

**PANEVROPSKI UNIVERZITET APEIRON
FAKULTET INFORMACIONIH TEHNOLOGIJA**

**Redovne studije
Smjer „Programiranje i softversko inženjerstvo”**

**„Projektovanje i izrada aplikacije za evidenciju radnih naloga i
servisnih listova”
(diplomski rad)**

**Mentor
Doc. dr Dražen Marinković**

**Student
Saša Pađen**
Index br. 9-23-FITIIT1-pro-240-s

Banja Luka, novembar 2024.godine

PANEVROPSKI UNIVERZITET**APEIRON****АПЕИРОН**

za multidisciplinarnu i virtualnu studiju
Pan-European University for Multidiscipline & Virtual Studies
Banja Luka

SASA PADEN

(Ime i prezime studenta)

IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Izjavljujem i svojim potpisom dajem saglasnost Panevropskom univerzitetu „APEIRON“, kao nosiocu prava iskorištavanja, da moj diplomski/specijalistički rad pod nazivom

PROJEKTOVANJE I IZRADA APLIKACIJE ZAEVIDENCIJU RADNIH NALOGA I SERVISNIH LISTOVA

(naslov rada)

koristi na način da ga, u svrhu stavljanja na raspolaganje javnosti, kao cjeloviti tekst ili u skraćenom obliku trajno objavi u javno dostupni repozitorijum Panevropskog univerziteta „APEIRON“, a sve u skladu sa Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom. Korištenje diplomskog/specijalističkog rada na navedeni način ustupam bez naknade.

Student

SASA PADEN

(potpis)

U B.Luci, 20.11.2024

SAŠA PADEN

(Ime i prezime studenta)

IZJAVA O AKADEMSKOJ ČESTITOSTI

Izjavljujem i svojim potpisom potvrđujem da je diplomski rad

PROJEKTOVANJE I IZRADA APLIKACIJE ZA

EVIDENCIJU RADNIH NALOGA I SERVISNIH LISTOVA

(naslov rada)

isključivo rezultat mog vlastitog rada uz preporuke i konsultacije sa mentorom. Rad se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju korišćene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da nije prepisan iz necitiranog rada, te da nijedan dio rada ne ugrožava bilo čija autorska prava. U izradi rada pridržavao(la) sam se Pravilnika o diplomskom/specijalističkom radu.

Student

U B.Luci, 20.11.2024

SAŠA PADEN

(potpis)

Sadržaj:

1. UVOD	3
1.1. Pozadina i kontekst projekta	3
1.2. Svrha i ciljevi rada	3
1.3. Metodologija rada	4
1.4. Struktura rada.....	5
2. TEORIJSKA OSNOVA.....	7
2.1. Tehnologije korišćene u projektu	7
2.2. Definicija,svrha i prednosti aplikacije	9
2.3. Savremeni trendovi u razvoju softvera.....	9
2.4. Osnove baza podataka	11
2.5. Uloga API-ja (Application Programming Interface).....	13
3. ANALIZA PROBLEMA	15
3.1. Identifikacija problema u trenutnim procesima	15
3.2. Razlozi za razvoj nove aplikacije.....	16
3.3. Ciljni korisnici i njihovi zahtjevi.....	17
3.4. Uticaj tehnologije na evidenciju radnih naloga.....	18
3.5. Analiza korisničkih potreba	19
4. PLANIRANJE I DIZAJN APLIKACIJE.....	20
4.1. Analiza zahtjeva korisnika.....	20
4.2. Vizuelni dizajn korisničkog interfejsa.....	20
4.3. Priprema za testiranje i prikupljanje povratnih informacija.....	21
4.4. Prednosti i izazovi korisničkog iskustva.....	21
4.5. Planiranje budućeg razvoja aplikacije	22
4.6. Razvoj i unapređenje korisničkog iskustva.....	23
4.7. Samostalno testiranje aplikacije i njegov značaj.....	23
5. IMPLEMENTACIJA APLIKACIJE	24
5.1. Struktura baze podataka.....	24
5.2. Razvoj backenda u PHP-u	28
5.3. Razvoj frontend-a (HTML, CSS).....	29
6. ZAKLJUČAK.....	40
7. LITERATURA	41

1. UVOD

1.1. Pozadina i kontekst projekta

Ako pogledamo današnje poslovno okruženje, u vođenju poslovnih procesa igraju ključnu ulogu zapravo efikasnost i tačnost. Većina kompanija, uglavnom one koje pružaju servisne usluge, imaju razne izazove u praćenju i evidencijama radnih naloga i servisnim aktivnostima. Recimo tradicionalni manuelni načini vođenja evidencija, poput bilježenja podataka na papiru ili u jednostavnijim tabelama, uglavnom dovodi do greške, gubitaka informacija i kašnjenja u izvršavanju radova. Ovi pristupi mogu uzrokovati nepreciznost u radu, što još više utiče na produktivnost zaposlenih, ali i na zadovoljstva klijenata.

Upravo su zato programeri razvili aplikacije za evidenciju radnih naloga i servisnih listova, koje omogućavaju digitalizaciju procesa i automatizaciju unosa podataka. Ove aplikacije pružaju tačne informacije, omogućavaju zaposlenicima jednostavnije preglede i ažuriranje podataka u stvarnom vremenu, čime se smanjuju mogućnosti ljudskih grešaka.

Ako uzmemo u obzir moderne trendove u razvoju softvera, poput korišćenja cloud tehnologija i mobilnih aplikacija, dodatno doprinose unapređenju poslovnih procesa. Mogućnosti pristupanja informacijama bez obzira na lokaciju omogućuje zaposlenicima veće fleksibilnosti i efikasnosti u bilo kojem trenutku. To je jako važno jer se u industriji zahtjevaju brza donošenja odluka i brzi odgovori na potrebe klijenata.

Projektovanje i izrada aplikacije za evidenciju radnih naloga i servisnih listova je korak naprijed u optimizaciji poslovanja, pogotovo u kompaniji koja se bavi servisnim uslugama. Ova aplikacija je razvijena sa ciljem da unaprijedi načine vođenja radnih naloga, omogući brže obrade podataka. Korišćene su tehnologije kao što su PHP, MySQL, HTML i CSS, i aplikacija će da pruži korisnicima intuitivan interfejs, jednostavnu navigaciju i robustan backend koji će osigurati sigurnost i integritet podataka.

1.2. Svrha i ciljevi rada

Svrha ovog rada je razvoj aplikacije za evidenciju radnih naloga i servisnih listova koja će da omogući efikasnija upravljanja administrativnim procesima unutar organizacije. Ako pogledamo današnje poslovno okruženje, možemo češće da vidimo velike količine podataka, važno je da imamo sistem koji neće samo da nam olakša unose informacija, već i da poboljša pristupe i analiziranje tih podataka. Razvijena je aplikacija koja će da omogući korisnicima da na jednostavne i intuitivne načine unose podatke o komitentima, rezervnim djelovima i radnim nalogima. To će da smanji vrijeme koje se provede na administrativnim poslovima i to će da poveća ukupnu produktivnost.

Takođe treba naglasiti još jedan od ključnih ciljeva a to je da se osiguraju pouzdanosti i sigurnosti podataka, što je od ključnog značaja za svaku aplikaciju koja upravlja osjetljivim informacijama. Osim što će se lakše unositi i upravljati podacima, aplikacija će da omogući i analiziranje informacija kroz generisanje izveštaja. Izveštaji će da doprinesu menadžmentu u

donošenjima informisanih odluka, što će definitivno da unaprijedi strategiju poslovanja i operativne procese. Kroz strukturu koda aplikacija će moći da se ubuduće nadograđuje, i tako će da omogućiti dugoročna korišćenja i prilagođavanja potrebama korisnika.

Na kraju, svrhe i ciljevi rada ne mogu da se ograniče samo na tehničke aspekte aplikacije. Cilj kroz izradu projekta je bio da se poveća svijest o značajima digitalizacije i modernizacije poslovnih procesa. Kako je kroz vrijeme implementirano ovo rješenje, rješenje bi bilo da organizacije zaslužuju priliku da unaprijede svoje radne tokove, da se smanje administrativni troškovi i da se stvore vrijednosti za svoje klijente. Svakako će u tom smislu rad poslužiti kao primjer dobre prakse u raznim sektorima i da će da inspiriše ljude da prepoznaju koristi od modernizacije svojih poslovnih procesa.

1.3. Metodologija rada

Metodologija rada je ono što je zapravo ključni dio svih istraživačkih i razvojnih projekata, budući da definiše pristupe koji se koriste za postizanja postavljenih ciljeva. Ako se pogleda ovaj projekat, razvoj aplikacije za evidenciju radnih naloga i servisnih listova, primjenjena je metodologija koja se oslanja na kombinaciju tradicionalnog i agilnog pristupa, a to dakle omogućuje fleksibilnosti, iterativna usavršavanja i prilagođavanja za potrebe korisnika. Budući da su poslužili obrasci od mentora to pripada strukturiranom pristupu što je karakteristično za tradicionalni model, a sa druge strane vršene su izmjene tokom procesa i to je u skladu sa agilnim pristupom.

Prikupljanje zahtjeva je jedan važan korak u razvijanju softvera, jer osnovna važnost da se razjasni potreba i očekivanje korisnika. U ovom projektu, proces prikupljanja zahtjeva olakšan je zahvaljujući angažmanu mentora, koji je jednim djelom pružio gotove obrasce. Ovi obrasci sadrže ključne podatke o funkcionalnostima koje korisnici očekuju od aplikacije, kao i o osnovnim karakteristikama koje treba implementirati. U toku implementacije analizirano je koja bi pitanja korisnici mogli postaviti:

- Ko su korisnici aplikacije? To mogu biti administratori, tehničari ili menadžeri koji će koristiti sistem za evidenciju radnih naloga i servisnih listova.
- Koje su osnovne funkcije koje korisnici očekuju od aplikacije? Na primjer, dodavanje novih komitenata, unos rezervnih djelova ili generisanje izveštaja.

Specifični zahtjevi odnose se na detaljne funkcionalnosti i aplikacija ih mora imati kako bi zadovoljila potrebe korisnika. Možemo ih podjeliti u nekoliko kategorija:

- Korisnici treba da mogu da dodaju nove komitente u sistem.
- Aplikacija treba da omogućiti pretragu rezervnih delova prema različitim kriterijumima (npr. naziv, cijena).
- Korisnici treba da mogu generisati izvještaje o radnim nalogima.

Ne-funkcionalni zahtjevi se odnose na performanse i kvalitet aplikacije, kao što su:

- Brzina učitavanja stranica.
- Odnos između korisnika i sistema (npr. maksimalni broj korisnika koji mogu istovremeno koristiti aplikaciju bez usporavanja).

- Bezbjednost podataka, kao što su šifrovanje korisničkih informacija i kontrola pristupa.

Planiranje i dizajn mogu se smatrati vitalnim značajem da bi uspio bilo koji softverski projekat. U ovoj fazi rada, fokus mora da bude na definisanju opsega projekta, postavljanju ciljeva i određivanju resursa potrebnih za realizaciju. Ključnim aspektima se može smatrati:

- Definisanje opsega: odrediti koje funkcionalnosti aplikacija mora da ima;
- Ciljevi i rokovi: mora se postaviti realan cilj i rok za svaki fazu razvoja. To će da pomogne pri organizaciji i planiranju resursa.
- Identifikacija resursa: Mora se razmotriti koje alate koristiti. U ovom slučaju, korišćene su tehnologije PHP, MySQL, HTML i CSS.

Da bi se omogućilo postizanje funkcionalnosti mora se uključiti stvaranje arhitekture i strukture a to zapravo uključuje dizajn aplikacije. Budući da nisu pravljenе skice, uz pomoć smjernica određen je aspekt dizajna. Ključni dio dizajna su strukture baze podataka, definisanje tabela i odnosi među njima:

- Korisnički interfejs: smjernice su pomogle da se definiše kako će izgledati forme za unos podataka.
- Funkcionalnosti: načini na koji će se vršiti komunikacija, šta treba da se unese, na taj način će se poboljšati korisnički interfejs.

