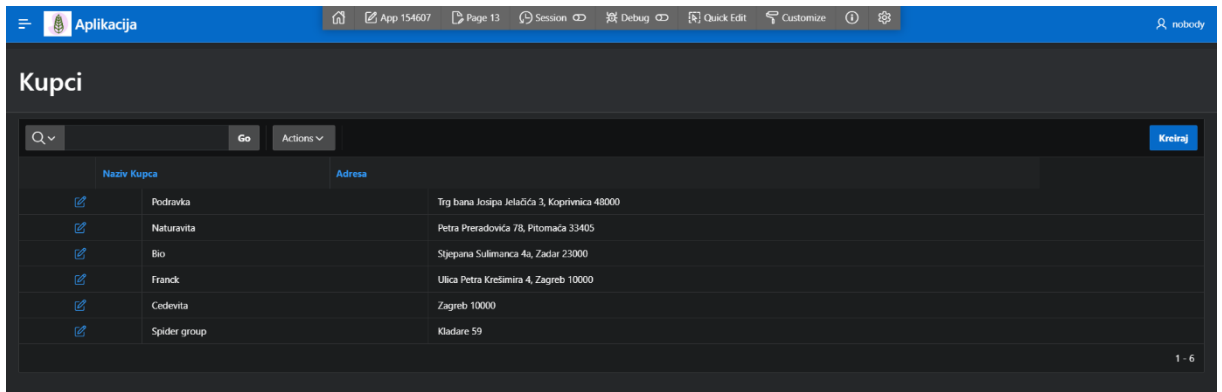
Slika 12: 1. Izgled aplikacije – crveno drvo



Slika 13: 2. izgled aplikacije – crveno drvo

Ovdje vidimo kako uz nekoliko klikova možemo izgled aplikacije podignuti na veću razinu. Naravno da tu možemo još uređivati neke stvari kod opcije međutim to ćemo malo kasnije.

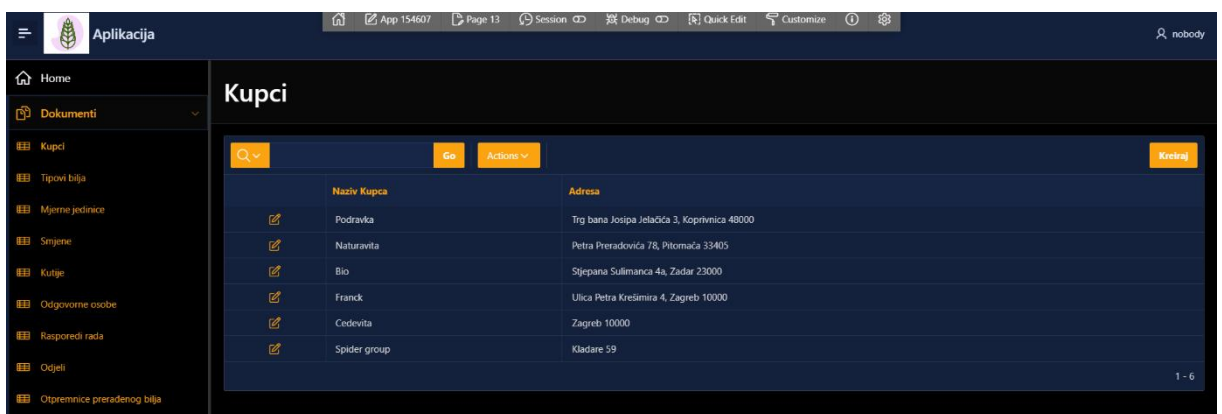
Ako u padajućem izborniku *Theme* odaberemo neku drugu temu npr. *Vita – Dark* tada nam izgled aplikacije izgleda ovako.



Slika 14: Crna tema

Kao što smo rekli, raspored stvari je isti kao i u onom prvom stanju samo je boja pozadine i slova drugačija i opet izgled aplikacije je malo moćniji od onog običnog.

Ovdje ćemo malo urediti stranicu sa CSS-om kako bi aplikacija dobila malo drugačiji izgled. Osim što ćemo koristiti CSS, staviti ćemo i neke druge boje u padajućim izbornicima. Tako ćemo dobiti aplikaciju kakvu mi želimo, a ne onakvu kakvu nam nudi pojedina tema. Odabirom željenih boja u padajućim izbornicima dobili smo sljedeći izgled aplikacije.



Slika 15: Uređena crna tema

Ovaj izgled nije dobiven jednim klikom na odabir neke određene teme nego je bilo više klikova i igranja s bojom teksta, pozadine, gumbova, menia s lijeve strane i sl.

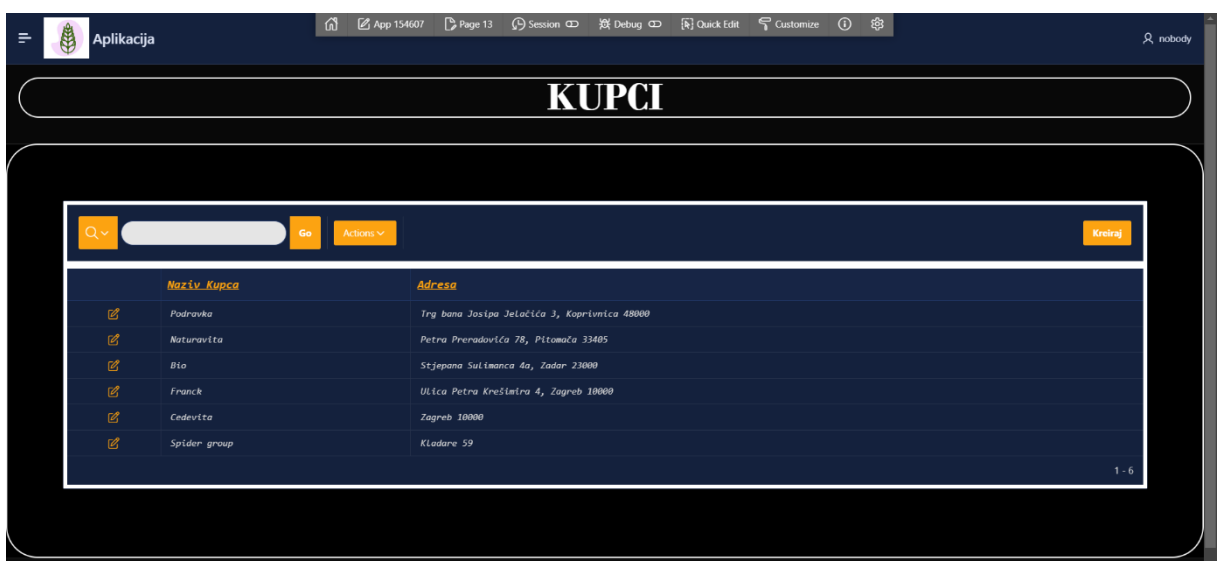
Nakon dodavanja željenih boja, sljedeća stavka je dodavanje CSS-a u *Custom CSS*. Prije nego pokažemo kako izgleda aplikacija nakon unosa CSS-a, moramo objasniti kako funkcionira unos CSS-a. Stvar je u tome što aplikacija već ima svoj CSS, koji se generira automatski kako kreiramo aplikaciju. Zbog toga treba voditi računa o pravilu nasljeđivanja.

