

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Luka Hrupec

**APLIKACIJA ZA TEMPORALNO MJERENJE
I OBRAČUN TROŠKOVA RADA -
TEMPORALNE I AKTIVNE BAZE
PODATAKA**

PROJEKT

TEORIJA BAZA PODATAKA

Varaždin, 2026.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Luka Hrupec

Matični broj: 1608003320016

Studij: Baze podataka i baze znanja

**APLIKACIJA ZA TEMPORALNO MJERENJE I OBRAČUN TROŠKOVA
RADA - TEMPORALNE I AKTIVNE BAZE PODATAKA**

PROJEKT

Mentor:

doc. dr. sc. Bogdan Okreša Đurić

Varaždin, siječanj 2026.

Luka Hrupec

Izjava o izvornosti

Izjavljujem da je ovaj projekt 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 potvrđio prihvaćanjem odredbi u sustavu FOI Radovi

Sažetak

U ovom projektu razvijena je aplikacija za temporalno mjerjenje rada i obračun troškova rada zaposlenika na projektima. Cilj aplikacije je omogućiti evidenciju radnih sesija različitih vrsta rada, automatski izračun trajanja rada i troška, te pružiti finansijski pregled projekata i angažmana zaposlenika. Sustav je implementiran korištenjem relacijske baze podataka PostgreSQL uz napredne mogućnosti poput okidača, funkcija, pohranjenih procedura i pogleda. Poseban naglasak stavljen je na aktivne i temporalne aspekte baze podataka, pri čemu se poslovna pravila za obračun rada provode na razini baze, a ne aplikacije. Time je osigurana konzistentnost podataka i centralizirana kontrola logike sustava. Korisničko sučelje aplikacije razvijeno je korištenjem programskog okvira Flask, a omogućuje unos zaposlenika, projekata i radnih sesija te pregled generiranih izvještaja. Sustav automatski razlikuje vrste rada poput redovnog, prekovremenog i noćnog rada, uz primjenu različitih koeficijenata obračuna. Rezultat projekta je funkcionalna aplikacija koja demonstrira primjenu aktivnih i temporalnih baza podataka u realnom poslovnom kontekstu, s jasnim razdvajanjem planiranih i stvarnih troškova te podrškom za donošenje upravljačkih odluka.

Ključne riječi: temporalne baze podataka; aktivne baze podataka; PostgreSQL; obračun troškova rada; radne sesije; projekti; okidači

Sadržaj

1. Uvod	1
2. Metode i tehnike rada	2
3. Razrada teme	3
3.1. Temporalne baze podataka	3
3.2. Aktivne baze podataka	3
3.3. Obračun rada i troškova u informacijskim sustavima	3
3.4. Primjena aktivnih i temporalnih koncepata u projektu	4
3.5. Model baze podataka	4
3.5.1. Konceptualni model	4
3.5.2. Logički model	5
3.5.3. Temporalni aspekti modela	6
3.5.4. Aktivni mehanizmi i ograničenja	7
3.6. ERA model baze podataka	8
4. Implementacija baze podataka	10
4.1. Implementacija relacijskih tablica	10
4.2. Temporalni aspekti implementacije	10
4.3. Funkcije i aktivni mehanizmi	11
4.4. Izračun planiranih i stvarnih troškova	12
4.5. Pogledi baze podataka	14
4.6. Pohranjene procedure	15
4.7. Sažetak implementacije	15
5. Opis aplikacije	16
5.1. Arhitektura aplikacije	16
5.2. Implementacija aplikacijskog sloja	16
5.2.1. Pregled i upravljanje projektima	16
5.2.2. Evidencija radnih sesija	18
5.2.3. Izvještaji	20
5.3. Sažetak aplikacije	22
6. Zaključak	23
Popis literature	24

Popis slika 25

Popis isječaka koda 26

1. Uvod

Praćenje rada zaposlenika i obračun troškova rada ključni su elementi upravljanja projektima u suvremenim organizacijama. Nepravilna ili neprecizna evidencija rada može dovesti do pogrešnih finansijskih procjena, prekoračenja budžeta i neoptimalne raspodjele resursa. Zbog toga je važno razviti sustav koji omogućuje točno, konzistentno i automatizirano praćenje rada kroz vrijeme.

Tema ovog projekta je razvoj aplikacije za temporalno mjerjenje rada i obračun troškova rada zaposlenika na projektima. Aplikacija omogućuje evidentiranje različitih vrsta rada, poput redovnog, prekovremenog i noćnog rada, uz automatski izračun trajanja rada i pripadajućeg troška. Poseban naglasak stavljen je na korištenje naprednih mogućnosti baze podataka kako bi se poslovna pravila provodila na razini baze, a ne aplikacijskog sloja.

Motivacija za odabir ove teme proizlazi iz praktične primjenjivosti sustava u stvarnim poslovnim okruženjima, kao i iz mogućnosti demonstracije aktivnih i temporalnih baza podataka. Projekt prikazuje kako se korištenjem okidača, funkcija i pogleda može izgraditi robusna aplikacija koja osigurava integritet podataka i pouzdane finansijske izvještaje.

2. Metode i tehnike rada

U razradi ovog projekta korišten je praktični razvojni pristup temeljen na analizi poslovnih zahtjeva i njihovoj implementaciji kroz bazu podataka i aplikacijski sloj. Najprije je definirana aplikacijska domena, identificirani su ključni entiteti i njihove relacije, te su određena poslovna pravila sustava.

Za implementaciju baze podataka korišten je sustav PostgreSQL, uz primjenu aktivnih i temporalnih koncepata baze podataka. Poslovna logika obračuna rada implementirana je pomoću okidača i funkcija, čime se osigurava da se izračuni izvršavaju automatski pri unosu podataka. Za agregaciju i prikaz podataka korišteni su pogledi nad bazom podataka.

Korisničko sučelje aplikacije razvijeno je korištenjem programskog okvira Flask u programskom jeziku Python. Aplikacija omogućuje unos i pregled podataka putem web sučelja, pri čemu se komunikacija s bazom podataka ostvaruje pomoću PostgreSQL upravljačkog programa.

Razvoj je proveden iterativno, uz kontinuirano testiranje funkcionalnosti sustava i provjeru ispravnosti izračuna troškova rada.

3. Razrada teme

3.1. Temporalne baze podataka

Temporalne baze podataka predstavljaju proširenje klasičnih relacijskih baza podataka s ciljem pohrane i obrade vremenski ovisnih podataka. Za razliku od standardnih baza, koje čuvaju samo trenutno stanje podataka, temporalne baze omogućuju praćenje promjena kroz vrijeme, uključujući povijest i valjanost podataka [1].

U kontekstu poslovnih informacijskih sustava, temporalni podaci često se odnose na razdoblja koliko vrijede određeni zapisi, poput trajanja projekata, razdoblja rada zaposlenika ili vremenskog okvira valjanosti ugovora. Takav pristup omogućuje analizu povijesnih podataka, usporedbu planiranih i stvarnih vrijednosti te donošenje informiranih odluka.