Razvoj i implementacija: može se reći da ovaj pristup omogućuje bržu reakciju na promjenu zahtjeva i lakše dodavanje novih funkcionalnosti. U ovom koraku, korišćeni su programi kao što su PHP za backend razvoj, HTML i CSS za frontend dizajn, kao i MySQL za upravljanje podacima. Svaki dio aplikacije razvijen je s obzirom na ranije definisane zahtjeve, i redovno je vršeno testiranje kako bi se vidjelo da sve ispravno radi.

Testiranje: kada je razvijena aplikacija, ona je prošla kroz proces testiranja. Cilj je zapravo bio da se identifikuje da li ima potencijalnih grešaka i nedostataka i da svakako se provjeri da li aplikacija ispunjava sve zahtjeve koji su definisani.

Održavanje i unapređenje: važno je da se obezbjede kontinuirana održavanja i unapređenja aplikacija nakon što se puste u rad. Ta održavanja podrazumjevaju pravovremena rješavanja problema, takođe i da se nadogradnja implementira da se poboljšaju funkcionalnosti i korisnička iskustva. Za ovu metodologiju rada može se reći da obezbjeđuje sveobuhvatan pristup razvijanju aplikacija, budući da garantuje da su aspekti isplanirani i realizovani.

1.4. Struktura rada

Ovaj rad je podjeljen u nekoliko poglavlja koja prate tok razvoja aplikacije za evidenciju radnih naloga i servisnih listova. Osmišljeno je tako da se logički vodi kroz svaki dio, od pozadinskih istraživanja, preko analize problema, do implementacije aplikacije.

Uvod (Poglavlje 1): dakle ovdje je rečeno o osnovnim detaljima o projektu, o pozadini, svrhi, ciljevima i metodologiji rada. Cilj je da se jasno predstavi kontekst, i da se omogući uvid u ključnu stvar na kojima se zasniva aplikacija.

Teorijska osnova (Poglavlje 2) biće objašnjeno koje su korišćene tehnologije. Svakako da će ovo poglavlje dati uvid u osnovne koncepte aplikacije za evidenciju radnih naloga i servisnih listova.

Analiza problema (Poglavlje 3) biće objašnjeno koji su to problemi sa kojima se suočava korisnik, takođe će da se obrazloži razlog za razvoj nove aplikacije.

Planiranje i dizajn aplikacije (Poglavlje 4) biće prikazani funkcionalni i nefunkcionalni zahtjevi aplikacije, kao i vizuelni dizajn korisničkog interfejsa.

Implementacija aplikacije (Poglavlje 5) opisuje tehničku realizaciju projekta, uključujući strukturu baze podataka, razvoj backend-a u PHP-u i frontend-a uz HTML i CSS. Takođe, objašnjava se integracija sa MySQL bazom i testiranje funkcionalnosti.

Zaključak (Poglavlje 6) sumiraće rezultate rada, biće predložene potencijalne nadogradnje aplikacije i pružiće osvrt na stečena znanja tokom procesa izrade ovog projekta.

Literatura se nalazi na kraju rada.

2. TEORIJSKA OSNOVA

2.1. Tehnologije korišćene u projektu

PHP (Hypertext Preprocessor) je serverski skriptni jezik koji se koristi u razvoju web aplikacija. Čine ga idealnim za realizaciju web aplikacija upravo zbog njegove fleksibilnosti, integracije sa bazama, i jednostavan je u pisanju dinamičkih kodova. Jedna od ključnih prednosti PHP-a je ta sposobnost za lako povezivanje sa različitim bazama podataka, uključujući MySQL, koji je korišćen u ovom projektu. Kroz jednostavnije SQL upite, PHP omogućuje manipulaciju podacima. PHP takođe nudi različite funkcionalnosti koje olakšavaju rad sa korisničkim unosima, kao što su validacija podataka, obrada formi, i prikaz povratnih informacija korisnicima. Kada se sve to implementira to dovodi do smanjenja grešaka prilikom unošenja podataka. U projektu je korišćen ovaj jezik da bih obradio podatke.

Na primjer korisnik želi da popuni formu za unos novog radnog naloga, PHP se koristi za validaciju podataka (provjera da li su sva polja ispravno popunjena) i unos tih podataka u MySQL bazu. Jednostavan PHP kod koji obavlja ovu funkciju:

```
if(isset($_POST['submit'])){\n\n    $naziv = $_POST['naziv'];\n\n    $opis = $_POST['opis'];\n\n\n    // Validacija unosa\n\n    if(!empty($naziv) && !empty($opis)){\n\n        $sql = "INSERT INTO radni_nalog (naziv, opis) VALUES ('$naziv', '$opis')";\n\n        mysqli_query($conn, $sql);\n\n        echo "Podaci su uspešno uneti.";\n\n    } else {\n\n        echo "Molimo popunite sva polja.";\n\n    }\n\n}
```

Slika 1

MySQL: Za ovaj projekat, izabran je MySQL, ona je jedna od najrasprostranjenijih relacionih baza podataka. MySQL se inače koristi u velikim aplikacijama i poznat je po svojoj stabilnosti i mogućnosti upravljanja velikim količinama podataka. MySQL je odlučan zato što je jednostavan u korišćenju, fleksibilan je, i lagan je za povezivanje sa PHP-om. MySQL je baza podataka koja koristi SQL (Structured Query Language) za manipulaciju podacima. SQL je

standardizovan jezik za rad sa bazama podataka, što ga čini veoma korisnim u slučajevima kada je potrebna precizna manipulacija podacima, poput kreiranja, čitanja, ažuriranja i brisanja podataka (poznato kao CRUD operacije). Jedan od glavnih razloga za izbor MySQL-a jeste i njegova podrška za relacioni model. Relacione baze podataka omogućuju čuvanja podataka u tabelama koje mogu biti međusobno povezane kroz odnose, što omogućava efikasnu organizaciju i pretraživanje podataka.

Na primjer, jednostavan SQL upit za prikaz svih radnih naloga može izgledati ovako:

```
SELECT * FROM radni_nalog;
```

Ovaj upit vraća sve redove iz tabele radni_nalog, omogućava korisniku da pregleda podatke. Takođe, MySQL podržava kompleksne upite, uključujući filtriranje podataka prema specifičnim kriterijumima, agregiranje podataka i spajanje više tabela.

HTML (Hypertext Markup Language) i CSS (Cascading Style Sheets) predstavljaju osnovne tehnologije za razvoj korisničkog interfejsa. HTML se koristi za strukturu sadržaja, dok CSS omogućava stilizaciju elemenata i prilagođavanje vizualnog izgleda stranice. U projektu za evidenciju radnih naloga i servisnih listova, HTML je korišćen za kreiranje formi, tabela i drugih elemenata koji omogućavaju unos i prikaz podataka.

Recimo forma za unos novog radnog naloga napravljena je pomoću HTML-a:

```
<form action="submit_radni_nalog.php" method="post">  
  <label for="naziv">Naziv naloga:</label>  
  <input type="text" id="naziv" name="naziv" required>  
  
  <label for="opis">Opis:</label>  
  <textarea id="opis" name="opis" required></textarea>  
  
  <input type="submit" name="submit" value="Pošalji">  
</form>
```

Slika 2

Cascading Style Sheets (CSS) je korišćen u ovom projektu kako bi se definisao vizuelni izgled korisničkog interfejsa. CSS omogućava jasno razdvajanje strukture (HTML) i stila, čime se doprinosi boljoj organizaciji koda i jednostavnijem održavanju projekta. CSS je implementiran za oblikovanja različitih elemenata, poput formi, tabela i navigacionih elemenata, kako bi aplikacija izgledala moderno, profesionalno i intuitivno za krajnje korisnike. Korišćenje CSS-a omogućilo je kontrolu nad bojama, marginama, fontovima, kao i rasporedu elemenata na stranici. Jedna od ključnih oblasti primjene CSS-a u projektu jeste stilizacija formi za unos

podataka. Da bi se poboljšalo korisničko iskustvo, formulari su oblikovani tako da budu jednostavni za korišćenje, a istovremeno i estetski prijatni. Na primjer, kroz CSS je definisan raspored elemenata, boje pozadine, dimenzije polja za unos, kao i stil dugmadi za slanje podataka.

2.2. Definicija, svrha i prednosti aplikacije

U savremenim poslovanjima, efikasna upravljanja radnim nalogima i servisnim listovima su od izuzetnog značaja za održavanja produktivnosti i organizacija. Aplikacija za evidenciju radnih naloga i servisnih listova osmišljena je sa ciljem da unaprijedi i pojednostavi procese unosa, praćenje i analizu radnih zadataka. Ova aplikacija ne može se smatrati samo alatom, već i sveobuhvatnim sistemom koji pomaže u optimizaciji radnih procesa, i tako se stvaraju osnove za poboljšavanje opštih efikasnosti.

Ova aplikacija znatno štedi vrijeme i poboljšava organizaciju, što je ključno u radnim okruženjima gde se donose brze odluke. Takođe, ova aplikacija omogućuje generisanje izveštaja koji sadrže informacije o pojedinim stavkama iz radnih naloga. Ovaj izveštaj može uključivati razne stavke recimo opis, datum, količina i ti podaci su od ključne važnosti za recimo menadžment, koji može na osnovu toga da donese važne odluke u unapređenju radnog procesa. Recimo, aplikacija takođe može da uključi i mehanizam povratnih informacija, koji će da omogući korisniku da izrazi svoje mišljenje i sugestiju u vezi sa radnim nalogima. U današnjim vremenima, kada se zahtjevi na tržištu brzo mijenjaju, od suštinskog je značaja da se aplikacija prilagodi potrebama korisnika. Takođe jedna od važnih elemenata aplikacija su i sigurnosti podataka. Pravilna upravljanja i zaštita podataka o radnim nalogima i korisnicima je od izuzetnog značaja, budući da je potrebno da se implementiraju mehanizmi zaštite poput autentifikacije korisnika i enkripcije podataka. Tako će da se osiguramo poverljivosti informacija i da se zaštitimo od neovlašćenih pristupa. U praksi, aplikacija za evidenciju radnih naloga i servisnih listova mogu da budu korišćene u raznim industrijama, od građevinskih do IT sektora. Recimo, u građevinskoj industriji, radni nalozi se često koriste za upravljanje projektima, i tu se prati napredak radova, troškovi i resursi. Implementacija aplikacije može da olakša praćenje statusa projekata i da pomogne menadžmentu da brže identifikuje i riješi eventualan problem. Aplikacija za evidenciju radnih naloga i servisnih listova predstavlja neophodan alat za savremena preduzeća koja žele da unaprijede svoje procese upravljanja radnim zadacima. Njihove funkcionalnosti, od jednostavnih unosa podataka do analiza i izvještavanja, pomažu korisnicima da optimizuju svoje poslovanje, poboljšaju efikasnost i donose bolje odluke na osnovu dostupnih informacija. Kroz sveobuhvatan pristup, ova aplikacija stvara temelj za uspješno upravljanje radnim procesima i resursima, što na kraju doprinosi postizanju poslovnih ciljeva.

2.3. Savremeni trendovi u razvoju softvera

U posljednjim godinama, razvoj softvera se znatno promjenio, prilagođavajući se brzim promjenama u tehnologiji i potrebama tržišta. Ovi trendovi ne samo da utiču na način na koji se softver razvija, već i na kvalitet i efikasnost konačnog proizvoda.

Dva ključna trenda su:

1. Agile metodologija ;
2. Iterativni razvoj.

Agile metodologija je postala prisutna u razvoju softvera zbog svoje sposobnosti da se prilagodi promjenljivim zahtjevima korisnika i tržišta. Za ovu metodologiju možemo reći da se zasniva na fleksibilnostima, saradnjama i iterativnim pristupima razvoja. Ključni principi Agile metodologije uključuju:

- Interakcija i saradnja: Umjesto da se oslanja na rigorozne planove i dokumentaciju, Agile naglašava timsku saradnju i direktnu komunikaciju među članovima tima. Ovo omogućava brže rješavanje problema i prilagođavanje promjenama.
- Iterativni i inkrementalni razvoj: Projekti se razvijaju u kratkim ciklusima, poznatim kao sprintovi. Svaki sprint donosi određeni dio funkcionalnosti, što omogućava korisnicima da dobiju djelove proizvoda u ranoj fazi razvoja i pruže povratne informacije.
- Fokus na korisničke potrebe: Agile metodologija stavlja korisnika u centar procesa razvoja. Timovi redovno prikupljaju povratne informacije od korisnika i prilagođavaju funkcionalnosti prema njihovim potrebama.