Pravilo nasljeđivanja kaže da djeca nasljeđuju upute od roditelja a pravilo prednosti izgleda ovako:

style atribut > jednoznačno > eksplicitno > implicitno

Svi ovi selektori koji su nabrojani su jednostavni selektori no međutim postoje i kompleksni i spojeni selektori koji imaju prednost nad jednostavnim. Neki elementi u aplikaciji imaju klasu i ID što znači da neki elementi imaju jednoznačni selektor i eksplicitni selektor. Ako je u jednoznačnom selektoru postavljena boja slova npr. na plavu i ako mi pokušamo u eksplicitnom selektoru dodati neku drugu boju, to se neće izvršiti jer će element slušati jednoznačni selektor. Pojedini elementi imaju dosta roditeljskih elementa koji sadrže svoje jednoznačne, eksplicitne ili neke druge selektore, stoga treba dobro proučiti nad kojim elementom možemo vršiti promjene i na koji način.

Sada ćemo vidjeti kako izgleda aplikacija nakon nekih promjena koristeći CSS.



Slika 16: Crna tema sa CSS-om

Na slici 16 možemo vidjeti kako smo naslov *Kupci* postavili na sredinu stranice te smo promijenili font i veličinu naslova. Prostoru koji je namijenjen za naslov smo dodali rub te smo ga malo zaoblili. Također smo i glavnom dijelu stranice odnosno *main* dijelu dodali rub i zaoblili ga kako bi izgledao ujednačeno i sa naslovom. Što se tiče same tablice, u njoj smo promijenili font slova te smo podvukli i povećali naslove stupaca. Tablici smo podebljali rubove kako bi se ona bolje istaknula. Iste te rubove kao i sve ostale koje smo kreirali obojali smo u bijelo. Inputu koji služi za pretraživanje u tablici smo stavili bijelu pozadinu i također smo i njega zaoblili. Iako

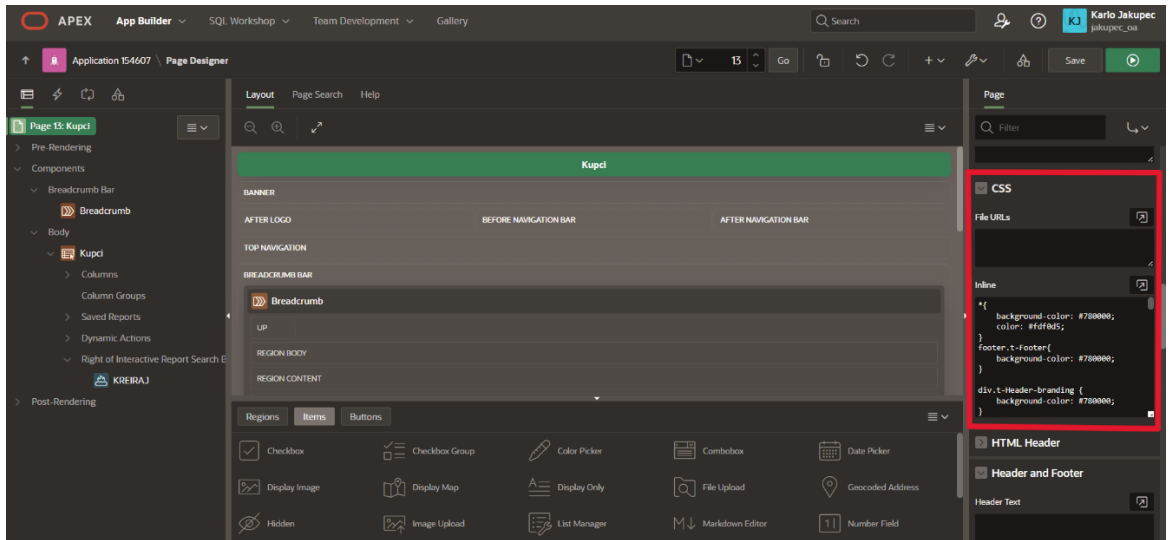
na slici 16 nije vidljivo ali prilikom prelaska miša iznad stupca, taj stupac nad kojim je miš, mijenja boju slova.

CSS kod za ovakav izgled stranice je sljedeći:

```
1 table tr {
2     font-family: 'Consolas';
3     font-style: italic;
4 }
5
6 a.a-IRR-headerLink {
7     font-size: 15px;
8     text-decoration: underline;
9 }
10
11 h1.t-Breadcrumb-label {
12     text-align: center;
13     text-transform: uppercase;
14     font-size: 50px;
15     border: 2px solid white;
16     border-radius: 40px;
17     font-family: 'Bodoni MT';
18 }
19
20 div.t-Body-title {
21     height: 100px;
22 }
23
24 table tr:hover{
25     color: #fca311;
26 }
27
28 main#main {
29     border: 2px solid white;
30     border-radius: 40px;
31 }
32
33 div.t-Body-contentInner {
34     padding: 70px;
35 }
36
37 div#R35002023031586462691_toolbar {
38     padding: 15px;
39     border: 5px solid white;
40 }
41
42 div#R35002023031586462691_content {
43     border: 5px solid white;
44 }
45
46 input.a-IRR-search-field {
47     border-radius: 40px;
48     margin: 5px;
49     background-color: #e5e5e5 ;
50 }
```