U ovom projektu temporalni aspekt baze podataka koristi se za evidentiranje radnih sesija zaposlenika, pri čemu svaka sesija ima definirani početak i završetak. Na temelju tih podataka automatski se izračunava trajanje rada i pripadajući trošak, uz poštivanje poslovnih pravila vezanih uz vrstu rada i vremensko razdoblje projekta.

3.2. Aktivne baze podataka

Aktivne baze podataka proširuju funkcionalnost tradicionalnih baza podataka uvođenjem mehanizama koji omogućuju automatsko reagiranje na promjene podataka. Takvi sustavi koriste pravila oblika događaj - uvjet - akcija, (engl. Event-Condition-Action, ECA), koja se u praksi najčešće implementiraju putem okidača [2].

Primjena aktivnih baza omogućuje provedbu poslovne logike izravno na razini baze podataka, čime se smanjuje ovisnost o aplikacijskom sloju. Na taj način osigurava se dosljednost podataka bez obzira na način njihova unosa te se pojednostavljuje razvoj aplikacije.

U ovom projektu aktivni koncepti baze koriste se za automatski izračun trajanja i troška radnih sesija. Prilikom unosa nove radne sesije, okidači pozivaju funkcije koje provode poslovna pravila, uključujući provjeru valjanosti datuma, obračun radnih dana i primjenu koeficijenata ovisno o vrsti rada.

3.3. Obračun rada i troškova u informacijskim sustavima

Obračun rada i pripadajućih troškova predstavlja važan dio upravljanja projektima i ljudskim resursima. Informacijski sustavi za evidenciju rada omogućuju praćenje angažmana zaposlenika, analizu produktivnosti te usporedbu planiranih i stvarnih troškova [3].

Uobičajeni pristupi obračunu rada temelje se na vremenskim intervalima rada i satnici zaposlenika, uz dodatne korekcije za prekovremen rad ili noćni rad. Automatizacija takvih izračuna smanjuje mogućnost pogreške i osigurava transparentnost financijskih podataka.

Projekt razvijen u ovom radu implementira navedene principe kroz funkcije baze podataka koje automatski izračunavaju trošak rada na temelju trajanja radne sesije, satnice zaposlenika i koeficijenta vrste rada.

3.4. Primjena aktivnih i temporalnih koncepata u projektu

Razvijeni sustav kombinira aktivne i temporalne koncepte baze podataka kako bi se ostvarila pouzdana i automatizirana evidencija rada. Temporalni aspekti omogućuju precizno praćenje trajanja radnih sesija, dok aktivni mehanizmi osiguravaju provedbu poslovnih pravila prilikom unosa podataka.

Izračun stvarnog troška rada temelji se na evidentiranim radnim sesijama, dok se planirani trošak računa na temelju trajanja projekta i predviđenog broja zaposlenika. Time se omogućuje jasna razlika između planiranih i ostvarenih vrijednosti, što je ključno za finansijsko praćenje projekata.

Korištenjem pogleda baze podataka generirani su finansijski izvještaji koji objedinjuju podatke o budžetu, planiranim i stvarnim troškovima te preostalom budžetu projekta, čime sustav pruža podršku upravljačkom odlučivanju.

3.5. Model baze podataka

Model baze podataka razvijen u ovom projektu osmišljen je kao relacijski model s nagnutim aktivnim i temporalnim obilježjima. Cilj modela je omogućiti pouzdanu evidenciju rada zaposlenika na projektima, automatizirani obračun troškova rada te pregled finansijskog stanja projekata kroz vrijeme. Prilikom dizajna modela posebna pažnja posvećena je normalizaciji podataka i razdvajanju planiranih i stvarnih vrijednosti troškova.

3.5.1. Konceptualni model

Konceptualni model baze podataka definira temeljnu strukturu sustava za evidenciju rada i obračun troškova rada zaposlenika na projektima. Model je osmišljen s ciljem jasnog mapiranja stvarnih poslovnih procesa na entitete baze podataka, pri čemu svaki entitet predstavlja jedan konkretni pojam. Konceptualni model obuhvaća sljedeće ključne entitete: *projekti*, *zaposlenici*, *radne sesije*, *vrste rada*, *članovi projekta* i *dodatni troškovi*.

Entitet *projekti* predstavlja osnovnu organizacijsku i poslovnu jedinicu sustava. Svaki projekt definiran je vremenskim rasponom trajanja, statusom te finansijskim parametrima koji omogućuju praćenje planiranih i stvarnih troškova. Projekt služi kao središnja točka sustava, budući da su na njega vezane radne sesije zaposlenika, pripadnost zaposlenika na projektu te dodatni troškovi koji nastaju tijekom njegova trajanja (prekovremeni rad). Na taj način projekt objedinjuje sve relevantne podatke potrebne za cjelovitu finansijsku i vremensku analizu.

Entitet *zaposlenici* modelira ljudske resurse koji sudjeluju u realizaciji projekata. Za

svakog zaposlenika evidentiraju se osnovni identifikacijski podaci te satnica, koja predstavlja temelj za izračun troškova rada. Zaposlenici mogu sudjelovati na više projekata te tijekom trajanja projekta ostvariti više radnih sesija, što omogućuje detaljno praćenje angažmana pojedinog zaposlenika kroz vrijeme.

Radni angažman zaposlenika na projektima modeliran je entitetom *radne sesije*. Svaka radna sesija povezuje jednog zaposlenika s jednim projektom u određenom vremenskom intervalu te predstavlja osnovnu temporalnu komponentu sustava. Radne sesije omogućuju precizno evidentiranje početka i završetka rada, na temelju čega se automatski izračunava trajanje rada (ovisno o vrsti rada) i pripadajući trošak. Na taj način radne sesije predstavljaju ključnu vezu između temporalnih podataka i finansijskih izračuna u sustavu.

Entitet *vrste rada* uveden je radi razlikovanja različitih oblika rada koji se mogu evidentirati u sustavu, poput redovnog, prekovremenog i noćnog rada. Svaka vrsta rada ima pripadajući koeficijent obračuna koji se primjenjuje prilikom izračuna troška rada. Ovim pristupom omogućena je fleksibilnost sustava i jednostavna prilagodba obračuna rada bez potrebe za izmjenama na aplikacijskom sloju.

Veza između zaposlenika i projekata ostvarena je putem entiteta *članovi projekta*, koji omogućuje modeliranje više-prema-više odnosa između navedenih entiteta. Ovaj entitet omogućuje evidentiranje sudjelovanja zaposlenika na projektima bez duplicitiranja podataka te osigurava jasnu strukturu odnosa unutar sustava. Pripadnost zaposlenika na projektu dodatno omogućuje praćenje angažmana zaposlenika na razini projekta, neovisno o pojedinačnim radnim sesijama.