U okviru Agile metodologije, često se koristi nekoliko specifičnih okvira, kao što su Scrum i Kanban. Scrum se fokusira na rad u timovima i definisanje uloga, dok Kanban naglašava vizualizaciju radnog toka i optimizaciju procesa.

Iterativni razvoj je pristup koji omogućuje timovima da kontinuirano poboljšavaju softver kroz seriju malih i kontrolisanih promjena. Ovaj pristup vrši čestu kombinaciju sa Agile metodologijom, mada može biti primjenjen i van Agile konteksta.

Osnovne karakteristike iterativnog razvoja uključuju:

- Postepeno unapređenje: Softver se razvija kroz niz iteracija, gdje se svaka iteracija fokusira na dodavanja ili poboljšanja određenih funkcionalnosti. Ovo omogućuje timovima da se brže prilagode promjeni i povratnoj informaciji korisnika.
- Testiranje i povratne informacije: U svim fazama razvoja, vrši se testiranje softvera i prikupljaju se povratne informacije. Te informacije su ključne da bi se donjela odluka o sljedećem koraku u razvoju.
- Smanjenje rizika: Budući da se razvoj odvija u malim koracima, rizici se smanjuju. Problemi se mogu identifikovati i rješavati brzo, prije nego što postanu veći problemi.

U okviru ovog projekta za evidenciju radnih naloga i servisnih listova, primjenjeni su principi Agile metodologije i iterativnog razvoja. Tokom faze razvoja, implementirane su male iteracije, gde su se nove funkcionalnosti, poput generisanja izveštaja o radnim nalogima, dodavale u sistem postepeno. Ovaj pristup je omogućio da se osigura visoka kvaliteta proizvoda, dok se istovremeno fokusira na potrebe korisnika.

Savremeni trendovi u razvoju softvera, kao što su Agile metodologija i iterativni razvoj, svakako su promijenili način na koji se softver razvija. Ova dva pristupa su omogućila timovima da budu fleksibilniji, efikasniji i orijentisani prema korisnicima, što će da rezultira kvalitetnijim proizvodima. Kao rezultat svega toga može se primjetiti da aplikacije sve više postaju prilagođene potrebi korisnika, a procesi razvoja su dosta transparentniji i saradnički orijentisani. Kroz primjenu ovih principa, ovaj projekat za evidenciju radnih naloga i servisnih listova osigurao je da uvijek uzimamo u obzir korisničke potrebe, i tako ćemo da povećamo vrijednost konačnog rješenja.

2.4. Osnove baza podataka

Tipovi baza podataka:

Relacione baze podataka (RDBMS) su najčešći tip baza podataka, one se zasnivaju na relacijskom modelu. U ovom modelu, podaci se organizuju u tabele koje su povezane međusobno putem zajedničkih atributa (ključeva). Ključne karakteristike RDBMS-a uključuju:

- **Struktura podataka:** Podaci se skladište u tabelama, svaka sa redovima (zapisi) i kolonama (atributi). Na primjer, u aplikaciji za evidenciju radnih naloga, možemo imati tabelu "radni_nalog" sa kolonama kao što su "id", "datum", "opis", "status" i "prioritet".
- **SQL (Structured Query Language):** RDBMS koristi SQL kao standardni jezik za upite i manipulaciju podacima. SQL omogućava korisnicima da izvrše razne operacije kao što su selektovanja, umetanja, ažuriranja i brisanja podataka.
- **Transakcije:** Relacione baze podataka podržavaju transakcije, što omogućuje izvršavanje niza operacija kao jedinstvenog procesa. Ovo osigurava integritet podataka, čak i u slučaju grešaka ili problema tokom izvršenja.

U ovoj aplikaciji za evidenciju radnih naloga i servisnih listova, korišćene su relacione baze podataka kako bi kreirali tabele kao što su "komitenti", "rezervni_dijelovi", "ček_lista" i "radni_nalog".

NoSQL baze podataka su alternativa relacijskim bazama, one su dizajnirane za skladištenja i upravljanja velikim količinama podataka, i često su neorganizovane ili poluorganizovane. Glavne karakteristike NoSQL baza uključuju:

- **Fleksibilna struktura podataka:** Podaci se mogu skladištiti u različitim formatima, uključujući dokumente, ključ-vrijednosti, grafovi ili kolone. Ovo omogućava brža prilagođavanje promjenama u podacima.
- **Horizontalna skalabilnost:** NoSQL baze su dizajnirane da se lako skaliraju dodavanjem više servera, što ih čini pogodnim za rad s velikim podacima.

Budući da ova aplikacija za evidenciju radnih naloga i servisnih listova koristi relacione baze podataka, možda bi se u nekoj situaciji moglo da razmotri korišćenje NoSQL baze podataka, poput MongoDB, za skladištenje nestrukturiranih podataka, kao što su korisnički komentari ili povratne informacije.

Za baze podataka možemo reći da su fundamentalni dio modernih aplikacija, jer omogućavaju skladištenja, upravljanja i pristupe podacima. U ovom djelu će se razmotriti osnovni tipovi baza

podataka, kako se oni koriste u aplikacijama, kao i značaj SQL jezika u interakciji s tim bazama. SQL (Structured Query Language) je standardni jezik za upravljanje relacijskim bazama podataka. Omogućava korisnicima da kreiraju, menjaju i upitavaju podatke u bazama. SQL se koristi u različitim aspektima aplikacija, uključujući:

- Definisanje strukture baze podataka: Kroz DDL (Data Definition Language) komande, kao što su CREATE, ALTER i DROP, korisnici mogu da definišu strukturu tabela i relacija.

Recimo u aplikaciji za evidenciju radnih naloga i servisnih listova, koristili bi SQL komandu za kreiranje tabele radnih naloga:

```
CREATE TABLE radni_nalog (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    datum DATE NOT NULL,  
    opis TEXT NOT NULL,  
    status ENUM('otvoren', 'u toku', 'zatvoren') NOT NULL,  
    prioritet ENUM('nizak', 'srednji', 'visok') NOT NULL  
);
```

Slika 3

- Manipulacija podacima: DML (Data Manipulation Language) komande, kao što su INSERT, UPDATE, DELETE i SELECT, koriste se za upravljanje podacima unutar tabela.

Recimo ako želimo da dodamo novi radni nalog, koristili bi SQL komandu:

```
INSERT INTO radni_nalog (datum, opis, status, prioritet)  
VALUES ('2024-10-01', 'Popravka klima uređaja', 'otvoren', 'srednji');
```

Slika 4

- Upiti za pretragu podataka: Kroz SQL upite, korisnici mogu da pretražuju i filtriraju podatke prema različitim kriterijumima.

Recimo da bi prikazali sve otvorene radne naloge, koristili bi SQL komandu:

SELECT * FROM radni_nalog WHERE status = 'otvoren';

Slika 5

U modernom razvoju softvera ključni aspekt zapravo predstavljaju osnove baza podataka. Razumjevanje tipova baza podataka, poput relacionih i NoSQL, omogućuje programerima da izaberu najprikladnije pristupe za skladištenja i upravljanja podacima. U interakciji sa bazama podataka SQL jezik igra ključnu ulogu, jer omogućava efikasna upravljanja podacima i kreiranje složenih upita. U okviru aplikacije za evidenciju radnih naloga i servisnih listova, upotreba relacione baze podataka i SQL jezika omogućava nam da precizno pratimo sve relevantne informacije o radnim nalogima, što doprinosi efikasnosti i produktivnosti poslovanja.

2.5. Uloga API-ja (Application Programming Interface)

API (Application Programming Interface) predstavlja skup pravila i protokola budući da omogućuje komunikacije među različitim softverskim aplikacijama ili komponentama. API-ji definišu kako različiti dijelovi softverskih sistema ili različitih aplikacija mogu međusobno da vrše komunikaciju, da razmjenjuju podatke i izvršavaju funkcionalnosti. U modernom razvoju softvera njihova uloga je od suštinskog značaja, jer omogućava izgradnju kompleksnih sistema na modularane i efikasne načine. Za API možemo reći da funkcioniše kao most među raznim sistemima, jer omogućava da razumiju jedni druge i vrše razmjenu informacija. Kroz API, aplikacija može da komunicira sa funkcijama ili uslugama drugih sistema, bilo da su one interne ili eksterne. API-ji mogu biti lokalni (unutar iste aplikacije) ili daljinski (komunikacija sa udaljenim uslugama).

Kako API-ji poboljšavaju funkcionalnost?

1. Modularnost i ponovna upotreba:

API-ji omogućuju razvoj modularnih aplikacija, gde se različiti dijelovi sistema mogu razvijati i nadograđivati nezavisno. Ovaj pristup smanjuje iznova pisanje koda, što smanjuje vrijeme razvoja i povećava efikasnost. Na primjer, u aplikaciji za evidenciju radnih naloga i servisnih listova, API može da se koristi za integraciju modula za upravljanje korisnicima, generisanje izveštaja, i plaćanje unutar aplikacije za radne naloge.

2. Integracija sa spoljnim uslugama:

Pomoću API-ja, aplikacije mogu komunicirati sa eksternim resursima i uslugama, čime se njihova funkcionalnost značajno proširuje. Na primjer, aplikacija za radne naloge može koristiti API za slanje obavještenja putem emaila ili SMS-a, automatsko preuzimanje informacija ili integraciju sa softverom poput Trello ili Asana. Kada se status naloga promjeni, API može poslati automatsku notifikaciju korisniku.

3. Fleksibilnost i prilagodljivost:

API-ji pružaju mogućnost jednostavne prilagodbe aplikacija novim poslovnim zahtevima ili tehnološkim promjenama. Na ovaj način, programeri mogu dodavati nove funkcionalnosti ili mijenjati postojeće bez potrebe za velikim promjenama u samoj aplikaciji, što je ključno u okruženjima koja se brzo mijenjaju.

4. Brž i laksa integracija:

API-ji omogućavaju bržu integraciju između različitih softverskih sistema, što olakšava razvoj novih funkcionalnosti. Umjesto kreiranja sopstvenih rješenja, razvojni timovi mogu koristiti već postojeće API-je, što skraćuje vrijeme razvoja, pojednostavljuje testiranje i održavanje sistema.

5. Povezivanje sa mobilnim aplikacijama:

API-ji su od suštinskog značaja za mobilne aplikacije jer omogućavaju povezivanje sa backend servisima. Recimo mobilna verzija aplikacije za radne naloge može koristiti API za pristup podacima o otvorenim nalogima, dodavanje novih i pregled prethodnih. API omogućava sinhronizaciju u realnom vremenu, čime korisnici imaju uvijek ažurirane informacije.

API-ji su neophodni za savremeni razvoj softvera, omogućavajući komunikaciju i integraciju između različitih komponenti i usluga. Korišćenjem API-ja, aplikacije postaju fleksibilnije, funkcionalnije i prilagodljivije potrebama korisnika. U slučaju aplikacije za radne naloge, API može automatizovati procese, poboljšati komunikaciju i značajno unaprediti korisničko iskustvo.

3. ANALIZA PROBLEMA

3.1. Identifikacija problema u trenutnim procesima

Da bi ovu tačku objasnili temeljno, potrebno je analizirati trenutne manuelne procese u sistemu i jasno naglasiti njihove slabosti. Takođe ćemo se osloniti na primjere kako bi analiza bila što transparentnija i zasnovana na činjenicama. U mnogim firmama, vođenje radnih naloga i servisnih listova i dalje se vrši ručno ili uz pomoć jednostavnih digitalnih alata poput Excel tabela ili papirnih dokumenata. Ovakav način rada ima brojne manjkavosti, uzrokujući neefikasnost, greške i poteškoće u analizi podataka. Analizirajući ove probleme, fokusiraćemo se na ključne elemente trenutnog sistema i njihov uticaj na poslovanje.