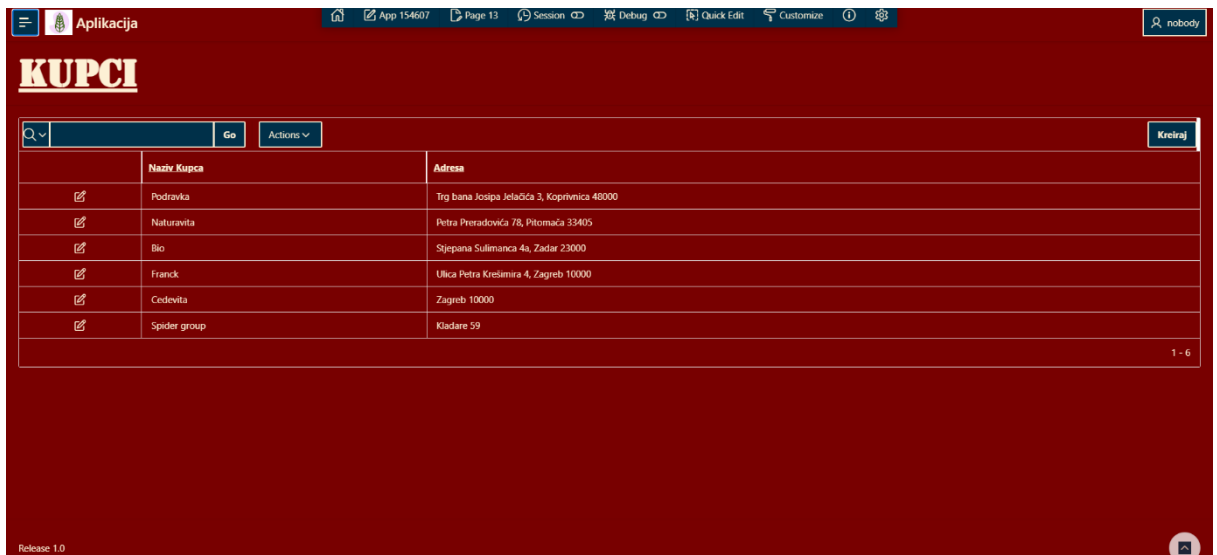
7.2. Inline CSS

Osim što CSS možemo unositi u *Customize* opciji, postoji još jedan način kako možemo stranice uređivati sa CSS-om. Prvo moramo otići u *Page designer* stranice koju želimo urediti.



Slika 17: Stranica za uređivanje aplikacije

Page Designer stranica izgleda ovako i ona služi za uređivanje odabrane stranice u aplikaciji. S desne strane u crveno označenom kvadratu možemo pronaći CSS postavke, točnije prostor za unos CSS-a iznad kojeg piše *Inline*. U taj prostor pišemo CSS kod kojim želimo promijeniti izgled stranice i nakon unosa potrebno je stisnuti tipku *Save* i pokrenuti aplikaciju te vidjeti promjene koje su se dogodile. Isto kao i kod pisanja CSS-a u *Customize* postavkama, treba voditi računa o pravilu nasljeđivanja.



Slika 18: CSS stranica

Ovako sada izgleda stranica gdje smo uređivali onu početnu stranicu sa slike 5 . Što se tiče izgleda možemo vidjeti da su promijenjene boje i font. Osim boje pozadine promijenili smo boje gumbova i boju teksta. Iako smo prvo koristili univerzalni selektor koji se odnosi na sve elemente, ipak nismo uspjeli sve s njime obojati u željenu boju stoga smo morali neke elemente posebno dohvaćati kako bi im promijenili boju pozadine. Elemente smo dohvaćali na način da smo pritisnuli desni klik na element nad kojim želimo vršiti promjene i odabrali opciju *Provjeri*. Pritiskom na opciju *Provjeri* otvara nam se novi prozor u kojem možemo vidjeti cijeli HTML i CSS kod naše stranice. Nismo morali puno tražiti po kodu gdje se nalazi element koji želimo uređivati jer nam je automatski označen dio koda koji se odnosi na taj element nad kojim smo pritisnuli desni klik. Nekad je označen roditeljski element od onog elementa koji tražimo pa smo sami morali ući malo dublje da dođemo do samog elementa. Nakon što smo pronašli traženi element pogledali smo klasu ili ID od tog elementa i nakon toga smo u se vratili u *Page designer* gdje smo za taj element pisali CSS kod. Klasu ili ID smo morali pronaći kako bi se povezali baš na taj div ili neki drugi element s određenom klasom ili ID. U kodu postoji više *span* elemenata stoga smo morali napisati klasu ili ID da znamo koji točno *span* element uređujemo.

Kod koji smo koristili da bi dobili izgled stranice kao na slici 18 je sljedeći:

```

1 *{
2   background-color: #780000;
3   color: #fdf0d5;

```

```

4 }
5 footer.t-Footer{
6     background-color: #780000;
7 }
8
9 div.t-Header-branding {
10     background-color: #780000;
11 }
12
13 tr th.a-IRR-header {
14     background-color: #780000;
15     border: 2px solid white;
16 }
17
18 button#R35002023031586462691_actions_button {
19     background-color: #003049;
20     color: #fdf0d5;
21     border: 2px solid #fdf0d5;
22 }
23
24 button.t-Button {
25     background-color: #003049;
26     border: 2px solid #fdf0d5;
27 }
28
29 div.a-IRR-colSelector{
30     background-color: #003049;
31 }
32
33 span {
34     background-color: #003049;
35 }
36
37 span.apex-logo-text {
38     background-color: #780000;
39 }
40
41 span.a-IRR-pagination-label {
42     background-color: #780000;
43 }
44
45 span.fa {
46     background-color: #780000;
47 }
48
49 span.t-Icon {
50     background-color: #003049;
51 }
52
53
54 button#R35002023031586462691_column_search_root {
55     padding: 0;
56     background-color: #003049;
57     border: 2px solid #fdf0d5;
58 }
59
60
61 input.a-IRR-search-field {
62     background-color: #003049;
63     border: 2px solid #fdf0d5;
64 }