Entitet *dodatni troškovi* služi za evidentiranje svih troškova koji nisu izravno povezani s radnim sesijama zaposlenika, poput troškova opreme, licenci ili drugih operativnih izdataka. Svaki dodatni trošak vezan je uz jedan projekt, čime se omogućuje cijelovit finansijski pregled projekta i točna usporedba planiranih i stvarnih troškova.

Na opisani način konceptualni model baze podataka osigurava jasnu i logičnu strukturu sustava, u kojoj su temporalni i finansijski aspekti rada zaposlenika integrirani u jedinstvenu cjelinu. Takav model predstavlja stabilnu osnovu za implementaciju logičkog i fizičkog modela baze podataka te učinkovitu provedbu poslovnih pravila sustava.

3.5.2. Logički model

Logički model baze podataka predstavlja formaliziranu implementaciju konceptualnog modela kroz skup relacijskih tablica. Model je realiziran u obliku normaliziranog relacijskog modela s jasno definiranim primarnim i stranim ključevima, čime se osigurava integritet podataka i konzistentnost odnosa među entitetima. Svaka tablica odgovara jednom entitetu iz konceptualnog modela te sadrži atribute koji su nužni za provedbu poslovne logike sustava.

Tablica *projekti* sadrži osnovne podatke o projektima, uključujući naziv projekta, datume početka i završetka, status projekta, predviđeni broj radnika te budžet buffer za nepredviđene troškove. Dodatno, u tablici se pohranjuje i planirani trošak projekta, koji se ne unosi ručno, već se automatski izračunava u trenutku kreiranja projekta. Izračun planiranog troška provodi

se pomoću pohranjene funkcije i okidača baze podataka, čime se osigurava da je planirani trošak uvijek u skladu s vremenskim rasponom projekta i predviđenim brojem radnika. Na taj način izbjegnute su potencijalne pogreške ručnog unosa i osigurana je dosljednost planiranih vrijednosti.

Tablica *zaposlenici* služi za pohranu podataka o zaposlenicima koji sudjeluju u radu na projektima. U tablici se pohranjuju osnovni identifikacijski podaci zaposlenika te satnica, koja predstavlja ključni ulazni parametar za izračun troška rada. Satnica zaposlenika koristi se u svim izračunima troškova radnih sesija, što omogućuje precizno financijsko praćenje rada pojedinih zaposlenika na projektima.

Evidencija stvarnog rada zaposlenika ostvarena je tablicom *radne_sesije*, koja predstavlja središnji element logičkog modela, vezano za temporalne podatke. Svaka radna sesija povezana je s jednim zaposlenikom, jednim projektom i jednom vrstom rada putem stranih ključeva prema tablicama *zaposlenici*, *projekti* i *vrste_rada*. Tablica sadrži podatke o početku i završetku rada, dok se trajanje rada i pripadajući trošak ne pohranjuju kao ulazni podaci, već se automatski izračunavaju pomoću funkcija i okidača baze podataka. Ovakav pristup osigurava da su svi izračuni centralizirani i konzistentni, neovisno o aplikacijskom sloju.

Tablica *vrste_rada* koristi se za definiranje različitih tipova rada koji se mogu evidentirati u sustavu. Svaka vrsta rada ima pridruženi koeficijent obračuna koji se primjenjuje prilikom izračuna troška radne sesije. Time je omogućeno razlikovanje redovnog, prekovremenog i noćnog rada, kao i jednostavno proširenje sustava novim vrstama rada bez potrebe za izmjenom postojeće strukture tablica ili aplikacijskog koda.

Veza između zaposlenika i projekata dodatno je modelirana tablicom *clanovi_projekta*, koja omogućuje ostvarivanje više-prema-više odnosa između ovih entiteta. Tablica koristi složeni primarni ključ sastavljen od identifikatora zaposlenika i projekta, čime se sprječava dupliranje zapisa o članstvu. Zaposlenici se u ovu tablicu ne dodaju ručno, već se automatski evidentiraju prilikom unosa prve radne sesije za određeni projekt, korištenjem aktivnih mehanizama baze podataka. Ovim pristupom osigurano je da članstvo zaposlenika na projektu uvijek odražava stvarni rad evidentiran u sustavu.

Opisani logički model omogućuje jasno razdvajanje planiranih i stvarnih podataka, precizno praćenje rada zaposlenika kroz vrijeme te pouzdanu provedbu financijskih izračuna. Struktura modela osigurava stabilnu osnovu za implementaciju aktivnih i temporalnih mehanizama baze podataka te učinkovitu integraciju s aplikacijskim slojem sustava.

3.5.3. Temporalni aspekti modela

Temporalni aspekti baze podataka ostvareni su kroz korištenje vremenskih atributa u tablicama projekata i radnih sesija. Svaki projekt ima definiran datum početka i završetka, dok svaka radna sesija sadrži vremenski raspon u kojem je rad obavljen. Na temelju tih podataka omogućeno je precizno praćenje trajanja rada i vremenskog okvira valjanosti podataka.

Posebna pažnja posvećena je kontroli vremenskih ograničenja, pri čemu se provjerava da se radne sesije ne mogu evidentirati izvan vremenskog raspona projekta. Ova pravila imple-

mentirana su na razini baze podataka pomoću okidača i pohranjenih funkcija, čime se osigurava konzistentnost temporalnih podataka.

3.5.4. Aktivni mehanizmi i ograničenja

Aktivni mehanizmi baze podataka u ovom projektu implementirani su korištenjem okidača i pohranjenih funkcija, čime je omogućena automatizirana provedba poslovnih pravila sustava. Okidači se automatski aktiviraju kao odgovor na određene događaje u bazi podataka, prvenstveno prilikom umetanja ili ažuriranja zapisa u ključnim tablicama sustava, poput tablica *projekti* i *radne_sesije*. Na taj način ostvarena je primjena koncepta aktivnih baza podataka temeljenog na pravilima događaj-uvjet-akcija (ECA).

Prilikom unosa novog projekta aktivira se okidač koji poziva pohranjenu funkciju za izračun planiranog troška projekta. Funkcija koristi vremenski raspon projekta i predviđeni broj radnika kako bi automatski izračunala planirani trošak rada. Ovakav pristup osigurava da je planirani trošak uvijek konzistentan s osnovnim parametrima projekta te uklanja potrebu za ručnim unosom i naknadnim korekcijama.

Kod unosa radnih sesija aktivni mehanizmi baze podataka imaju ključnu ulogu u provedbi poslovne logike obračuna rada. Okidači povezani s tablicom *radne_sesije* pozivaju pohranjene funkcije koje automatski izračunavaju trajanje rada i pripadajući trošak na temelju vremenskog raspona rada, satnice zaposlenika i koeficijenta vrste rada. Time se osigurava da se svi izračuni provode na jedinstven i kontroliran način, neovisno o aplikacijskom sloju ili načinu unosa podataka.

Osim izračuna troškova, aktivni mehanizmi koriste se i za provedbu poslovnih pravila vezanih uz valjanost unosa podataka. Implementirane su funkcije koje onemogućuju evidentiranje radnih sesija za projekte čiji je status označen kao završen, čime se sprječavaju nelogični i poslovno nedopušteni zapisi. Ova pravila provode se izravno u bazi podataka, što osigurava njihovu primjenu čak i u slučaju izravnog pristupa bazi podataka izvan aplikacije.