Ručno vođenje procesa, oslanjajući se na papirnu dokumentaciju, sporo je, sklono greškama i teško za praćenje. U mnogim firmama radni nalozi se upisuju u tabele ili na papir. Na primjer, kada se izdaje nalog za popravku uređaja, tehničar dobija papir sa osnovnim podacima poput datuma, opisa posla i imena klijenta. Međutim, kako posao napreduje, status naloga (npr. "u procesu", "završeno") mora se ručno ažurirati, što često rezultira kašnjenjem u informisanju menadžera o trenutnom stanju. Ovo je posebno izazovno za veće firme sa većim brojem naloga. Recimo u firmi koja dnevno izda desetine naloga, praćenje svih tih naloga može postati kompleksno. Nalozi se često zaborave ili dupliraju, jer nema centralnog sistema za praćenje u realnom vremenu. Tehničari se često oslanjaju na bilješke, što otežava praćenje zadataka i njihov napredak. Takođe, ručni unos podataka povećava mogućnost grešaka. Na primjer, prilikom unošenja radnog naloga, može doći do unosa netačnog datuma, klijenta ili detalja posla, što kasnije dovodi do zbrke i nepotrebnih troškova. U nekim slučajevima, važni podaci se izgube jer se papirni dokumenti lako oštete ili pogube.

Još jedna važna mana ručnih procesa je nedostatak transparentnosti. U manuelnim sistemima često nema jasan pregled svih zadataka, što otežava donošenje odluka menadžmentu. Na primjer, ako menadžer želi pregled trenutnog stanja zadataka, mora ručno pregledati papirne dokumente ili tabele, što zahtjeva vrijeme i napor. Ovo dodatno otežava brzo i efikasno donošenje odluka.

Firmama koje su prešle na digitalne alate, kao što su Excel ili Google tabele, i dalje nedostaje veća efikasnost. Iako oni omogućavaju određeni nivo organizacije, praćenje statusa zadataka, raspodela resursa i generisanje izveštaja nije dovoljno automatizovano. Primjeri problema u ovim rješenjima su sledeći:

- **Praćenje zadataka:** U alatima poput Excel-a korisnici mogu uneti osnovne podatke o radnom nalogu, ali praćenje statusa zadataka ("na čekanju", "u toku", "završeno") zahtjeva ručno ažuriranje. Ovo usporava rad, posebno kada je veliki broj zadataka u toku. Primjer: Operater mora ručno da menja status u Excel tabeli kada tehničar završi posao, što često kasni, pa menadžment nema uvid u tačne podatke.
- **Nedostatak automatizacije:** Postojeći alati ne omogućavaju automatsko slanje obavještenja ili izvještaja, već se sve obavlja ručno. Automatizacija bi ubrzala procese i smanjila greške. Primjer: Menadžer traži izveštaj o završenim zadacima, ali operater mora ručno pregledati sve podatke, što oduzima vrijeme i može rezultirati greškama.

- **Nedostatak integracije sa drugim sistemima:** Ovi alati obično nisu povezani sa drugim poslovnim softverom, kao što su CRM ili ERP, što otežava razmjenu podataka između odjeljenja.
Primjer: Završene informacije iz Excel tabele treba ručno preneti u drugi sistem za fakturisanje, što produžava obradu i uvodi greške.
- **Ograničene mogućnosti analize i izvještavanja:** Excel omogućava analizu podataka, ali zahtjeva složene ručne radnje i specifična znanja korisnika. Firmama su potrebna rješenja koja automatski generišu izveštaje i daju jasan pregled performansi.
Primjer: Menadžer želi da analizira prosječno vrijeme obrade određenih zadataka, ali pravljenje izveštaja zahteva složene formule, što povećava šansu za greške.

3.2. Razlozi za razvoj nove aplikacije

Razvoj nove aplikacije za evidenciju radnih naloga i servisnih listova igra ključnu ulogu u modernizaciji poslovnih procesa, poboljšanju organizacije i povećanju efikasnosti. Postoji niz ključnih razloga zašto je ovakva aplikacija neophodna, od automatizacije ručnih procesa do poboljšanja tačnosti i pouzdanosti podataka. U nastavku su navedeni neki od osnovnih razloga za razvoj ove aplikacije:

Povećanje efikasnosti: Jedan od glavnih ciljeva razvoja nove aplikacije jeste unapređenje efikasnosti pri vođenju radnih naloga i servisnih listova. Tradicionalni načini vođenja evidencije, koji uključuju ručni unos i praćenje naloga, često su spori i skloni greškama. Digitalna aplikacija ubrzava i pojednostavljuje ove procese, čineći ih mnogo efikasnijim. Automatizacija je ključna prednost ovakve aplikacije. Umjesto ručnog popunjavanja papirnih obrazaca ili Excel tabela, korisnici mogu unositi podatke direktno u aplikaciju, koja automatski pohranjuje te podatke u bazu. Time se omogućava trenutni pristup informacijama svim ovlaštenim korisnicima. Aplikacija također može slati automatske obavijesti, smanjujući vrijeme potrebno za informisanje tehničara o novim zadacima.

Primjer unapređenja efikasnosti: U postojećem sistemu, operateri moraju ručno evidentirati svaki radni nalog, unositi ga u Excel ili na papir, te obavještavati tehničare. Ako dođe do promjene statusa naloga, operateri ponovo ručno ažuriraju podatke i obavještavaju tim. Nova aplikacija omogućava digitalno kreiranje radnih naloga, dok se ažuriranja i obavijesti tehničarima odvijaju automatski, u realnom vremenu.

Pristup podacima postaje brži i jednostavniji. Menadžeri i tehničari mogu odmah vidjeti status naloga i sve potrebne informacije bez potrebe za pretragom kroz više dokumenata ili tabela. Bolja organizacija zadataka dodatno doprinosi povećanju produktivnosti. Menadžeri će moći pratiti raspodjelu radnih zadataka, prepoznati uska grla u procesu i, prema potrebi, preusmjeriti zadatke. Ovo omogućava ravnomjernu raspodjelu posla i efikasniju upotrebu resursa.

Primjer: Ako određeni tehničar ima previše naloga na čekanju, menadžer može jednostavno prepoznati problem i dio posla preusmjeriti na tehničara s manje zadataka. Time se smanjuju zastoji u radu, a svi zadaci se obavljaju brže i efikasnije.

Smanjenje grešaka: Ručni unos podataka često dovodi do grešaka koje mogu negativno uticati na cijeli radni proces. Neki od najčešćih problema su netačno uneseni podaci, poput pogrešnih datuma, brojeva naloga, šifri klijenata ili tehničara. Ove greške mogu prouzrokovati kašnjenja, nezadovoljstvo klijenata, dodatne troškove i probleme u praćenju rada.

Poboljšanje komunikacije i transparentnosti: Aplikacija omogućava pregled trenutnog statusa svih radnih naloga u realnom vremenu, osiguravajući da svi uključeni u proces budu informisani o tome koji zadaci su završeni, a koji su u toku. Komunikacija između menadžera, tehničara i ostalih zaposlenika je poboljšana, jer svi imaju uvid u stanje naloga u svakom trenutku.

Primjer: Tehničar može u aplikaciji vidjeti koje zadatke treba izvršiti, dok menadžer može pratiti koji su radni nalozi završeni, a koji su još aktivni. Ovo smanjuje nesporazume i omogućava brži odgovor na moguće probleme.

Jednostavnije praćenje i organizacija podataka: Automatskim unosom podataka u bazu, svi podaci postaju lako dostupni i organizovani. Aplikacija omogućava centralizovano pohranjivanje informacija o radnim naložima, komitentima i rezervnim dijelovima, što omogućava menadžerima i tehničarima da jednostavno pretražuju i pregledavaju naloge.

Primjer: Menadžer može jednostavno otvoriti aplikaciju i pregledati sve aktivne ili završene radne naloge, sortirati ih po prioritetu ili datumu te pratiti status svakog zadatka. Time se izbjegava ručno pretraživanje starih papira ili Excel tabela.

3.3. Ciljni korisnici i njihovi zahtjevi

Razvijanje aplikacije za evidenciju radnih naloga i servisnih listova zahtjeva temeljno razumjevanje potreba različitih korisničkih grupa. Svaka od ovih grupa ima specifične zahtjeve koji treba da budu uzeti u obzir kako bi aplikacija bila što funkcionalnija i efikasnija.

1. Prvu grupu korisnika čine tehničari i serviseri koji su aktivno uključeni u izvršavanje radnih zadataka na terenu. Njihova osnovna potreba je brz i jednostavan unos podataka o obavljenim poslovima, uz mogućnost evidentiranja problema ili zastoja. Zbog prirode njihovog posla, interfejs mora biti intuitivan i efikasan, s fokusom na jednostavnost kako bi mogli obaviti posao pod vremenskim pritiskom.
2. Druga ključna grupa su menadžeri, koji nadziru operacije i raspoređuju resurse. Njihov fokus je na analitici i optimizaciji performansi. Očekuju mogućnost generisanja izvještaja koji će im omogućiti uvid u produktivnost, troškove i trajanje različitih projekata, kako bi doneli informisane odluke i usmjerili timove na prioritete zadatke.
3. Administratori čine treću grupu korisnika. Njihov zadatak je da osiguraju stabilan rad aplikacije, upravljaju korisničkim naložima i privilegijama, i rješavaju tehničke poteškoće. Za njih je ključno da imaju potpuni uvid u rad sistema kako bi mogli brzo reagovati na bilo kakve probleme.

Na kraju, klijenti su korisnici koji iniciraju radne naloge. Njihove potrebe uključuju transparentnost u vezi sa statusom naloga i povratne informacije o izvršenim poslovima. Brza i tačna komunikacija s klijentima, bilo putem e-maila ili SMS-a, ključna je za povećanje zadovoljstva i povjerenja u pružene usluge.

Sve ove grupe imaju specifične zahtjeve koji direktno utiču na dizajn aplikacije, od jednostavnog unosa podataka za tehničare do napredne analitike za menadžere. Razumjevanje i implementacija ovih potreba doprinosi izradi aplikacije koja efikasno podržava svakodnevne aktivnosti svih korisnika.

3.4. Uticaj tehnologije na evidenciju radnih naloga

Savremene tehnologije značajno su promijenile način upravljanja radnim nalogima. Digitalizacija je omogućila prelazak sa tradicionalnog papirnog sistema na elektronske platforme, čime je poboljšana pristup informacijama. Korisnici sada mogu brzo da pretražuju i dijele podatke u realnom vremenu, što vodi do veće efikasnosti. Mobilne aplikacije postale su neophodne za timove na terenu, omogućavajući servisnim radnicima direktno unošenje podataka, praćenje statusa naloga i rješavanje problema u hodu. Ova fleksibilnost skraćuje vrijeme reakcije i povećava produktivnost.

Analitički alati takođe igraju značajnu ulogu u ovom procesu, pomažući menadžmentu da prati performanse i identifikuje obrasce u radu, što vodi ka optimizaciji operacija. Na primjer, analiza vremena potrebnog za izvršenje radnih naloga može ukazati na potencijalna uska grla u poslovanju, što omogućava brže donošenje odluka. Automatizacija, koja koristi vještačku inteligenciju za unapređenje procesa, takođe je važna komponenta. Automatsko generisanje radnih naloga na osnovu prethodnih aktivnosti smanjuje potrebu za manuelnim unosom podataka, čime se povećava preciznost i smanjuje mogućnost grešaka.