```

```

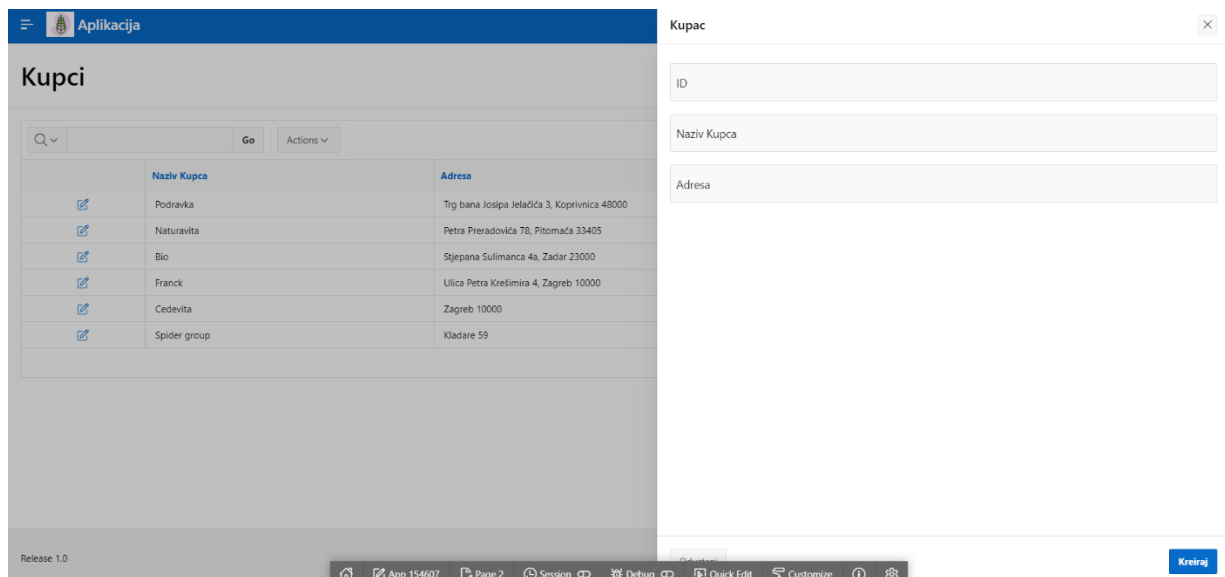
65
66 button#R35002023031586462691_search_button {
67     background-color: #003049;
68     border: 2px solid #fdf0d5;
69 }
70
71 li.t-NavigationBar-item {
72     background-color: #003049;
73     border: 2px solid #fdf0d5;
74 }
75
76 div#R35002023031586462691_toolbar {
77     padding: 4px;
78 }
79
80
81 a.a-IRR-headerLink {
82     color: #fdf0d5;
83     text-decoration: underline;
84 }
85
86 li {
87     background-color: #003049;
88 }
89
90 li.a-IRR-pagination-item {
91     background-color: #780000;
92 }
93
94 h1.t-Breadcrumb-label {
95     text-transform: uppercase;
96     text-decoration: underline;
97     font-size: 50px;
98     font-family: 'Stencil';
99     color: #fdf0d5;
100 }

```

Iz koda kao i sa slike možemo vidjeti da smo najviše mijenjali boje jer je to najlakše za promijeniti. Isto kao i kod CSS-a koji smo pisali u *Customize*, nije lako mijenjati sve na stranici najviše zbog tog pravila nasljeđivanja kojeg smo već nekoliko puta spomenuli. Osim tih boja jedna od većih promjena koje smo napravili je i uređenje naslova koji je sada malo bolje istaknut. Gumbovima smo dodali obrub kako bi i oni bolje izgledali. Postoji još puno mogućnosti za istaknuti promjene, ali je cilj bio pokazati gdje se piše CSS kod i kako pronaći element koji želimo promijeniti. Ako razumijemo osnovne korake, kasnije neće biti problem raditi veće promjene.

7.3. CSS u textbox-u

U ovom poglavlju prikazat ćemo 2 zanimljiva i vrlo korisna primjera kako sa CSS-om urediti *textbox* da izgleda bolje i funkcionalnije. Prvi primjer odnosit će se na prazan *textbox*, odnosno želimo da nam *textbox* javlja da nešto nije uredu odnosno da je prazan i da treba unijeti vrijednost u njega. *Textbox* će grešku javiti na način da će svoje rubove obojati u crveno kada je unesena vrijednost = null. Ovaj primjer ćemo prikazati također na stranici za kupce ali pošto nam je potreban *textbox* ovog puta ćemo otići na stranicu za kreiranje kupaca. Ta stranica nam se otvara kada pritisnemo gumb na kojem piše kreiraj na stranici za kupce.



Slika 19: Kreiranje kupaca

Stranica za kreiranje kupaca izgleda ovako i vidimo da su nam za novog kupca potrebne tri stvari i da imamo također i tri *textboxa*. Pošto nam je ID primarni ključ i bez njega ne možemo kreirati kupca onda ćemo za taj *textbox* napraviti da javlja grešku ako vrijednost nije unesena.

Prvi korak je da otvorimo *Page designer* za tu stranicu. Zatim za ID kupca kreiramo *Dynamic Actions*. U postavkama za *Dynamic Actions* s desne strane trebamo postaviti željeno ime, nakon toga kod opcije *When* pod stavku *Event* stavljamo *Lose Focus* i kod opcije *Client-side Condition* pod stavku *Type* stavljamo *Item is null*. Za *Type* smo odabrali da vrijednost mora biti nula što smo i željeli, međutim postoje i neke druge opcije.