Dodatno, aktivni mehanizmi koriste se za automatizaciju odnosa između zaposlenika i projekata. Prilikom unosa prve radne sesije zaposlenika na određenom projektu, automatski se evidentira njegovo članstvo u tablici *clanovi_projekta*. Na taj način osigurano je da članstvo zaposlenika na projektu uvijek odražava stvarni rad evidentiran u sustavu, bez potrebe za ručnim održavanjem povezanih zapisa.

Uz aktivne mehanizme, u modelu su definirana i ograničenja integriteta podataka koja dodatno osiguravaju ispravnost pohranjenih informacija. Ograničenja integriteta uključuju provjere vremenskih raspona, poput osiguranja da kraj radne sesije ne može prethoditi njezinu početku, kao i ograničenja nad vrijednostima atributa, primjerice pozitivne vrijednosti satnice i koeficijenata obračuna. Takva ograničenja sprječavaju unos nelogičnih ili nedopuštenih podataka već na razini baze podataka.

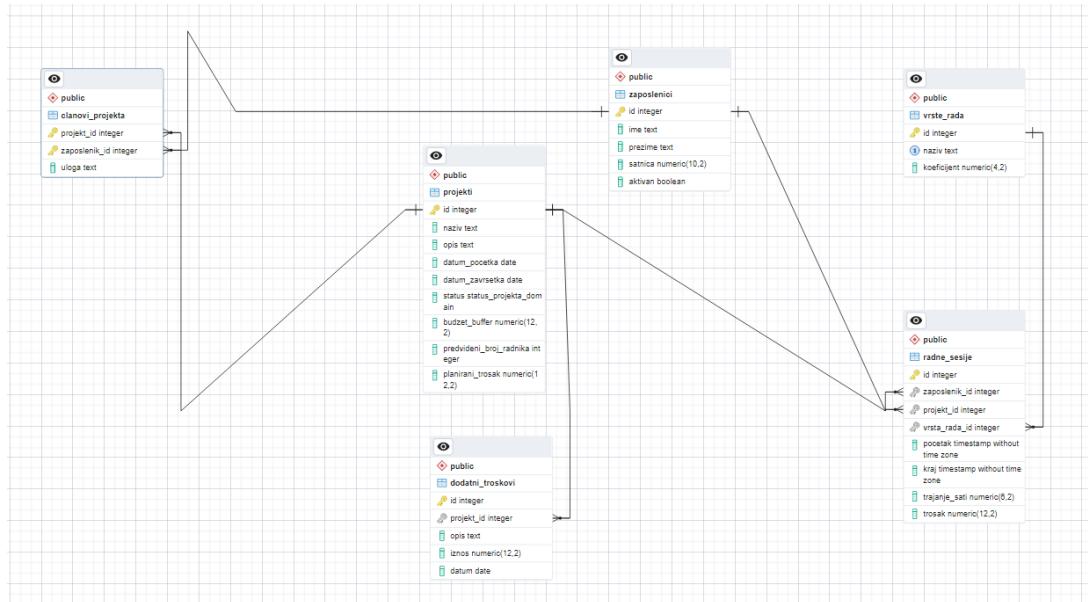
Kombinacijom aktivnih mehanizama i ograničenja integriteta postignuta je visoka razina pouzdanosti i konzistentnosti podataka. Poslovna pravila sustava centralizirana su u bazi podataka, čime je smanjena ovisnost o aplikacijskom sloju i povećana robusnost cijelog sustava.

Ovakav pristup jasno demonstrira prednosti aktivnih baza podataka u kontekstu informacijskih sustava za evidenciju rada i obračun troškova.

3.6. ERA model baze podataka

Model baze podataka razvijen je s ciljem podrške web aplikaciji za evidenciju rada zaposlenika i praćenje troškova po projektima. Baza podataka implementirana je u sustavu PostgreSQL te koristi relacijski model uz primjenu aktivnih elemenata baze podataka poput funkcija, okidača i pogleda.

Na slici je prikazan ERA dijagram modela baze podataka izrađen korištenjem alata ERD Tool u sklopu pgAdmin 4 razvojnog okruženja. Dijagram prikazuje strukturu baze podataka, definirane entitete, njihove attribute te odnose među entitetima.



Slika 1: ERA model

Model baze podataka sastoji se od šest osnovnih entiteta: *projekti*, *zaposlenici*, *radne_sesije*, *vrste_rada*, *clanovi_projekta* i *dodatni_troskovi*. Svaki entitet ima jasno definiranu ulogu u sustavu te je povezan s ostalim entitetima putem primarnih i vanjskih ključeva.

Entitet *projekti* predstavlja osnovnu poslovnu cjelinu sustava te sadrži informacije o nazivu projekta, opisu, datumu početka i završetka, statusu projekta, budžetu bufferu, predviđenom broju radnika i planiranom trošku. Primarni ključ entiteta je atribut *id*. Entitet *projekti* povezan je s entitetima *radne_sesije* i *dodatni_troskovi* u odnosu jedan prema više, dok se veza sa zaposlenicima ostvaruje putem poveznog entiteta *clanovi_projekta*.

Entitet *zaposlenici* sadrži podatke o zaposlenicima koji sudjeluju u radu na projektima, uključujući ime, prezime, satnicu i status aktivnosti. Primarni ključ entiteta je atribut *id*. Zaposlenici mogu sudjelovati na više projekata te imati više evidentiranih radnih sesija.

Entitet *vrste_rada* definira tipove rada koji se mogu evidentirati u sustavu, poput redov-

nog, prekovremenog i noćnog rada. Svaka vrsta rada ima pripadajući koeficijent koji se koristi pri izračunu troška rada. Entitet je povezan s entitetom *radne_sesije* u odnosu jedan-prema-više.

Entitet *radne_sesije* služi za evidenciju konkretnih radnih aktivnosti zaposlenika na projektima. Svaka radna sesija povezana je s jednim zaposlenikom, jednim projektom i jednom vrstom rada. Entitet sadrži podatke o početku i završetku rada, trajanju u satima te izračunatom trošku. Izračun trajanja i troška provodi se automatski korištenjem funkcija i okidača baze podataka.

Entitet *clanovi_projekta* predstavlja povezni entitet koji omogućuje ostvarivanje više-prema-više odnosa između entiteta *projekti* i *zaposlenici*. Na taj način definira se koji zaposlenici sudjeluju na određenom projektu te njihova uloga unutar projekta.

Entitet *dodatni_troskovi* koristi se za evidentiranje troškova koji nisu izravno povezani s radnim sesijama, kao što su troškovi opreme ili ostali operativni troškovi. Svaki dodatni trošak vezan je uz jedan projekt, čime se omogućuje cjelovit finansijski pregled projekta.