U konačnici, tehnologija ne samo da poboljšava efikasnost, već i unapređuje ukupno korisničko iskustvo. Razvoj aplikacija koje integrišu ove inovacije donosi prednosti kao što su brža usluga i bolja komunikacija unutar tima, čime se stvara dodatna vrijednost za organizaciju. U nastavku su neki primjeri:

1. Mobilne aplikacije: Mnoge kompanije koriste aplikacije kao što su ServiceTitan ili FieldAware. Ove aplikacije omogućavaju tehničarima na terenu da unose podatke o radnim nalogima, fotografišu oštećenja i bilježe vrijeme rada. Informacije se odmah sinhronizuju sa centralnom bazom podataka, čime se smanjuje rizik od grešaka i omogućava brzo praćenje statusa.
2. Analitički alati: Korišćenje analitičkih alata kao što su Tableau ili Power BI omogućava menadžmentu da vizualizuje podatke o radnim nalogima. Na primjer, analiza vremena potrebnog za završavanje različitih vrsta radnih naloga može otkriti obrasce koji pokazuju gde se često javljaju zastoje. Ove informacije pomažu menadžmentu da donese odluke o obuci zaposlenih ili unapređenju opreme.
3. Automatizacija: Aplikacije kao što su Zapier ili Automate.io omogućavaju automatizaciju radnih tokova. Na primjer, kada se unese novi radni nalog, automatizovani proces može poslati obavještenje svim relevantnim članovima tima putem e-pošte ili mobilnih aplikacija, čime se smanjuje potreba za manuelnim obavještanjem i povećava brzina reakcije.
4. Sistem za upravljanje zadacima: Softver poput Trello ili Asana koristi se za praćenje radnih naloga i zadataka. Korisnici mogu kreirati kartice za svaki radni nalog, dodavati komentare, postavljati rokove i pratiti napredak, čime se poboljšava komunikacija unutar tima i olakšava upravljanje prioritetima.
5. Integracija sa IoT uređajima: Korišćenje Internet of Things (IoT) tehnologija može dodatno unaprediti evidenciju radnih naloga. Na primjer, senzori na opremi mogu automatski slati obavještenja o potrebnim servisima ili popravkama, čime se smanjuje potreba za manuelnim praćenjem stanja opreme.

3.5. Analiza korisničkih potreba

Razumjevanje potreba korisnika predstavlja ključni dio procesa razvoja aplikacije za evidenciju radnih naloga i servisnih listova. Korisnici, uključujući tehničare, menadžere i administrativno osoblje, imaju različite zahtjeve koji direktno utiču na efikasnost i zadovoljstvo prilikom korišćenja aplikacije. Jedan od prvih koraka u ovoj analizi je identifikacija ključnih korisnika i njihovih specifičnih ciljeva. Na primjer, tehničari često traže jednostavan način za unos radnih naloga, dok menadžeri žele sveobuhvatan pregled statusa radnih zadataka. Ova razlika pomaže u razumjevanju kako bi aplikacija mogla zadovoljiti raznovrsne interese. Prikupljanje povratnih informacija je takođe ključno za razumjevanje potreba. Kroz ankete i intervju sa korisnicima, moguće je identifikovati koje funkcionalnosti su najvažnije, kao što su jednostavnost korišćenja, automatska obaveštenja i mogućnost generisanja izveštaja. Ove informacije omogućavaju razvojnim timovima da se fokusiraju na ključne aspekte koji će poboljšati korisničko iskustvo.

Testiranje prototipa predstavlja još jedan važan korak, jer omogućava korisnicima da interaguju sa aplikacijom pre nego što postane konačna verzija, pružajući dragocjene uvide koji mogu dovesti do značajnih poboljšanja. Na osnovu povratnih informacija, tim može vršiti iteracije, dodavati nove funkcionalnosti i prilagođavati dizajn kako bi se bolje uklopio u potrebe korisnika.

Na kraju, važno je napomenuti da analiza korisničkih potreba nije jednokratni proces. Kako se organizacija razvija, tako se i potrebe korisnika mogu menjati. Kontinuirano prikupljanje povratnih informacija i prilagođavanje aplikacije tim promjenama ključno je za dugoročnu uspješnost rješenja.

4. PLANIRANJE I DIZAJN APLIKACIJE

4.1. Analiza zahtjeva korisnika

Analiza korisničkih zahtjeva predstavlja ključnu fazu u razvoju aplikacije jer osigurava da su svi aspekti potreba korisnika pažljivo razmotreni prije nego što se pristupi dizajnu i implementaciji. Ovaj proces obuhvata dubinsko razumijevanje potreba korisnika, koje se mogu prikupiti različitim metodama, iako u ovom slučaju nije bilo direktnih konsultacija. Umjesto formalnih razgovora, analiza se zasnivala na postojećem znanju i iskustvima članova tima. Istraživanjem postojećih rješenja i analizom povratnih informacija iz prethodnih projekata, identifikovane su ključne oblasti koje zahtijevaju poboljšanje. Na osnovu toga, definisani su osnovni zahtjevi za novu aplikaciju za praćenje radnih naloga. Jedan od glavnih funkcionalnih zahtjeva bio je omogućiti jednostavan unos radnih naloga. Korisnici često traže brzi i efikasni način za unos podataka, što im omogućava da se više fokusiraju na sam rad, a manje na administraciju. Interfejs bi trebao biti intuitivan i omogućiti brzo unošenje svih potrebnih informacija, kao što su naziv naloga, opis zadatka, datum, vrijeme, i kontakt podaci klijenata.

Pored funkcionalnih, identifikovani su i nefunkcionalni zahtjevi, poput brzine i efikasnosti rada aplikacije. Korisnici često rade pod vremenskim pritiskom, stoga je važno da aplikacija bude optimizovana za brzo učitavanje i izvršavanje zadataka. Sigurnost podataka također je bila od primarne važnosti, posebno kad je riječ o osjetljivim informacijama, poput ličnih podataka klijenata ili internih poslovnih podataka. U analizi su uzete u obzir i karakteristike drugih sličnih aplikacija, kao i korisnički interfejsi koji su bili uspješni ili izazivali frustracije kod korisnika. Ovi uvidi omogućili su timu da prepozna ključne funkcionalnosti koje aplikacija treba imati, poput naprednih alata za pretragu, lakšu organizaciju informacija i filtriranje radnih naloga. Na osnovu prikupljenih podataka, kreirana je tehnička specifikacija koja će voditi proces razvoja, s ciljem da konačni proizvod zadovolji očekivanja korisnika.

Tim je postavio listu prioriteta funkcionalnosti za prvu verziju aplikacije, uz planove za buduća poboljšanja na osnovu povratnih informacija korisnika. Ovaj proces analize korisničkih zahtjeva osigurava da će razvoj aplikacije biti usmjeren na stvarne potrebe korisnika, čime se povećava vjerovatnoća uspjeha finalnog proizvoda.

4.2. Vizuelni dizajn korisničkog interfejsa

Vizuelni dizajn korisničkog interfejsa (UI) jedan je od ključnih elemenata koji oblikuje korisničko iskustvo u aplikaciji za upravljanje radnim nalogima, servisnim listovima, šifrarnicima komitenata i rezervnim dijelovima. Zbog složenosti zadataka koje korisnici obavljaju, interfejs je kreiran s naglaskom na jednostavnost i intuitivnost, omogućavajući lak pristup svim funkcijama. Posebna pažnja posvećena je preglednosti i jasnoći, gdje su svi elementi pažljivo organizovani kako bi korisnici lako pronašli željene opcije. Dizajn je prilagođen ključnim funkcionalnostima kao što su unos radnih naloga, pregled servisnih listova te upravljanje šifrarnicima komitenata i rezervnih dijelova. Na primjer, korisnici mogu brzo i jednostavno unijeti podatke poput broja naloga, opisa i datuma, zahvaljujući preglednim i jasno definisanim sekcijama koje olakšavaju unos informacija. Ovakav pristup smanjuje mogućnost grešaka i povećava efikasnost rada. Proces testiranja dizajna s krajnjim korisnicima bio je ključan za postizanje optimalnog korisničkog iskustva. Na temelju povratnih informacija,

interfejs je prilagođavan kako bi se osigurala jednostavnost korištenja, uz istovremeno ispunjavanje svih tehničkih zahtjeva.

Sveukupno, vizuelni dizajn aplikacije razvijen je kako bi bio funkcionalan, pregledan i intuitivan, osiguravajući korisnicima brz i jednostavan pristup alatima za upravljanje radnim nalogima, servisnim listovima, šifrnica komitenata i rezervnim dijelovima, čime se poboljšava cjelokupna efikasnost aplikacije.

4.3. Priprema za testiranje i prikupljanje povratnih informacija

Ovaj dio procesa od suštinske je važnosti za osiguravanje da aplikacija ne samo da zadovoljava funkcionalne zahtjeve, već i pruža pozitivno korisničko iskustvo. Ključni korak u pripremi za testiranje je definiranje ciljeva koji će se postići tim procesom. Ovi ciljevi mogu uključivati procjenu intuitivnosti korisničkog interfejsa, jednostavnost navigacije, brzinu učitavanja te opće zadovoljstvo korisnika aplikacijom. Jasno postavljeni ciljevi omogućuju timu da se fokusira na ključne aspekte koje treba ocijeniti tijekom testiranja. U svrhu poboljšanja ovog procesa, profesor je dostavio obrasce koji su pomogli u prikupljanju potrebnih podataka. Ovi obrasci omogućili su sistematičan pristup prikupljanju povratnih informacija i olakšali analizu rezultata. Nakon postavljanja ciljeva i dobivanja obrazaca, sljedeći korak je odabir odgovarajuće metode za prikupljanje povratnih informacija. Postoji nekoliko pristupa koji se mogu koristiti, uključujući ankete, intervjue, fokus grupe ili analizu podataka o korištenju aplikacije. Ove metode pružaju važne uvide u korisnička iskustva, omogućavajući timu da prepozna potencijalne probleme i prilagodi dizajn gdje je potrebno. Osim toga, važno je uključiti raznoliku grupu korisnika kako bi se osigurala različitost perspektiva. Korisnici s različitim razinama tehničkog znanja, iskustvom i specifičnim potrebama mogu pružiti uvide koje ne bi bilo moguće dobiti od jedne homogenizirane grupe. To pomaže u boljem razumijevanju kako aplikacija funkcionira u stvarnim uvjetima te koje funkcionalnosti imaju najveću važnost za krajnje korisnike.

4.4. Prednosti i izazovi korisničkog iskustva

Korisničko iskustvo (UX) igra ključnu ulogu u uspjehu svake aplikacije, uključujući i aplikaciju za evidenciju radnih naloga i servisnih listova. Poboljšanje korisničkog zadovoljstva i olakšavanje interakcije sa aplikacijom donosi brojne prednosti, ali također uključuje izazove koji zahtjevaju pažljivo planiranje i strategiju. Jedna od najvećih prednosti kvalitetnog UX dizajna jeste povećanje produktivnosti. Kada je interfejs logički organizovan, korisnici mogu brže i efikasnije obavljati zadatke kao što su unos radnih naloga ili pregled rezervnih delova. Na primjer, jasno istaknute funkcionalnosti omogućavaju korisnicima da lakše pronađu potrebne informacije, što smanjuje gubljenje vremena i ubrzava proces rada. Zadovoljstvo korisnika direktno utiče na njihovu produktivnost. Kada je aplikacija jednostavna za korišćenje i radi bez problema, korisnici mogu obavljati zadatke sa većom efikasnošću. Na primjer, brzi unos podataka i laka dostupnost relevantnih informacija mogu značajno skratiti vrijeme potrebno za izvršenje zadataka. Dobro optimizovan korisnički interfejs može smanjiti broj grešaka koje korisnici prave, što rezultira tačnijim unosom podataka i manjom potrebom za kasnijim ispravkama. Jedan od važnih elemenata dobrog UX dizajna su vizuelni vodiči i upute, koji korisnicima pomažu da pravilno koriste aplikaciju. To mogu biti objašnjenja za svako polje unosa ili kontekstualni saveti koji se pojavljuju pri prelasku mišem preko određenih elemenata,

čime se smanjuje rizik od grešaka i povećava preciznost unosa. Osim toga, kvalitetan korisnički interfejs doprinosi većem zadovoljstvu korisnika. Kada je aplikacija vizuelno privlačna i jednostavna za upotrebu, korisnici su skloniji njenom redovnom korišćenju. Ovo povećano zadovoljstvo može doprineti lojalnosti korisnika i pozitivnim preporukama, što je ključno za dugoročni uspeh aplikacije na tržištu. Međutim, postoje i izazovi u dizajniranju korisničkog iskustva. Jedan od najvećih izazova je različit nivo tehničke pismenosti korisnika. S obzirom na to da korisnici dolaze s različitim nivoima znanja i iskustva, može biti teško razviti univerzalno rješenje koje odgovara svima. U ovakvim situacijama važno je obezbediti podršku, kao što su uputstva i obuke, kako bi se korisnici lakše prilagodili novom sistemu. Takođe, izazov je i stalna promena potreba korisnika i tehnologije. Potrebe korisnika evoluiraju, a tehnologija napreduje, zbog čega je važno redovno evaluirati korisničko iskustvo i prilagođavati aplikaciju novim zahtjevima i povratnim informacijama. Jedan od ključnih izazova u razvoju interfejsa je postizanje ravnoteže između bogate funkcionalnosti i jednostavnosti korišćenja. Ponekad, dodavanje previše funkcionalnosti može preopteretiti korisnički interfejs, što može dovesti do konfuzije i smanjiti efikasnost korisnika. Zbog toga je važno kreirati interfejs koji nudi sve potrebne opcije, ali je i dalje pregledan i lako razumljiv.