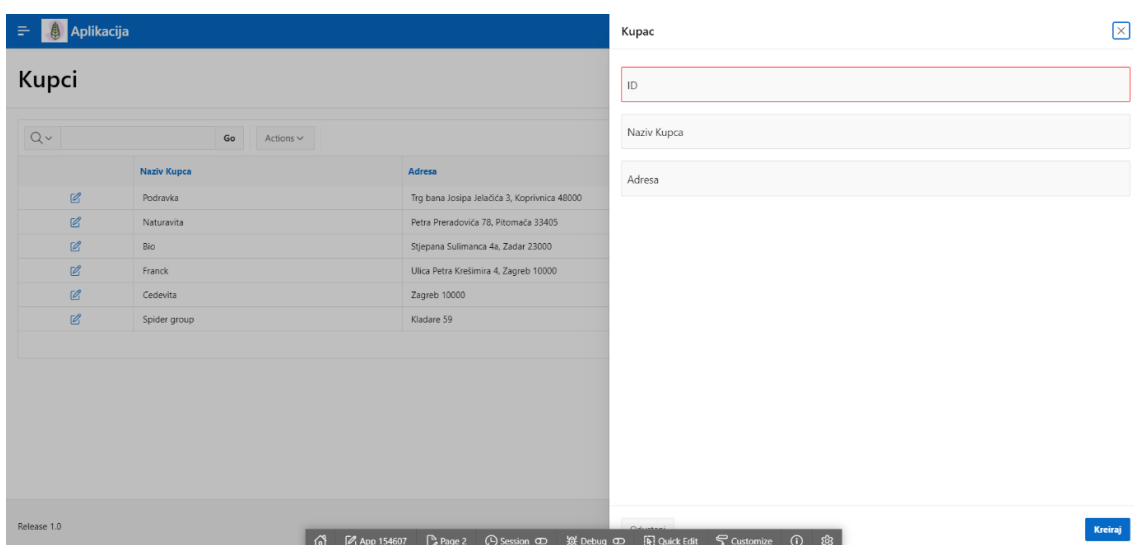
Svaka *Dynamic Actions* ima dva rješenja, a to su *True* i *False*. Zapravo trebamo definirati što će se dogoditi ako je *Dynamic Actions* istinit odnosno što ako nije istinit. Tako ćemo kod opcije *True* pod stavku *Action* odabrati *Add Class*, što znači da će se dodati neka klasa ako je *Dynamic Actions* jednak *True*. Klasa koja će se dodati definiramo ju u stavci *Class* koja se nalazi ispod *Settings* opcije. Stoga u to polje treba napisati ime klase, a to ime će u našem primjeru biti *crveni_rub*. Zadnja stvar koju još trebamo napraviti je da kod stavke *Item(s)* odaberemo *textbox* nad kojim se vrši promjena, a to je ID kupca.

Nakon toga trebamo kreirati akciju za rješenje *False*, odnosno što će se dogoditi ako je *Dynamic Actions* jednak *False*. To radimo na način da stisnemo desni klik na *True* akciju i odaberemo *Create Opposite Action*. Stavci koju smo kreirali trebamo kod *Action* staviti suprotno od onoga što smo napravili kod *True* rješenja, odnosno trebamo odabrati *Remove Class*. Također i kod stavke *Class* trebamo napisati ime klase koja će se maknuti kad rješenje bude *False*, što znači da trebamo ponovno upisati *crveni_rub*.

Nakon što smo postavili sve ove postavke, preostaje nam samo napraviti tu klasu *crveni_rub* koja se dodaje i briše sa našeg *textboxa*. U postavkama koje se odnose za cijelu stranicu također u *Page designeru* treba pronaći postavke za CSS i pronaći *Inline* polje gdje se unosi CSS kod.

```
1 .crveni_rub{
2     border-color: red;
3 }
```

Ovo je kod koji smo napisali i za kraj je potrebno spremi promjene i pokrenuti aplikaciju.



Slika 20: Crveni okvir za unos

Na slici 20 možemo vidjeti kako izgleda promjena koju smo napravili, a sada prelazimo na drugi primjer. Drugi primjer nešto je sličan ovom prvom no međutim sad nam je zadatak da *textbox* u kojem se nalazimo, promijeni boju pozadine kako bi se naglasio taj *textbox* u kojem se trenutno nalazimo.

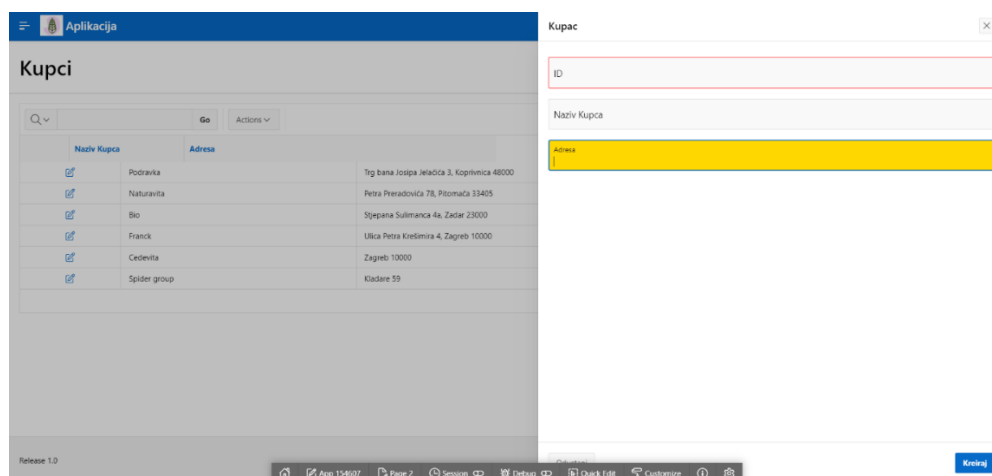
U ovom drugom primjeru također kreiramo *Dynamic Actions* samo ovog puta ćemo kreirati 2 *Dynamic Actions* i svaki od njih će imati samo jedno rješenje, a to je *True* rješenje. Kod prvog *Dynamic Actions* u postavkama postavimo ime *ITEM GET FOCUS* i ispod *When* kod stavke *Event* odaberemo *Get Focus*, a ispod stavke *Region* odaberemo našu stranicu na kojoj radimo ove promjene. Nakon toga trebamo posložiti postavke za rješenje *True* gdje ispod *Action* odaberemo *Add Class* pa ispod *Class* napišemo ima klase odnosno u ovom slučaju pišemo *pozadina* i na kraju ispod *Selection Type* odaberemo *Event Source*.

Nakon što smo posložili sve ove postavke, krećemo na kreiranje druge *Dynamic Actions*. Toj drugoj dodjeljujemo ime *LOSE FOCUS* i ispod *When* kod stavke *Event* odaberemo *Lose Focus* te ispod *Region* opet biramo stranicu na kojoj radimo promjene. Zatim odlazimo na rješenje *True* gdje sada ispod *Action* stavljamo *Remove Class* a ispod *Class* i *Selection Type* stavljamo iste vrijednosti kao i kod prošlog rješenja *True*.