Definirani ERA model omogućuje precizno praćenje rada zaposlenika, izračun planiranih i stvarnih troškova projekata te generiranje finansijskih izvještaja. Struktura baze podataka osigurava konzistentnost podataka i predstavlja stabilnu osnovu za daljnje proširenje sustava.

4. Implementacija baze podataka

Implementacija baze podataka predstavlja središnji dio razvijenog sustava za temporalno mjerjenje rada i obračun troškova rada zaposlenika na projektima. Baza podataka implementirana je u sustavu PostgreSQL te koristi relacijski model proširen aktivnim i temporalnim mehanizmima. Poslovna pravila sustava implementirana su na razini baze podataka korištenjem ograničenja, funkcija, okidača, pohranjenih procedura i pogleda, čime se osigurava konzistentnost podataka neovisno o aplikacijskom sloju **postgresql2023**.

4.1. Implementacija relacijskih tablica

Osnovu baze podataka čine relacijske tablice koje modeliraju ključne entitete sustava. Tablica *zaposlenici* pohranjuje podatke o zaposlenicima, uključujući ime, prezime, satnicu i status aktivnosti. Satnica je definirana kao numerička vrijednost s ograničenjem koje sprječava unos negativnih ili nultih vrijednosti, čime se osigurava ispravnost financijskih izračuna.

Tablica *projekti* služi za pohranu osnovnih informacija o projektima, uključujući naziv, opis, datume početka i završetka, status projekta te budžet buffer za nepredviđene troškove. Implementirano je ograničenje koje osigurava da datum završetka ne može biti raniji od datuma početka projekta. Status projekta ograničen je domenom dozvoljenih vrijednosti, čime se sprječava unos nedozvoljenih stanja.

Veza između zaposlenika i projekata ostvarena je tablicom *clanovi_projekta*, koja implementira više-prema-više odnos između navedenih entiteta. Tablica koristi složeni primarni ključ sastavljen od identifikatora projekta i zaposlenika, čime se sprječava duplicitiranje zapisa o članstvu.

Tablica *vrste_rada* definira tipove rada koji se mogu evidentirati u sustavu, poput redovnog, prekovremenog i noćnog rada. Svaka vrsta rada ima pripadajući koeficijent koji se koristi u izračunu troška rada.

Evidencija stvarnog rada zaposlenika ostvarena je tablicom *radne_sesije*. Svaka radna sesija povezana je s jednim zaposlenikom, jednim projektom i jednom vrstom rada, te sadrži vremenski raspon rada, izračunato trajanje u satima i pripadajući trošak.

Dodatni troškovi koji nisu izravno povezani s radnim sesijama pohranjuju se u tablici *dodatni_troskovi*. Na taj način omogućeno je cijelovito financijsko praćenje projekta, uključujući operativne i materijalne troškove.

4.2. Temporalni aspekti implementacije

Temporalni aspekti baze podataka implementirani su korištenjem vremenskih atributa u tablicama *projekti* i *radne_sesije*. Projekti imaju jasno definiran vremenski interval važenja, dok svaka radna sesija sadrži početak i kraj rada. Implementirano je ograničenje koje osigurava da kraj radne sesije mora biti nakon njezina početka.

Dodatno, kroz funkcije i okidače provode se kontrole kojima se sprječava evidentiranje rada izvan vremenskog raspona projekta. Time se osigurava dosljednost temporalnih podataka i sprječavaju nelogični zapisi.

4.3. Funkcije i aktivni mehanizmi

Poslovna logika sustava implementirana je korištenjem funkcija pisanih u programskom jeziku PL/pgSQL. Ključna funkcija *izracunaj_trosak_rada* automatski izračunava trajanje radne sesije u satima te pripadajući trošak rada na temelju satnice zaposlenika i koeficijenta vrste rada. Funkcija se aktivira putem okidača prije umetanja ili ažuriranja zapisa u tablici *radne_sesije*.

```
1 CREATE OR REPLACE FUNCTION izracunaj_trosak_rada()
2 RETURNS TRIGGER AS $$ 
3 DECLARE
4     v_satnica NUMERIC(10,2);
5     v_koeficijent NUMERIC(4,2);
6 BEGIN
7     SELECT satnica
8         INTO v_satnica
9     FROM zaposlenici
10    WHERE id = NEW.zaposlenik_id;
11
12    SELECT koeficijent
13        INTO v_koeficijent
14    FROM vrste_rada
15    WHERE id = NEW.vrsta_rada_id;
16
17    NEW.trajanje_sati :=
18        EXTRACT(EPOCH FROM (NEW.kraj - NEW.pocetak)) / 3600;
19
20    NEW.trosak :=
21        NEW.trajanje_sati * v_satnica * v_koeficijent;
22
23    RETURN NEW;
24 END;
25 $$ LANGUAGE plpgsql;
```

Isječak koda 1: Funkcija za automatski izračun trajanja i troška radne sesije

Funkcija *zabrani_rad_za_zavrsen_projekt* implementira poslovno pravilo kojim se one-moguće unos novih radnih sesija za projekte čiji je status označen kao završen. Ova funkcija se automatski izvršava putem okidača prije umetanja zapisa, čime se osigurava integritet podataka.

```

1 CREATE OR REPLACE FUNCTION zabrani_rad_za_zavrseren_projekt()
2 RETURNS TRIGGER AS $$ 
3 DECLARE
4     v_status TEXT;
5 BEGIN
6     SELECT status INTO v_status
7     FROM projekti
8     WHERE id = NEW.projekt_id;
9
10    IF v_status = 'zavrseren' THEN
11        RAISE EXCEPTION
12            'Nije dopusteno evidentirati rad za zavrseren projekt.';
13    END IF;
14
15    RETURN NEW;
16
17 $$ LANGUAGE plpgsql;

```

Isječak koda 2: Funkcija za zabranu unosa rada na završenom projektu

Automatsko dodavanje zaposlenika u tablicu *clanovi_projekta* implementirano je funkcijom *dodaj_clana_projekta_ako_ne_postoji*. Na taj način zaposlenik postaje član projekta čim evidentira prvu radnu sesiju, bez potrebe za ručnim unosom.