Konačno, prikupljanje povratnih informacija od korisnika je ključno za kontinuirano unapređenje UX-a. Testiranje sa stvarnim korisnicima, ankete i analiza korišćenja aplikacije omogućavaju timovima da identifikuju probleme i prilagode interfejs kako bi bolje odgovarao potrebama korisnika.

4.5. Planiranje budućeg razvoja aplikacije

Planiranje budućeg razvoja aplikacije za evidenciju radnih naloga i servisnih listova, kao i šifarnika komitenata i rezervnih delova, ključno je za njen dugoročni uspeh. Nakon inicijalne implementacije, aplikacija mora biti spremna na stalno prilagođavanje promenama u poslovnim potrebama i tehnologijama. Važno je kontinuirano pratiti tehnološke trendove jer se alati i metode brzo razvijaju. Na primjer, integracija vještačke inteligencije može značajno unaprediti aplikaciju, omogućavajući automatske preporuke za rezervne delove ili optimizaciju radnih naloga na osnovu prikupljenih podataka. Tehnologije poput analitike podataka mogu doprineti boljem praćenju performansi aplikacije, identifikaciji potencijalnih problema i optimizaciji procesa, dok unapređenja u oblasti sigurnosti mogu zaštititi osetljive poslovne podatke. Razumevanje potreba korisnika takođe igra ključnu ulogu u planiranju. Redovno prikupljanje povratnih informacija omogućava prilagođavanje aplikacije prema zahtjevima korisnika. Kroz ankete, intervjuue i testiranja može se identifikovati koje nove funkcionalnosti korisnici žele ili kako poboljšati postojeće, čime se poboljšava njihovo iskustvo. Na primjer, korisnici mogu zatražiti dodatne izveštaje ili jednostavnije praćenje statusa radnih naloga, što treba uzeti u obzir pri budućim nadogradnjama.

Jedan od ključnih izazova u razvoju aplikacije je održavanje ravnoteže između funkcionalnosti i jednostavnosti korišćenja. Iako je važno ponuditi što više opcija, interfejs mora ostati intuitivan kako bi korisnici lako obavljali zadatke. Planiranje unapređenja interfejsa treba da bude vođeno stvarnim potrebama korisnika i fokusirano na smanjenje složenosti, a sve sa ciljem povećanja produktivnosti i smanjenja grešaka. Osim toga, planiranje mora uključiti strategiju održavanja i podrške. Nakon puštanja aplikacije u rad, potrebno je obezbediti da se redovno ažurira kako bi bila usklađena s promjenama u poslovanju ili tehnologijama. Kvalitetna

korisnička podrška može značajno doprinjeti boljem iskustvu, dok redovne nadogradnje osiguravaju stabilnost i funkcionalnost aplikacije.

Sve ove aktivnosti treba pažljivo uskladiti s budžetskim planom kako bi razvoj aplikacije mogao teći nesmetano. Razvijanje realnog budžeta omogućava efikasno upravljanje resursima, a time i uspješnu implementaciju novih funkcionalnosti bez prekida u radu aplikacije. Dugoročno gledano, ovakav pristup osigurava da aplikacija raste zajedno s potrebama korisnika i promjenama u tehnološkom okruženju.

4.6. Razvoj i unapređenje korisničkog iskustva

Tokom razvoja aplikacije, prikupljanje povratnih informacija i unapređenje korisničkog iskustva postalo je ključan deo procesa. Iako nisu korišćene formalne ankete, oslonac su bili profesor i kolege, što je omogućilo da se bolje razumije potreba korisnika. Kroz razgovore saznalo se koje funkcionalnosti su im najvažnije i gdje aplikacija može biti poboljšana. Ova direktna komunikacija sa korisnicima pružila je dragocjene uvide o tome kako aplikacija funkcioniše u stvarnim uslovima. Na primjer, korisnici su izrazili potrebu za lakšim unosom i pregledom radnih naloga, kao i za boljom organizacijom podataka u šifrniku komitenata i rezervnih delova. Ovi komentari su omogućili da se reorganizuje interfejs aplikacije kako bi olakšali pristup najčešće korišćenim funkcijama, čime se unaprijedila efikasnost i intuitivnost aplikacije. Razgovori sa profesorom i kolegama dodatno su pomogli da se sagleda šira slika i identifikuju mogući pristupi za rješavanje izazova sa kojima su se korisnici suočavali. Njihovo iskustvo u radu na sličnim projektima dalo je smjernice koje su omogućile da aplikacija ne samo zadovolji tehničke zahtjeve, već i da bude prijatna i laka za korišćenje.

Iako formalne ankete nisu bile dio ovog procesa, neformalne konsultacije su bile izuzetno korisne. Ovaj pristup je omogućio da brzo se reaguje na potrebe korisnika i prilagodi aplikaciju njihovim očekivanjima, čime je poboljšana njena funkcionalnost i korisničko iskustvo.

4.7. Samostalno testiranje aplikacije i njegov značaj

Tokom razvoja aplikacije za evidenciju radnih naloga, samostalno testiranje igralo je ključnu ulogu u osiguranju njene funkcionalnosti i kvaliteta korisničkog iskustva. Iako nije bilo organizovanog testiranja sa krajnjim korisnicima, samostalno testiranje omogućilo je detaljan pregled svih funkcionalnosti i identifikaciju eventualnih problema. Testiranje je sprovedeno u različitim fazama, s fokusom na osnovne funkcionalnosti poput unosa radnih naloga, prikaza servisnih listova i rada sa šifrnica komitenata i rezervnih delova. Pažljivo su se analizirali svi aspekti aplikacije kako bi se identifikovale moguće prepreke za krajnje korisnike, uključujući i funkcionisanje aplikacije na različitim uređajima. Na primjer, testirana je brzina učitavanja stranica i ispravno prikazivanje elemenata interfejsa. Tokom ovog procesa, uočeni su manji problemi koji su odmah ispravljeni, što je dovelo do značajnog poboljšanja upotrebljivosti aplikacije. Posebna pažnja je posvećena korisničkom iskustvu, postavljajući pitanje poput: "Da li su informacije lako dostupne?" Ovakav pristup pomogao je u oblikovanju interfejsa koji je jednostavan i intuitivan, omogućavajući korisnicima brz pristup ključnim funkcijama.

5. IMPLEMENTACIJA APLIKACIJE

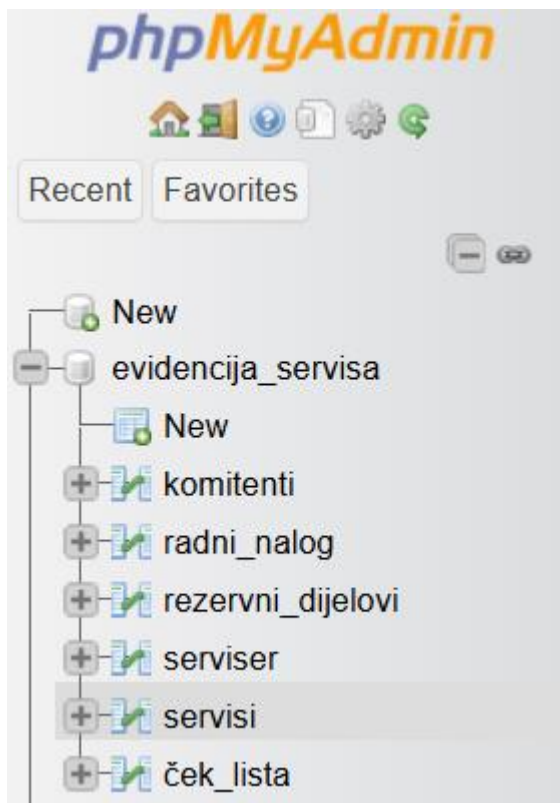
5.1. Struktura baze podataka

Baza podataka predstavlja osnovu svakog informacionog sistema i ključna je za pravilno skladištenje i upravljanje podacima. U ovom projektu, MySQL baza podataka se koristi za čuvanje svih relevantnih informacija o komitentima, rezervnim dijelovima, radnim nalogima, i servisnim listovima. U okviru ovog sistema, kreirano je više tabela koje su međusobno povezane i omogućavaju efikasan rad aplikacije.

Glavne tabele uključuju:

- **Komitenti:** id, naziv_komitenta, adresa, kontakt_osoba, telefon, email;
- **Rezervni dijelovi:** id, naziv_dijela, sifra_dijela, opis_dijela, cijena, količina;
- **Radni nalog:** id, broj_naloga, datum_narudzbine, plan, nalog_izdao_odobrio, naručilac, ref_nar, kontakt, zahtjev_naručioca, serviser, prihod, odhod, čas_na_putu, radni_čas, vozilo, polazak_km, povratak_km, km, servisni_uređaj, rad_sati, opter_sati, br_vklopov, pritisak_sis, temp_bloka, lopt_A, iras_a, ulje, promjena_ulja, vrsta_servisa, napomene_greške, specifikacija_radova, potpis_servisera, potpis_naručioca, uređaj, datum;
- **Ček lista:** id, datum, tip_kompresora, serijski_broj, temperatura_ulja, temperatura_vode, izlazna_temperatura_zraka, provjera_sigurnosti_ventila, provjera_elektro_komponenti, podmazivanje_ležajeva_pogonskog_elektromotora, pregled_zračnih_prolaza_elektromotora, pregled_predfiltera_na_kućištu_kompresora, pregled_razine_ulja_i_dolivanje_po_potrebi, pregled_kompresorskog_ulja, pregled_filtera_ulja, pregled_filtera_zraka, pregled_elementa_separatora_ulja, pregled_pneumatskog_cilindra_usisnog_ventila, pregled_elektromagnetnog_rasteretnog_ventila, pregled_ventila_minimalnog_pritiska, pregled_termostatskog_ventila, SPM_i_UZV_mjerenje_vibracija, pregled_stanja_sklopa_rashladnog_ventilatora, provjera_ispravnosti_odvajača_kondenzata, provjera_funkcionalnosti_hladnjaka_ulja_zraka, uzrokovanje_i_analiza_kompresorskog_ulja, provjera_funkcionalnosti_rada_sušača, ispušt_kondenzata_iz_boce, posebni_prijedlozi_ili_dodatne_radove, potpis_servisera, potpis_naručioca;
- **Serviser:** id, username, password, created_at;
- **Servisi:** id, naziv_komitenata, datum_servisa, opis_usluge, cijena, napomena, status_servisa.