Za kraj nam preostaje ispod stavke *Inline* napraviti klasu *pozadina* čiji CSS kod izgleda ovako:

```
1 .pozadina {  
2     background-color: gold ;  
3 }
```

Ako pokrenemo aplikaciju naša stranica za kreiranje kupaca sada izgleda ovako:



Slika 21: Zlatna pozadina za unos

Na slici 21 možemo vidjeti kako izgleda *textbox* koji je u fokusu odnosno onaj u kojem se nalazimo ili unosimo neku vrijednost. Također možemo vidjeti kako uz nekoliko linija CSS koda i nekoliko sitnih postavka možemo imati puno bolji i intuitivniji izgled *textboxova*, a samim time i cijele stranice.

U oba primjera mi možemo pomoću CSS-a dodati još neke promjene. Mogli smo umjesto crvenog ruba staviti crvenu pozadinu kao upozorenje da nešto nije uredu. Također smo umjesto ove žute pozadine mogli staviti da se rub podeblja i promjeni boju u neku drugu. Ima puno mogućnosti, no ipak je najbitnije da vidimo kako se to radi i kako funkcionira, a kasnije svatko radi prema svojim željama.

6.4 HTML i CSS

Osim CSS u Oracle Apexu se može koristiti i HTML, stoga ćemo prikazati i taj dio. Uz HTML koristit ćemo malo i CSS, pa da vidimo kako se mogu zajedno koristiti. Prvo ćemo napraviti *Blank page*, a onda unutar te stranice ćemo raditi sadržaj pomoću HTML-a. Točnije kreirat ćemo tablicu pomoću HTML a sa CSS-om ćemo urediti tablicu. Pošto smo kreirali *Blank page* onda moramo kreirati *Region* u kojem će se nalaziti tablica. Prije samog pisanja HTML-a prvo moramo posložiti postavke za *Region*. Prvo moramo staviti neko ime za *Region* a nakon toga ispod *Type* odabrati *Dynamic Content*. Ako nije automatski postavljeno onda moramo još *Language* postaviti na PL/SQL i onda u polje *PL/SQL Function Body returning a CLOB* stavljamo HTML i CSS kod. Sada ćemo vidjeti kako izgled naš HTML i CSS kod koji smo napisali.

```
1 DECLARE
2 tablica clob;
3 begin
4 tablica:='<!DOCTYPE html>
5 <html>
6 <style>
7 th{
8   border: 1px solid black;
9   text-align:center;
10  font-size: 20px;
11  background-color: darkred;
12 }
13
14 td {
15  border: 1px solid black;
16  text-align:center;
17 }
18 </style>
19
20 <body>
21 <h2>HTML tablica za Izvještaj o obradi</h2>
22
```

```

23 <table style="width:100%">
24     <tr>
25         <th rowspan="2">ID</th>
26         <th colspan="2">Datum</th>
27         <th colspan="2">Obrađeno</th>
28         <th rowspan="2">Radnik</th>
29         <th rowspan="2">Bilje</th>
30     </tr>
31     <tr>
32         <td>Datum Početka</td>
33         <td>Datum Završetka</td>
34         <td>Količina obrađenog</td>
35         <td>Mjerna jedinica</td>
36     </tr>';
37     for i in (SELECT * FROM IZVJEŠTAJ_O_OBRADI) LOOP
38         tablica:=tablica|| '<tr>
39                                 <td>' || i.ID_OBRADA || '</td>
40                                 <td>' || i.DATUM_POCETKA || '</td>
41                                 <td>' || i.DATUM_ZAVRSETKA || '</td>
42                                 <td>' || i.KOLICINA_OBRADENOG || '</td>
43                                 <td>' || i.ID_MJERNE_JEDINICE || '</td>
44                                 <td>' || i.ID_RADNIK || '</td>
45                                 <td>' || i.ID_TIP_BILJA || '</td>
46         </tr>';
47     END LOOP;
48     tablica:=tablica|| '</table>
49 </body>
50
51 </html>';
52 return tablica;
53 end;

```

Prve tri linije koda odnose se na PL/SQL kojim smo deklarirali varijablu *tablica* i onda smo u tu varijablu stavili naš HTML i CSS kod. Od 4. linije kreće HTML i CSS kod i odmah nakon toga možemo vidjeti da se do 18. linije nalazi CSS kod. Kao što vidimo, CSS je uključen interno unutar elementa *style*. Sve promjene koje želimo napraviti sa CSS-om trebamo napisati unutar tog elementa *style*. Osim toga u 23. liniji koda možemo vidjeti kako je ponovno pozvan *style* unutar linije odnosno unutar elementa *table*, što je još jedan od načina kako možemo uključiti CSS kod. U ovom primjeru nismo puno pisali CSS kod jer nam je bitno samo da pokažemo kako možemo na jednom mjestu dodati CSS i HTML kod zajedno. Nakon CSS-a i elementa *style* kreće HTML kod. Što se tiče HTML-a, nekome tko je radio s njim, ovaj kod bi trebao razumjeti u cijelosti. Ovo je standardni kod za kreiranje tablica gdje prevladavaju *th*, *tr* i *td* elementi, unutar glavnog elementa *table*. Od 37. do 45. linije koda radi se popunjavanje tablice dinamički. To znači da selektiramo sve elemente iz tablice *izvještaj_o_obradi* i sve što se nalazi u njoj će se nalaziti i u ovoj našoj HTML tablici. Ukoliko dođe do nekih promjena u tablici *izvještaj_o_obradi*, te će se promjene dogoditi i u našoj tablici. Tablicu smo mogli popuniti i statički, što znači da bi morali sami popunjavati svaku ćeliju sa vrijednostima koje želimo imati prikazane u tablici. Takvim načinom popunjavanja tablice bi nam trebalo nešto više linija koda, a ovako imamo samo 53 linije. Izgled tablice možete vidjeti dolje na slici 22.