4.4. Izračun planiranih i stvarnih troškova

Planirani trošak projekta izračunava se automatski prilikom unosa projekta korištenjem funkcije *izracunaj_planirani_trosak*. Izračun se temelji na broju radnih dana unutar trajanja projekta, predviđenom broju radnika i prosječnoj satnici zaposlenika. Funkcija *izracunaj_radne_dane* koristi se za izračun broja radnih dana, pri čemu se uzimaju u obzir samo radni dani od ponedjeljka do petka.

```

1 CREATE OR REPLACE FUNCTION izracunaj_planirani_trosak(
2     p_datum_pocetka DATE,
3     p_datum_zavrsetka DATE,
4     p_broj_radnika INT
5 )
6 RETURNS NUMERIC AS $$ 
7 DECLARE
8     radni_dani INT;
9     prosjecna_satnica NUMERIC;
10 BEGIN
11     radni_dani :=
12         izracunaj_radne_dane(
13             p_datum_pocetka,
14             p_datum_zavrsetka
15         );
16
17     SELECT AVG(satnica)
18     INTO prosjecna_satnica
19     FROM zaposlenici;
20
21     RETURN radni_dani * 8 *
22             p_broj_radnika *
23             COALESCE(prosjecna_satnica, 0);
24 END;
25 $$ LANGUAGE plpgsql;

```

Isječak koda 3: Funkcija za izračun planiranog troška projekta

Stvarni trošak projekta predstavlja ukupni iznos svih evidentiranih troškova nastalih tijekom trajanja projekta te se sastoji od dva osnovna dijela: troškova rada zaposlenika i dodatnih troškova projekta. Troškovi rada generiraju se automatski na temelju radnih sesija zaposlenika, pri čemu se uzimaju u obzir trajanje rada, satnica zaposlenika i koeficijent vrste rada.

Dodatni troškovi obuhvaćaju sve troškove koji nisu izravno vezani uz radno vrijeme zaposlenika, poput troškova opreme, licenci ili drugih operativnih izdataka. Ovi troškovi evidentiraju se zasebno u tablici *dodatni_troskovi* te se povezuju s odgovarajućim projektom.

Agregacija stvarnog troška projekta provodi se isključivo na razini baze podataka koristenjem pogleda *v_financijski_pregled_projekta*, koji objedinjuje troškove rada i dodatne troškove u jedinstveni iznos stvarnog troška. Aplikacijski sloj ne provodi nikakve dodatne izračune, već isključivo prikazuje rezultate dobivene iz baze podataka, čime je osigurana centralizacija poslovne logike i konzistentnost financijskih podataka.

```

1 CREATE VIEW v_financijski_pregled_projekta AS
2 SELECT
3     p.id AS projekt_id,
4     p.naziv AS naziv_projekta,
5     p.budzet_buffer,
6     p.planirani_trosak,
7
8     ROUND (
9         COALESCE(r.ukupni_trosak_rada, 0) +
10        COALESCE(d.ukupni_dodatni_troskovi, 0),
11        2
12    ) AS stvarni_trosak,
13
14     ROUND (
15         p.planirani_trosak + p.budzet_buffer
16         -
17         (
18             COALESCE(r.ukupni_trosak_rada, 0) +
19             COALESCE(d.ukupni_dodatni_troskovi, 0)
20         ),
21         2
22     ) AS preostali_budzet
23 FROM projekti p
24 LEFT JOIN (
25     SELECT projekt_id, SUM(trosak) AS ukupni_trosak_rada
26     FROM radne_sesije
27     GROUP BY projekt_id
28 ) r ON r.projekt_id = p.id
29 LEFT JOIN (
30     SELECT projekt_id, SUM(iznos) AS ukupni_dodatni_troskovi
31     FROM dodatni_troskovi
32     GROUP BY projekt_id
33 ) d ON d.projekt_id = p.id;

```

Isječak koda 4: Pogled *v_financijski_pregled_projekta*

4.5. Pogledi baze podataka

Za potrebe izvještavanja implementirani su pogledi koji omogućuju jednostavan i pregledan uvid u financijsko stanje projekata. Pogled *v_financijski_pregled_projekta* objedinjuje planirani trošak, stvarni trošak i preostali budžet projekta. Dodatni pogledi omogućuju pregled troška rada po projektu i po zaposleniku.

```

1 CREATE VIEW v_financijski_pregled_projekta AS
2 SELECT
3     p.id AS projekt_id,
4     p.naziv AS naziv_projekta,
5     p.budzet_buffer,
6     p.planirani_trosak,
7     COALESCE(SUM(rs.trosak), 0)
8         + COALESCE(SUM(dt.iznos), 0)
9             AS stvarni_trosak,
10            (p.planirani_trosak + p.budzet_buffer)
11            -
12            (COALESCE(SUM(rs.trosak), 0)
13             + COALESCE(SUM(dt.iznos), 0))
14                 AS preostali_budzet
15 FROM projekti p
16 LEFT JOIN radne_sesije rs ON rs.projekt_id = p.id
17 LEFT JOIN dodatni_troskovi dt ON dt.projekt_id = p.id
18 GROUP BY
19     p.id, p.naziv,
20     p.budzet_buffer,
21     p.planirani_trosak;

```

Isječak koda 5: Pogled za financijski pregled projekta

Korištenjem pogleda smanjuje se složenost aplikacijskog sloja, jer se složeni izračuni provode izravno u bazi podataka.

4.6. Pohranjene procedure

Za upravljanje statusom projekta implementirana je pohranjena procedura *promjeni_status_projekta*, koja omogućuje promjenu statusa projekta na kontroliran način. Time se dodatno osigurava centralizirano upravljanje poslovnom logikom.

4.7. Sažetak implementacije

Implementacija baze podataka u ovom projektu demonstrira primjenu aktivnih i temporalnih koncepata baze podataka u realnom poslovnom kontekstu. Poslovna pravila sustava implementirana su na razini baze podataka korištenjem funkcija, okidača i ograničenja, čime je osigurana konzistentnost podataka, automatizacija izračuna i pouzdano financijsko praćenje projekata.

5. Opis aplikacije

U sklopu projekta razvijena je web aplikacija za evidenciju rada zaposlenika i praćenje troškova projekata. Aplikacija predstavlja korisnički sloj sustava te omogućuje unos i pregled podataka, dok se sva poslovna logika i izračuni provode na razini baze podataka. Aplikacija je implementirana korištenjem programskog okvira Flask u programskom jeziku Python te koristi PostgreSQL bazu podataka.

5.1. Arhitektura aplikacije

Aplikacija je implementirana prema klijent–poslužitelj arhitekturi. Korisničko sučelje realizirano je pomoću HTML predložaka, dok Flask aplikacija služi kao posrednik između korisnika i baze podataka. Komunikacija s bazom podataka ostvarena je korištenjem biblioteke *psycopg2*.

Flask aplikacija ne implementira poslovnu logiku obračuna rada, već se ona u potpunosti oslanja na aktivne i temporalne mehanizme baze podataka. Time je osigurana konzistentnost podataka neovisno o načinu pristupa sustavu.

5.2. Implementacija aplikacijskog sloja

Glavna aplikacijska logika implementirana je u datoteci `app.py`. Povezivanje s bazom podataka ostvareno je pomoću pomoćne funkcije koja uspostavlja vezu prema PostgreSQL poslužitelju.

```
1 def get_db_connection():
2     return psycopg2.connect(
3         host="localhost",
4         port=5434,
5         database="TBP_projekt",
6         user="postgres",
7         password="postgres"
8     )
```

Isječak koda 6: Funkcija za uspostavu veze s bazom podataka

Svaka ruta u aplikaciji koristi navedenu funkciju za dohvati ili izmjenu podataka u bazi, nakon čega se veza zatvara. Time se osigurava jednostavna i čitljiva struktura aplikacijskog koda.