U prilogu su slike da se prikaže kako zapravo izgleda ta struktura u MySQL:



Slika 6: Prikaz strukture baze

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id 🔑	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 naziv_komitenta	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 adresa	varchar(150)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	4 kontakt_osoba	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 telefon	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	6 email	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Slika 7: Prikaz tabele komitenti

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id 🔑	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 naziv_dijela	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 sifra_dijela	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	4 opis_dijela	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 cijena	decimal(10,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	6 kolicina	int(11)			Yes	NULL			Change Drop More

Slika 8: Prikaz tabele rezervni dijelovi

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 br_naloga	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 datum_narudzbine	date			Yes	NULL			Change Drop More
<input type="checkbox"/>	4 plan	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 nalog_izdao_odobrio	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	6 narucilac	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	7 ref_nar	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	8 kontakt	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	9 zahtjev_narucioca	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	10 serviser	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	11 prihod	decimal(10,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	12 odhod	decimal(10,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	13 cas_na_putu	decimal(5,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	14 radni_cas	decimal(5,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	15 vozilo	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	16 polazak_km	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/>	17 povratak_km	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/>	18 km	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/>	19 servisni_uredjaj	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	20 rad_sati	decimal(5,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	21 opter_sati	decimal(5,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	22 br_vklopov	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/>	23 pritisak_sis	decimal(5,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	24 temp_bloka	decimal(5,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	25 lopt_A	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	26 iras_a	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	27 ulje	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	28 promjena_ulja	tinyint(1)			Yes	NULL			Change Drop More
<input type="checkbox"/>	29 vrsta_servisa	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	30 napomene_greske	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	31 specifikacija_radova	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	32 potpis_servisera	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	33 potpis_narucioca	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	34 uredjaj	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	35 datum	date			Yes	NULL			Change Drop More

Slika 9: Prikaz tabele radni nalog

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 datum	date			No	None			Change Drop More
<input type="checkbox"/>	3 tip_kompresora	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 serijski_broj	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 temperatura_ulja	decimal(5,2)			No	None			Change Drop More
<input type="checkbox"/>	6 temperatura_vode	decimal(5,2)			No	None			Change Drop More
<input type="checkbox"/>	7 izlazna_temperatura_zraka	decimal(5,2)			No	None			Change Drop More
<input type="checkbox"/>	8 provjera_sigurnosti_ventila	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	9 provjera_elektro_komponenti	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	10 podmazivanje_lezajeva_pogonskog_elektromotora	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	11 pregled_zračnih_prolaza_elektromotora	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	12 pregled_predfiltera_na_kućištu_kompresora	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	13 pregled_razine_ulja_i_dolivanje_po_potrebi	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	14 pregled_kompresorskog_ulja	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	15 pregled_filtera_ulja	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	16 pregled_filtera_zraka	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	17 pregled_elemenata_separatora_ulja	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	18 pregled_pneumatskog_cilindra_uisnog_ventila	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	19 pregled_elektromagnetnog_rasteretnog_ventila	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	20 pregled_ventila_minimalnog_pritiska	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	21 pregled_termostatskog_ventila	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	22 SPM_i_UZV_mjerenje_vibracija	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	23 pregled_stanja_sklopa_rashladnog_ventilatora	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	24 provjera_ispravnosti_odvajaca_kondenzata	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	25 provjera_funkcionalnosti_hladnjaka_ulja_zraka	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	26 uzrokovanje_i_analiza_kompresorskog_ulja	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	27 provjera_funkcionalnosti_rada_susača	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	28 ispust_kondenzata_iz_boce	tinyint(1)			No	None			Change Drop More
<input type="checkbox"/>	29 posebni_prijedlozi_ili_dodatne_radove	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	30 potpis_servisera	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	31 potpis_naručioca	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Slika 10: Prikaz tabele ček lista

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 naziv_komitenta	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 datum_servisa	date			Yes	NULL			Change Drop More
<input type="checkbox"/>	4 opis_usluge	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 cijena	decimal(10,2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	6 napomena	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	7 status_servisa	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Slika 11: Prikaz tabele servisi

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 created_at	timestamp			No	current_timestamp()			Change Drop More

Slika 12: Prikaz tabele serviser

5.2. Razvoj backenda u PHP-u

Backend sistema razvijen je u PHP-u, koji je poznat po svojoj efikasnosti u web razvoju i lakoj integraciji s MySQL bazama podataka. PHP se koristi za obradu podataka sa forma, uspostavljanje veze s bazom podataka, izvršavanje SQL upita i prikazivanje rezultata korisnicima. Prvi korak u razvoju backenda bio je implementacija mehanizma za povezivanje s bazom podataka. Za tu svrhu korišćen je fajl db_connect.php, koji osigurava konekciju između PHP skripte i MySQL baze. U ovom fajlu korišćena je funkcija `mysqli_connect()`, koja omogućava povezivanje sa serverom baze podataka pomoću odgovarajućih kredencijala, uključujući korisničko ime, lozinku i ime baze. Ovaj pristup osigurava stabilnu i sigurnu interakciju između aplikacije i baze podataka, što je ključno za pravilno funkcionisanje celog sistema.

```
<?php
// Povezivanje s MySQL bazom podataka
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "evidencija_servisa";

$conn = new mysqli($servername, $username, $password, $dbname);
```

Slika 13: Povezivanje sa MySQL bazom podataka

Jedan od ključnih elemenata PHP backenda je logika unosa podataka. Kada korisnik popuni formu za unos novog radnog naloga, podaci se šalju PHP skripti koja ih obrađuje. Ova skripta najpre proverava validnost unetih podataka, a zatim kreira odgovarajući SQL upit i izvršava ga koristeći funkciju `mysqli_query()`. Na taj način, informacije iz forme se sačuvavaju u odgovarajućoj tabeli u bazi podataka. Pored unosa podataka, backend je odgovoran i za prikaz informacija korisnicima. Kada, na primjer, korisnik želi da pregleda sve unose iz tabele radnih naloga, PHP skripta generiše SQL upit koji preuzima sve zapise iz te tabele i prikazuje ih u

formi tabele na korisničkom interfejsu. Ova interakcija između PHP-a i MySQL baze omogućava korisnicima lako pretraživanje i upravljanje podacima.

Takođe, u slučaju da korisnik unese nevalidne informacije ili izostavi obavezne podatke, PHP skripta može generisati poruku greške koja traži ponovni unos ispravnih informacija. Nakon uspješne validacije podataka, PHP koristi SQL upite da unese te podatke u MySQL bazu, čime se osigurava tačnost i integritet informacija u sistemu.

```
$stmt = $conn->prepare("INSERT INTO radni_nalog
(br_naloga, datum_narudzbine, plan, nalog_izdao_odobrio, narucilac, ref_nar, kontakt, zahtjev_narucioca, serviser, prihod,
odhod, cas_na_putu, radni_cas, vozilo, polazak_km, povratak_km, km, servisni_uredjaj, rad_sati, opter_sati, br_vklopov, uredjaj,
pritisak_sis, temp_bloka, lopt_A, iras_a, ulje, promjena_ulja, vrsta_servisa, napomene_greske, specifikacija_radova,
potpis_servisera, potpis_narucioca, datum)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
```

Slika 14: Prikaz korišćenja funkcije prepare

Ovaj način unosa podataka koristi prepared statements, što predstavlja najbolju praksu jer obezbjeđuje dodatnu sigurnost i efikasnost prilikom interakcije sa bazom podataka, naročito kada se radi sa korisničkim unosima. Ovaj pristup uspješno sprečava SQL injection napade i poboljšava performanse kada se rukuje velikim količinama podataka.

Takođe, korišćena je funkcija bind_param(), koja je ključna za rad sa prepared statements u MySQLi. Ova funkcija povezuje promjenljive koje sadrže vrijednosti sa parametrima unutar SQL upita, čime se osigurava pravilno povezivanje i unos vrednosti u bazu podataka. U ovom kodu, primjenjen je bind_param() da odredimo tipove podataka i povežemo ih s odgovarajućim mjestima u SQL upitu. Ovaj korak je od suštinskog značaja jer garantuje sigurnost i validnost unosa podataka.

```
$stmt->bind_param('ssssssssssssssssssssssssssss',
$br_naloga, $datum_narudzbine, $plan, $nalog_izdao_odobrio, $narucilac, $ref_nar, $kontakt, $zahtjev_narucioca, $serviser, $prihod,
$odhod, $cas_na_putu, $radni_cas, $vozilo, $polazak_km, $povratak_km, $km, $servisni_uredjaj, $rad_sati, $opter_sati, $br_vklopov,
$uredjaj, $pritisak_sis, $temp_bloka, $lopt_A, $iras_a, $ulje, $promjena_ulja, $vrsta_servisa, $napomene_greske, $specifikacija_radova,
$potpis_servisera, $potpis_narucioca, $datum);
```

Slika 15: Prikaz korišćenja funkcije bind_param()

5.3. Razvoj frontend-a (HTML, CSS)

Korisnički interfejs (UI) aplikacije igra ključnu ulogu u omogućavanju korisnicima jednostavne i efikasne interakcije sa sistemom. U ovoj aplikaciji, UI je osmišljen sa posebnim naglaskom na intuitivnost, jasnoću i lakoću korišćenja. Forme su organizovane tako da su slična polja grupisana u blizini, čime se smanjuje vrijeme potrebno za unos podataka. HTML (Hypertext Markup Language) se koristi za strukturu web stranica, a u ovoj aplikaciji se primjenjuje za kreiranje formi i prikaz podataka. Svaka komponenta korisničkog interfejsa je pažljivo

izrađena, tako da je jasno definisana i lako razumljiva, čime se dodatno poboljšava korisničko iskustvo.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Radni Nalog</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <div class="container">
    <header class="header">
      <div class="company-info">
        <p>Pađen doo, Banjaluka</p>
        <p>Modrička 15, 78000 Banjaluka</p>
        <p>Tel: 066/858-309 spadjen@gmail.com</p>
      </div>
      <h1 class="title">RADNI NALOG</h1>
      <p class="subtitle">PUTNI NALOG</p>
    </header>

    <form action="../includes/submit_radni_nalog.php" method="POST">

      <div class="form-section">
        <div class="form-row">
          <label for="br_naloga">BR. NALOGA:</label>
          <input type="text" id="br_naloga" name="br_naloga">
          <label for="datum_narudzbine">DATUM NARUDŽBINE:</label>
          <input type="date" id="datum_narudzbine" name="datum_narudzbine">
          <label for="plan">PLAN:</label>
        </div>
      </div>
    </form>
  </div>
</body>
</html>
```

Slika 16: Prikaz djela HTML koda

U dokumentu se najprije koristi deklaracija koja jasno označava da se radi o HTML5 formatu, što je ispravno.

Zatim počinje HTML dokument sa oznakom koja definiše jezik stranice kao engleski.

U dijelu koji se zove "glava" (head) nalaze se metapodaci, kao što su karakter set, naslov stranice i link ka eksternoj CSS datoteci. Ovi elementi omogućavaju pravilno prikazivanje i stilizaciju stranice.

U "tijelu" stranice (body) nalaze se svi vidljivi elementi, uključujući tekst, slike i forme. U ovom dijelu organizovani su div elementi koji grupišu sadržaj, header koji sadrži informacije o firmi i naslov stranice, kao i forma koja omogućava korisnicima da unose podatke. U formi su prisutna tekstualna polja i polja za unos datuma. U nastavku ću da prikažem dio CSS koda:


```

* {
  box-sizing: border-box;
  font-family: Arial, sans-serif;
}

body {
  margin: 0;
  padding: 0;
  background-color: #f2f2f2;
}

.container {
  width: 80%;
  margin: auto;
  background-color: white;
  padding: 20px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.header {
  text-align: center;
  margin-bottom: 20px;
}