HTML tablica

HTML tablica

HTML tablica za Izvještaj o obradi

ID	Datum		Obradeno		Radnik	Bilje
	Datum Početka	Datum Završetka	Količina obrađenog	Mjerna jedinica		
1	6/1/2022	6/1/2022	5	3	4	1
3	1/8/2022	1/8/2022	8	3	8	3
2	5/1/2022	5/1/2022	4	3	2	2
4	1/3/2022	1/3/2022	5	3	8	7
5	1/23/2022	1/23/2022	6300	1	3	2

Slika 22: HTML tablica

U tablici možemo vidjeti da su nam u stupcima *Mjerna jedinica*, *Radnik* i *Bilje* napisani brojevi zbog toga jer su takve vrijednosti unesene u tablici *Izvještaj_o_obradi*. Osim toga to su vanjski ključevi koji moraju biti brojevi i ako želimo znati što se nalazi pod određenim brojem, tada moramo otići u njihove tablice i vidjeti što se nalazi pod tim brojevima. Što znači, ako želimo znati na što se odnosi broj 3 u stupcu *Mjerna jedinica*, tada moramo otići u tablicu *Mjerna jedinica* i pogledati što piše pod brojem 3.

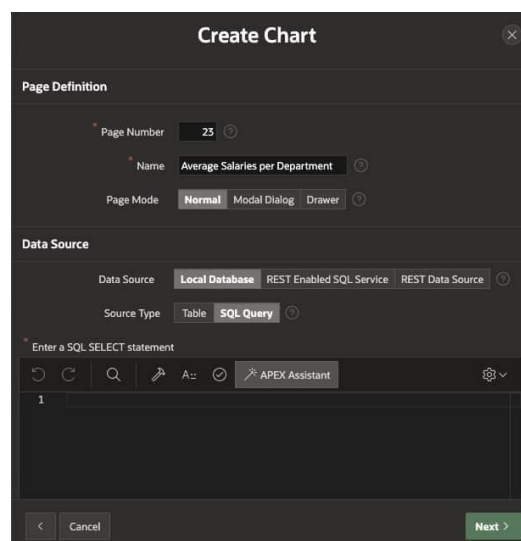
8. Umjetna inteligencija u Oracle Apexu

Danas je AI neizostavan u bilo kojoj aplikaciji ili platformi. Sve više ljudi se koristi njime jer znatno olakšava i ubrzava poslovanje i pojedine radnje.

Oracle Apex i veliki jezični modeli (LLM-ovi) jako brzo se razvijaju i omogućuju programerima da lakše obavljaju složene zadatke poput pisanja SQL upita korištenjem umjetne inteligencije (AI). *APEX Assistant* je alat unutar Oracle Apex platforme te omogućuje programerima da koriste prirodni jezik za generiranje SQL koda. To znatno ubrzava i olakšava proces pisanja SQL koda. Primjerice, umjesto ručnog pisanja SQL upita, programer može unijeti jednostavno pitanje kao što je „prikaži prosječne plaće zaposlenika po odjelima“ , a *APEX Assistant* automatski generira odgovarajući SQL upit. (Mueller, Patra, Carocari, Kareshk, Moghadam, 2023)

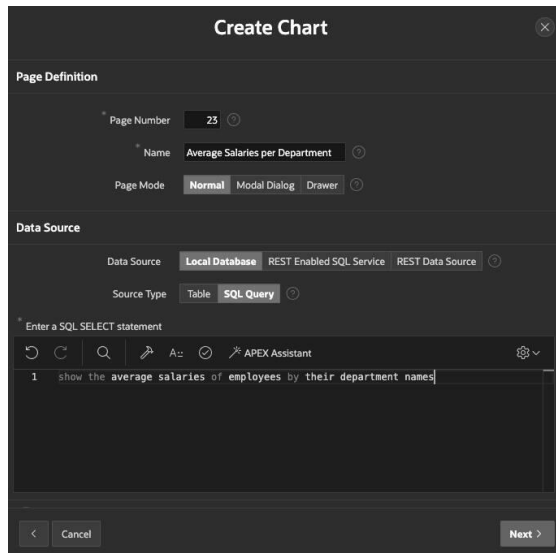
Ovaj alat se nalazi u različitim dijelovima *APEX-a* gdje je SQL potreban, poput izrade grafikona ili nekih novih stranica u aplikaciji. Umjesto ručnog unosa SQL upita za dohvaćanje podataka za grafikon, možemo tekstualno opisati zahtjev, a *APEX Assistant* će generirati potreban SQL. Na taj način, čak i složeni zahtjevi kao što su grupiranje podataka ili spajanje tablica mogu biti automatski generirani, što štedi vrijeme i smanjuje mogućnost grešaka. (Mueller i sur., 2023)

Sada ćemo vidjeti kako izgleda postupak korištenja alata *APEX Assistant* na primjeru kreiranja grafikona. Grafikon koji radimo će prikazivati prosječne plaće zaposlenika grupirane prema nazivu odjela zaposlenika.



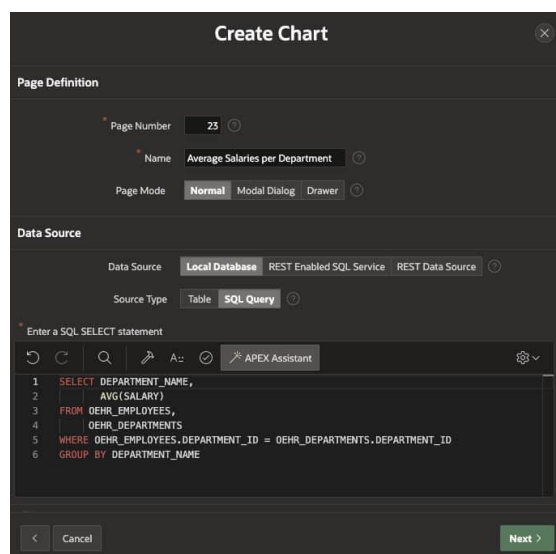
Slika 23: Kreiranje grafikona (Izvor: Ralf Mueller, Rhicheek Patra, Giulia Carocari, Moein Kareshk, Hesam Moghadam, 2023)

Na slici 23 vidimo standardnu stranicu za kreiranje grafikona kojem je postavljeno ime i kod *Source Type* je odabrano *SQL Query*. Sada u prozor za pisanje SQL koda treba napisati prirodnim jezikom što želimo dohvatiti.