5.2.1. Pregled i upravljanje projektima

Početna stranica aplikacije prikazuje popis svih projekata pohranjenih u bazi podataka. Korisniku je omogućen pregled osnovnih podataka o projektu te promjena statusa projekta

putem pohranjene procedure baze podataka.

```
1 @app.route("/")
2 def projekti():
3     conn = get_db_connection()
4     cur = conn.cursor()
5
6     cur.execute("""
7         SELECT id, naziv, datum_pocetka, datum_zavrsetka, status
8         FROM projekti
9         ORDER BY id;
10    """)
11    projekti = cur.fetchall()
12
13    cur.close()
14    conn.close()
15
16    return render_template("projekti.html", projekti=projekti)
```

Isječak koda 7: Ruta za prikaz popisa projekata

Promjena statusa projekta provodi se pozivom pohranjene procedure, čime se dodatno naglašava centralizacija poslovne logike na razini baze podataka.

Popis projekata					
ID	Naziv	Početak	Završetak	Status	Promjena statusa
1	Web aplikacija za evidenciju rada	2026-01-08	2026-02-27	u_izradi	<input type="button" value="planiran"/> <input type="button" value="Spremi"/>
2	Interni HR sustav	2026-03-02	2026-03-20	planiran	<input type="button" value="planiran"/> <input type="button" value="Spremi"/>
3	Migracija baze podataka	2026-01-15	2026-02-13	planiran	<input type="button" value="planiran"/> <input type="button" value="Spremi"/>

Slika 2: Korisničko sučelje za pregled i upravljanje projektima

Osim pregleda postojećih projekata, aplikacija omogućuje dodavanje novih projekata putem web obrasca. Korisnik unosi osnovne podatke o projektu, uključujući naziv, vremenski raspon, predviđeni broj radnika, status projekta i budžet buffer. Prilikom unosa projekta automatski se izračunava planirani trošak projekta korištenjem funkcija baze podataka, čime se osigurava konzistentnost planiranih vrijednosti.

Osim unosa osnovnih podataka o projektu, aplikacija omogućuje i evidentiranje dodatnih troškova koji nisu izravno povezani s radnim sesijama zaposlenika.

Prilikom unosa dodatnog troška korisnik odabire projekt na koji se trošak odnosi te unosi opis i iznos troška, dok se datum evidentiranja automatski postavlja na dan unosa. Uneseni

dodatni troškovi pohranjuju se u tablicu *dodatni_troskovi* baze podataka te se ne obrađuju dodatno na razini aplikacije.

Evidentirani dodatni troškovi automatski se uključuju u izračun stvarnog troška projekta putem pogleda baze podataka, čime se osigurava da financijski pregled projekta u svakom trenutku odražava stvarno stanje, uključujući i operativne troškove koji nisu povezani s radnim vremenom zaposlenika.

The screenshot displays two side-by-side forms. The left form is titled 'Upravljanje projektima i dodatnim troškovima' and contains fields for adding a new project: Naziv projekta (Project Name), Datum početka (Start Date), Datum završetka (End Date), Broj potrebnih radnika (Number of workers), Status projekta (Project Status), and Budžet buffer (Budget buffer). A 'Dodaj projekt' (Add project) button is at the bottom. The right form is titled 'Dodaj dodatni trošak' and contains fields for adding an additional expense: Projekt (Project dropdown), Opis troška (Description of expense), Iznos (Amount), and a 'Dodaj trošak' (Add expense) button.

Slika 3: Obrazci za dodavanje novog projekta i dodatnih troškova

5.2.2. Evidencija radnih sesija

Aplikacija omogućuje unos radnih sesija zaposlenika na projektima uz odabir vrste rada i vremenskog razdoblja. Prilikom unosa podataka poziva se SQL naredba za umetanje zapisa u tablicu *radne_sesije*, dok se svi izračuni trajanja i troška rada provode automatski pomoću okidača baze podataka.

```

1 @app.route("/dodaj_radnu_sesiju", methods=["POST"])
2 def dodaj_radnu_sesiju():
3     conn = get_db_connection()
4     cur = conn.cursor()
5
6     cur.execute("""
7         INSERT INTO radne_sesije
8             (zaposlenik_id, projekt_id, vrsta_rada_id, pocetak, kraj)
9             VALUES (%s, %s, %s, %s, %s)
10        """, (
11            request.form["zaposlenik_id"],
12            request.form["projekt_id"],
13            request.form["vrsta_rada_id"],
14            request.form["pocetak"],
15            request.form["kraj"]
16        ))
17
18     conn.commit()
19     cur.close()
20     conn.close()
21
22     return redirect(url_for("radne_sesije"))

```

Isječak koda 8: Ruta za unos nove radne sesije

Korisničko sučelje dodatno koristi JavaScript za dohvrat vremenskog raspona projekta te automatsko postavljanje datuma kod redovnog rada, čime se smanjuje mogućnost pogrešnog unosa podataka.

Unos radne sesije

Zaposlenik:

Projekt:

Vrsta rada:

Početak rada:

Kraj rada:

Spremi radnu sesiju

Slika 4: Sučelje za unos radnih sesija

Postojeće radne sesije

Zaposlenik	Projekt	Vrsta rada	Trajanje (h)	Trošak
Pero Petrović	Migracija baze podataka	redovni	176.00	1672.00
Petra Horvat	Migracija baze podataka	redovni	176.00	2288.00
Pero Petrović	Interni HR sustav	redovni	120.00	1140.00
Ana Kovač	Interni HR sustav	prekovremeni	32.00	696.00
Ivan Ivic	Web aplikacija za evidenciju rada	prekovremeni	0.00	0.00
Marko Marić	Web aplikacija za evidenciju rada	redovni	288.00	3456.00
Ana Kovač	Web aplikacija za evidenciju rada	redovni	288.00	4176.00
Ivan Ivic	Web aplikacija za evidenciju rada	redovni	288.00	2880.00

Slika 5: Pregled postojećih sesija

5.2.3. Izvještaji

Za potrebe analize i upravljanja projektima implementiran je modul za izvještavanje. Izvještaji se temelje na pogledima baze podataka koji objedinjuju planirane i stvarne troškove rada.

```

1 @app.route("/izvjestaji")
2 def izvjestaji():
3     conn = get_db_connection()
4     cur = conn.cursor()
5
6     cur.execute("SELECT * FROM v_financijski_pregled_projekta;")
7     financijski_pregled = cur.fetchall()
8
9     cur.execute("SELECT * FROM v_trosak_po_zaposleniku;")
10    trosak_po_zaposleniku = cur.fetchall()
11
12    cur.execute("SELECT * FROM v_trosak_rada_po_projektu;")
13    trosak_po_projektu = cur.fetchall()
14
15    cur.close()
16    conn.close()
17
18    return render_template(
19        "izvjestaji.html",
20        financijski_pregled=financijski_pregled,
21        trosak_po_zaposleniku=trosak_po_zaposleniku,
22        trosak_po_projektu=trosak_po_projektu
23    )

```

Isječak koda 9: Ruta za generiranje izvještaja

Izvještaji korisniku pružaju cijelovit uvid u financijsko stanje projekata, uključujući ukupne troškove rada, dodatne troškove i preostali budžet.