.company-info {
  text-align: left;
  margin-bottom: 20px;
}

```

Slika 17: Prikaz CSS koda

U ovom CSS kodu izvršene su osnovne stilizacije koje značajno poboljšavaju vizuelni doživljaj aplikacije. U nastavku su navedeni ključni aspekti ovog dijela koda:

Globalni stil (*):

*Selektor : Ovaj selektor se odnosi na sve HTML elemente na stranici. Primjena `box-sizing: border-box;` omogućava da se `padding` i `border` uračunavaju u ukupnu širinu i visinu elemenata, čime se olakšava upravljanje njihovim dimenzijama.

Font porodica: Postavljena je univerzalna porodica fontova `font-family: Arial, sans-serif;`, gdje se Arial koristi kao primarni font, a u slučaju nedostupnosti, pretraživač će odabrati prvi raspoloživi "sans-serif" font.

Stilizacija tijela stranice (body):

Margin i padding: Postavke `margin: 0;` i `padding: 0;` uklanjaju standardne margine i padding koje pretraživači obično dodaju stranicama.

Boja pozadine: `background-color: #f2f2f2;` postavlja svijetlosivu boju pozadine na cijeloj stranici, čime se sadržaj unutar stranice vizuelno izdvaja.

Kontejner (.container):

Širina: `width: 80%;` postavlja širinu kontejnera na 80% širine prozora pretraživača, omogućavajući centriranje sadržaja bez preuzimanja cijele širine ekrana.

Centriranje: margin: auto; automatski centriraju kontejner horizontalno unutar stranice.

Boja pozadine: background-color: white; daje belu pozadinu kontejneru, čime se sadržaj ističe na svijetlosivoj pozadini.

Unutrašnji razmak: padding: 20px; dodaje unutrašnje razmake, čineći sadržaj "prozračnijim".

Efekat senke: box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); dodaje blagu senku, što kontejneru daje 3D izgled i čini ga vizuelno odvojenim od pozadine.

Zaglavlje (.header):

Poravnanje teksta: text-align: center; centrirano poravnava tekst unutar zaglavlja, tako da su naslovi, poput "RADNI NALOG" i "PUTNI NALOG", u sredini.

Prostor ispod zaglavlja: margin-bottom: 20px; dodaje prostor ispod zaglavlja, razdvajajući ga od ostatka sadržaja.

Informacije o firmi (.company-info):

Poravnanje teksta: text-align: left; poravnava tekst informacije ulijevo, čime se stvara struktura koja je uobičajena za poslovne dokumente.

Prostor ispod: margin-bottom: 20px; dodaje prostor ispod ovog bloka teksta, kako bi bio vizuelno odvojen od drugih elemenata.

Sada će biti prikazano kako aplikacija funkcioniše:

ČEK LISTA ZA SVE VRSTE RADOVA

Tip kompresora: Serijski broj:

R. Br.	OPIS	Čekirano	Zamjenjeno	Ponuditi novi deo
1	Temperatura ulja, temperatura vode, izlazna temperatura zraka	<input type="text" value="Ulje °C"/>	<input type="text" value="Voda °C"/>	<input type="text" value="Zrak °C"/>
2	Provjera sigurnosti ventila	<input checked="" type="checkbox"/>		
3	Provjera elektro komponenti	<input checked="" type="checkbox"/>		
4	Podmazivanje ležajeva pogonskog elektromotora	<input checked="" type="checkbox"/>		
5	Pregled zračnih prolaza elektromotora	<input checked="" type="checkbox"/>		
6	Pregled predfiltera na kućištu kompresora	<input checked="" type="checkbox"/>		
7	Pregled razine ulja i dolivanje po potrebi	<input checked="" type="checkbox"/>		
8	Pregled kompresorskog ulja	<input checked="" type="checkbox"/>		
9	Pregled filtera ulja	<input type="checkbox"/>		
10	Pregled filtera zraka	<input type="checkbox"/>		
11	Pregled elemenata separatora ulja	<input type="checkbox"/>		
12	Pregled pneumatskog cilindra usisnog ventila	<input type="checkbox"/>		
13	Pregled elektromagnetnog rasteretnog ventila	<input type="checkbox"/>		
14	Pregled ventila minimalnog pritiska	<input type="checkbox"/>		
15	Pregled termostatskog ventila	<input type="checkbox"/>		
16	SPM i UZV mjerenje vibracija zbog praćenja stanja ležajeva vijčanog bloka	<input type="checkbox"/>		
17	Pregled stanja sklopa rashladnog ventilatora	<input type="checkbox"/>		
18	Provjera ispravnosti odvajaa kondenzata	<input type="checkbox"/>		
19	Provjera funkcionalnosti hladnjaka ulja zraka	<input type="checkbox"/>		
20	Uzrokovanje i analiza kompresorskog ulja	<input type="checkbox"/>		
21	Provjera funkcionalnosti rada sušača	<input type="checkbox"/>		
22	Ispust kondenzata iz boce	<input type="checkbox"/>		

Posebni prijedlozi ili dodatne radove: Potpis servisera: Potpis naručioca: Datum:

Slika 18

Podaci su uspešno uneseni!

Slika 19

	id	datum	tip_kompresora	serijski_broj
<input type="checkbox"/> Edit Copy Delete	3	0000-00-00	54ap2	87878

Slika 20

posebni_prijedlozi_ili_dodatne_radove	potpis_servisera	potpis_naručioca
mali servis	SSSSS	SSSSS

Slika 21

Pađen doo, Banjaluka
Modrička 15, 78000 Banjaluka
Tel: 066/858-309 spadjen@gmail.com

RADNI NALOG PUTNI NALOG

BR.NALOGA:	<input type="text" value="74747474"/>	DATUM NARUDBINE:	<input type="text" value="mm/dd/yyyy"/>	PLAN:	<input type="text" value="pivara"/>	NALOG IZDAO/ODOBRIO:	<input type="text"/>										
NARUČILAC:	<input type="text" value="sasa padjen"/>	REF.NAR.:	<input type="text"/>	KONTAKT:	<input type="text" value="99999999"/>												
ZAHTJEV NARUČIOCA:	<input type="text"/>																
SERVISER:	<input type="text"/>	PRIHOD:	<input type="text"/>	ODHOD:	<input type="text"/>	ČAS NA PUTU:	<input type="text"/>	RADNI ČAS:	<input type="text"/>								
VOZILO:	<input type="text"/>	Polazak kilometraža:	<input type="text"/>	Povratak kilometraža:	<input type="text"/>	KM:	<input type="text"/>										
SERVISNI UREĐAJ:	<input type="text"/>	RAD.SATI:	<input type="text"/>	OPTER.SATI:	<input type="text"/>	BR.VKLOPOV:	<input type="text"/>										
UREĐAJ:	<input type="checkbox"/>	PRITISAK SIS.:	<input type="checkbox"/>	TEMP.BLOKA:	<input type="checkbox"/>	loptA:	<input type="checkbox"/>	IRAS/A:	<input type="checkbox"/>	ULJE:	<input type="checkbox"/>	PROMJENA ULJA:	<input type="checkbox"/>	DA	<input type="checkbox"/>	NE	<input type="checkbox"/>
VRSTA SERVISIA:	<input type="text"/>																

Specifikacija radova i specifikacija zamjenjenih djelova

Koda	Opis	Kol	Koda	Opis	Kol
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Napomene-greške:

Specifikacija radova:

Potvrđujem da su sve usluge ispravno izvedene i svi dijelovi ispravno instalirani.

Potpis servisera:

Potpis naručioca:

Datum:

<input type="text"/>	<input type="text"/>	<input type="text" value="10/15/2024"/>
----------------------	----------------------	---

Submit

Slika 22

Radni nalog je uspešno unesen!

Slika 23

	id	br_naloga	datum_narudzbine	plan	nalog_izdao_odobrio	narucilac	ref_nar	kontakt
<input type="checkbox"/>		3 74747474	0000-00-00	pivara		sasa padjen		99999999

Slika 24

A ovo je sad prikaz izvještaja:

Broj Naloga	Datum	Plan
777	0000-00-00	sss
111	2024-02-22	
7474747474	2024-10-15	pivara

Slika 25

Unos Novog Komitenta

Naziv Komitenta:

Adresa:

Kontakt Osoba:

Telefon:

Email:

Slika 26

	id	naziv_komitenta	adresa	kontakt_osoba	telefon	email
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Sasa	Modrička 15	494994944	9876	sasapadjenit@gmail.com

Slika 27

Unos rezervnih dijelova

Naziv dijela:

Šifra dijela:

Opis dijela:

Cijena:

Količina:

Slika 28

	id	naziv_dijela	sifra_dijela	opis_dijela	cijena	kolicina
<input type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	2	auspuh	444444	autp dio	44.99	1

Slika 29

Svakako je dodana i registracija i forma logovanja servisera i forma izvještaja gdje se može vidjeti pregled servisa sedmični, mjesečni, od datuma do datuma, i može se vidjeti posebno servisi rađeni prije više od mjesec dana. Recimo prvo se mora izvršiti registracija nakon toga se uloguje i automatski se nakon logovanja prebacuje na dashboard.php gdje imate razne opcije. U prilogu ce biti prikazano:

Registracija Servisera

Korisničko ime: Lozinka:

Serviser uspešno registrovan!

<input type="checkbox"/> <input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	4 Sale	\$2y\$10\$6sEnjfvVqaHfRbNWrx1uvQUyltZYZW23ODU/zJ3LG...	2024-10-31 15:49:46
--	--------	--	---------------------

Slika 30

Logovanje Servisera

Korisničko ime: Lozinka:

Slika 31

I sada nakon prijave automatski prebacuje na dashboard.php.

Dobrodošli, Sale!

Opcije

- [Unos komitenta](#)
- [Unos rezervnog dela](#)
- [Unos novog servisa](#)
- [Pregled servisa](#)
- [Izveštaji](#)

Izveštaji

- [Sedmični izveštaj](#)
- [Mesečni izveštaj](#)
- [Izveštaj po datumu](#)

Pregled servisa

- [Pregled servisa starijih od mesec dana](#)

Slika 32

I sada se recimo klikne na unos novog servisa:

Unos Novog Servisa

Naziv Komitenta:

Datum Servisa:

Opis Usluge:

Cijena:

Napomena:

Status Servisa:

Slika 33

Novi servis je uspešno dodat!

Tip izveštaja: Od datuma: Do datuma:

ID	Naziv Komitenta	Opis Usluge	Datum Servisa	Cijena	Napomena	Status Servisa
11	volan	mehanika	2024-10-10	68.29	Pažljivo	Završeno

Slika 34

U prilogu će biti prikazano kad se želi vidjeti da je nešto rađeno prije par mjeseci:

Unos Novog Servisa

Naziv Komitenta:

Datum Servisa:

Opis Usluge:

Cijena:

Napomena:

Status Servisa:

Novi servis je uspješno dodat!

Slika 35

ID	Naziv Komitenta	Opis Usluge	Datum Servisa	Cijena	Napomena	Status Servisa
12	provjera opreme	provjera opreme u autu	2024-10-01	60.00	Sve je u redu	Zavrшено

Slika 36

6. ZAKLJUČAK

U ovom diplomskom radu razmatrana je tema razvoja aplikacije za vođenje evidencije o radnim nalogima i servisnim listovima. Aplikacija je izrađena korištenjem savremenih tehnologija, uključujući PHP, MySQL, HTML i CSS. Kroz sistematičan pristup analizi i dizajnu, implementirana je funkcionalnost koja omogućava korisnicima efikasno upravljanje podacima vezanim za radne naloge, čime se unapređuje organizacija i praćenje izvršenih usluga.

Analizirajući zahtjeve korisnika i prateći principe dobrih praksi u razvoju softverskih rješenja, ostvarena je intuitivna navigacija i preglednost korisničkog interfejsa. Osim toga, posvećena je posebna pažnja sigurnosti podataka i optimizaciji performansi sistema. Ova aplikacija ne samo da omogućava brži pristup informacijama, već i olakšava proces donošenja odluka temeljen na dostupnim podacima.

Završetkom ovog projekta stvorena je osnova za budući razvoj i unapređenje aplikacije, uključujući mogućnost dodavanja novih funkcionalnosti prema potrebama korisnika. Planira se i integracija naprednijih analitičkih alata koji će omogućiti detaljnije izvještavanje i analizu podataka.

Na temelju ovog istraživanja, možemo zaključiti da je aplikacija uspješno ispunila postavljene ciljeve i da predstavlja značajan korak ka efikasnijem upravljanju radnim procesima unutar kompanije. Vjerujemo da će njena primjena doprinijeti unapređenju kvaliteta usluga i zadovoljstvu korisnika.

7. LITERATURA

- [1] Beighley, L., & Morrison, M. (2007). Head First PHP & MySQL. O'Reilly Media.
- [2] Freeman, A., & Robson, E. (2015). Head First HTML and CSS. O'Reilly Media.
- [3] Baker, R. (2015). PHP and MySQL Web Development. Addison-Wesley.
- [4] Welling, L., & Thomson, L. (2008). PHP and MySQL Web Development, 4th Edition. Addison-Wesley.
- [5] <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [6] <https://www.w3schools.com/php/>
- [7] <https://www.w3schools.com/css/>
- [8] Lerdorf, R., & Tatro, K. (2013). Programming PHP, 3rd Edition. O'Reilly Media.
- [9] Duckett, J. (2011). HTML & CSS: Design and Build Websites. Wiley.