Slika 24: Prirodni jezik za APEX Assistant (Izvor: *Ralf Mueller i sur., 2023*)

Na slici 24 vidimo kako je napisano „show the average salaries of employees by their department names“. Sada treba pritisnuti gumb *APEX Assistant* koji će taj tekst poslati u *Generative AI REST* i vraća rezultat u obliku Oracle SQL upita.



Slika 25: SQL upit od APEX Assistant (Izvor: *Ralf Mueller i sur., 2023*)

Na slici 25 možemo vidjeti SQL upit koji je vratio *APEX Assistant*.

APEX Assistant koristi NL2SQL infrastrukturu za obradu prirodnog jezika u SQL. Ta infrastruktura primjenjuje metodu *Retrieval Augmented Generation* (RAG) koja obogaćuje korisničke upite dodatnim informacijama iz baze podataka. RAG dohvaća kontekstualne informacije, poput imena tablica, njihovih atributa, primarnih i vanjskih ključeva, te ih koristi za stvaranje preciznijih SQL upita. Time se osigurava da generirani kod odgovara strukturi i podacima specifičnim za aplikaciju na kojoj programer radi. Osim toga RAG tehnologija omogućuje *APEX Assistantu* da bolje razumije složenije upite i da generira ispravne SQL upite koji uzimaju u obzir specifične tablice i odnose između njih. To je posebno korisno u situacijama gdje je potrebno raditi s privatnim podacima, stvarnim vremenskim kontekstom ili velikim količinama poslovnih dokumenata. (Mueller i sur., 2023)

Za *APEX Assistant* napravljena su mnoga testiranja kako bi se utvrdila njegova ispravnost. Testiranja su pokazala da *APEX Assistant* uspješno generira SQL kod za različite scenarije, poput izračuna prihoda, grupiranje podataka po godinama ili kreiranje složenih upita koji uključuju spajanje više tablica. Bez obzira na to, ipak se preporuča da iskusni programeri uvijek provjere generirani kod i po potrebi ga prilagode specifičnim potrebama. (Mueller i sur., 2023)

Sve u svemu, *APEX Assistant* predstavlja korisnog „suradnika“ za programere, pomažući im u bržem generiranju i testiranju SQL koda. Osim toga, istovremeno nudi mogućnost proširenja i fine prilagodbe kroz daljnji razvoj i optimizaciju LLM-ova. (Mueller i sur., 2023)

9. Zaključak

Iako platforme s malo koda programiraju u omjeru 80 na prema 20, odnosno 80% „drag and drop“ načinom a 20% ručno, čitajući ovaj rad rekli bih da je možda obrnuti omjer. Najviše zbog toga jer nigdje nismo vidjeli „drag and drop“, ali to nije bila ni tema ovoga rada. Tema je bila da prikažemo kako se radi ovih 20% odnosno kako se radi ovaj ručni dio. Prije nego smo pokazali ručni dio, morali smo se upoznati sa poduzećem i spomenuti neke važne tablice i dijagrame koji su bitni za izradu aplikacije. Nakon toga smo krenuli u ručni rad i prikazali većim dijelom kako se koristi CSS a manjim dijelom HTML. Pokazali smo gdje se oni i na koje načinu uključuju te smo vidjeli i promjene koje su se dogodile u aplikaciji prilikom pisanja određenog koda. Također smo vidjeli kako se na više mjesta može upisivati kod i kako se može svim elementima mijenjati izgled. Sve primjere smo potkrijepili sa slikama i sa kodom. Na samom kraju spomenuli smo i prikazali neizostavan AI. Te nakon svega toga možemo zaključiti da je Oracle Apex prije svega jednostavan za korištenje, a osim toga ima razne mogućnosti i širok spektar funkcija koje može obavljati. AI odnosno *APEX Assistant* podiže platformu na malo veću razinu jer time omogućuje još brže i jednostavnije kreiranje aplikacija. Na samom kraju uz malo ručnog programiranja Oracle Apex nudi funkcionalne i intuitivne aplikacije koje može svatko uz malo truda kreirati i koristiti.

10. Popis literature

Valentina Starčević (2021) *LOW-code/no-code – Što će vama program bez programiranja ?!* Preuzeto 15.9.2024 s <https://lidermedia.hr/tehnolo/low-code-no-code-sto-ce-vama-program-bez-programiranja-138725>

Ralf Mueller, Richeek Patra, Giulia Carocari, Moein Kareshk, Hesam Moghadam (2023) *Generative AI + Oracle APEX for Low-Code Application Development* Preuzeto 14.9.2024 s <https://blogs.oracle.com/apex/post/generative-ai-apex-1>

Velchandru M. (2024) *The Complete Oracle APEX Fundamentals Course (2024)* Preuzeto 20.8.2024 s <https://www.udemy.com/>

Muhammad Arshad (2024) *Oracle APEX Professional (2024)* Preuzeto 27.8.2024 s <https://www.udemy.com/>

11. Popis slika

Slika 1: 1. dio popisa procesa.....	4
Slika 2: 2. dio popisa procesa.....	5
Slika 3: PK matrica	6
Slika 4: ERA model	8
Slika 5: Početna stranica kupci	11
Slika 6: Opcija za uređivanje boja.....	12
Slika 7: Uređivanje ikone naslova	12
Slika 8: Slika i tekst kod uređivanja ikone.....	13
Slika 9: Uređivanje aplikacijske ikone	14
Slika 10: Teme	14
Slika 11: Tema crveno drvo	15
Slika 12: 1. Izgled aplikacije – crveno drvo	16
Slika 13: 2. izgled aplikacije – crveno drvo.....	16
Slika 14: Crna tema.....	17
Slika 15: Uređena crna tema	17
Slika 16: Crna tema sa CSS-om	18
Slika 17: Stranica za uređivanje aplikacije	20
Slika 18: CSS stranica	21
Slika 19: Kreiranje kupaca	24
Slika 20: Crveni okvir za unos	25
Slika 21: Zlatna pozadina za unos.....	26
Slika 22: HTML tablica.....	29
Slika 23: Kreiranje grafikona (Izvor: Ralf Mueller, Rlicheek Patra, Giulia Carocari, Moein Kareshk, Hesam Moghadam, 2023).....	30
Slika 24: Prirodni jezik za APEX Assistant (Izvor: Ralf Mueller i sur., 2023)	31
Slika 25: SQL upit od APEX Assistant (Izvor: Ralf Mueller i sur., 2023)	31