Financijski pregled projekata					
Projekt	Budžet buffer	Planirani trošak	Dodatni troškovi	Stvarni trošak	Preostali budžet
Interni HR sustav	800.00	2780.00	50.00	1886.00	1694.00
Izrada aplikacije za blagajne	2500.00	12232.00	50.00	50.00	14682.00
Migracija baze podataka	500.00	4077.33	20.00	3980.00	597.33
Web aplikacija za evidenciju rada	1500.00	10286.00	0.00	10512.00	1274.00

Slika 6: Prikaz izvještaja o troškovima projekata

Jedan od generiranih izvještaja prikazuje ukupno evidentirano vrijeme rada i pripadajući trošak po zaposleniku. Izvještaj omogućuje analizu angažmana zaposlenika na svim projektima te uvid u ukupne troškove rada po pojedinom zaposleniku.

Trošak po zaposleniku		
Zaposlenik	Ukupno sati	Ukupni trošak
Ivan Ivč	288.00	2880.00
Marko Marić	288.00	3456.00
Ana Kovač	320.00	4872.00
Ivo Ivč	0	0
Pero Petrović	296.00	2812.00
Petra Horvat	176.00	2288.00

Slika 7: Izvještaj o trošku rada po zaposleniku

Osim pregleda po zaposlenicima, aplikacija omogućuje i analizu troškova rada po projektima. Ovaj izvještaj, iako malo redundantan zbog 6, prikazuje ukupni trošak rada akumuliran kroz sve radne sesije povezane s pojedinim projektom. Ovaj izvještaj postoji kako bi se smanjio broj podataka koji se prikazuju i da se korisniku prikaže samo projekt i njegov trenutni trošak.

Trošak rada po projektu		
Projekt		Trošak rada
Interni HR sustav		1836.00
Migracija baze podataka		3960.00
Web aplikacija za evidenciju rada		10512.00

Slika 8: Izvještaj o trošku rada po projektu

5.3. Sažetak aplikacije

Razvijena aplikacija predstavlja jednostavan, ali funkcionalan korisnički sloj nad aktivnom i temporalnom bazom podataka. Jasno razdvajanje aplikacijskog sloja i baze podataka omogućilo je centraliziranu provedbu poslovnih pravila, automatizirani obračun rada i pouzdano generiranje finansijskih izvještaja. Sustav demonstrira praktičnu bazu podataka u realnom poslovnom kontekstu kako bi se olakšalo praćenje troškova rada vezano uz radnike i planiranje budžeta projekata.

6. Zaključak

U ovom projektu razvijen je informacijski sustav za temporalno mjerjenje rada i obračun troškova rada zaposlenika na projektima, s naglaskom na primjenu aktivnih i temporalnih baza podataka. Cilj rada bio je demonstrirati kako se napredne mogućnosti relacijskih baza podataka mogu učinkovito koristiti za provedbu poslovnih pravila, automatizaciju izračuna i osiguravanje konzistentnosti podataka u realnom poslovnom kontekstu.

Razvijeni sustav omogućuje evidenciju radnih sesija različitih vrsta rada, automatski izračun trajanja rada i pripadajućih troškova te generiranje finansijskih izvještaja za projekte i zaposlenike. Temporalni aspekti baze podataka omogućuju precizno praćenje rada kroz vrijeme, dok aktivni mehanizmi, implementirani pomoću okidača i pohranjenih funkcija, osiguravaju da se poslovna pravila dosljedno provode prilikom svakog unosa ili izmjene podataka.

Posebna vrijednost projekta ogleda se u centralizaciji poslovne logike na razini baze podataka. Izračun planiranih i stvarnih troškova, kontrola vremenskih ograničenja te zabrana nedopuštenih unosa provode se neovisno o aplikacijskom sloju, čime se smanjuje mogućnost pogreške i povećava pouzdanost sustava. Takav pristup jasno demonstrira prednosti aktivnih baza podataka u odnosu na tradicionalne sustave u kojima se većina logike nalazi u aplikaciji.

Korisničko sučelje razvijeno korištenjem programskog okvira Flask pruža jednostavan i pregledan pristup sustavu te omogućuje unos i pregled podataka bez potrebe za poznavanjem strukture baze podataka. Aplikacija služi kao tanak sloj iznad baze podataka, dok se ključni izračuni i pravila izvršavaju izravno u bazi.

Rezultati projekta pokazuju da je moguće izgraditi funkcionalan i proširiv sustav za evidenciju rada i praćenje troškova korištenjem standardnih tehnologija relacijskih baza podataka uz primjenu aktivnih i temporalnih koncepata. Sustav omogućuje jasno razdvajanje planiranih i stvarnih troškova, što predstavlja važnu osnovu za upravljanje projektima i donošenje finansijskih odluka.

Mogući smjerovi daljnog razvoja sustava uključuju proširenje sustava korisničkim ulogama i ovlastima, detaljniju evidenciju dodatnih troškova, podršku za izvještavanje po vremenskim razdobljima te integraciju s vanjskim sustavima za upravljanje projektima ili ljudskim resursima. Također, sustav se može proširiti naprednjim temporalnim modelima koji bi omogućili analizu povijesnih promjena podataka kroz dulja vremenska razdoblja.

Popis literature

- [1] C. S. Jensen i R. T. Snodgrass, *Temporal Databases*. San Francisco, USA: Morgan Kaufmann, 1999., ISBN: 978-1558604369.
- [2] N. W. Paton, *Active Database Systems*. Berlin, Germany: Springer, 1999., ISBN: 978-0387984246.
- [3] H. Kerzner, *Project Management: A Systems Approach*, 12. izdanje. New York, USA: Wiley, 2017., ISBN: 978-1119165354.

Popis slika

1.	ERA model	8
2.	Korisničko sučelje za pregled i upravljanje projektima	17
3.	Obrazci za dodavanje novog projekta i dodatnih troškova	18
4.	Sučelje za unos radnih sesija	20
5.	Pregled postojećih sesija	20
6.	Prikaz izvještaja o troškovima projekata	21
7.	Izvještaj o trošku rada po zaposleniku	22
8.	Izvještaj o trošku rada po projektu	22

Popis isječaka koda

1.	Funkcija za automatski izračun trajanja i troška radne sesije	11
2.	Funkcija za zabranu unosa rada na završenom projektu	12
3.	Funkcija za izračun planiranog troška projekta	13
4.	Pogled <i>v_financijski_pregled_projekta</i>	14
5.	Pogled za finansijski pregled projekta	15
6.	Funkcija za uspostavu veze s bazom podataka	16
7.	Ruta za prikaz popisa projekata	17
8.	Ruta za unos nove radne sesije	19
9.	Ruta za generiranje izvještaja	21