

Web aplikacija za upravljanje ljudskim resursima

Borić, Rene

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:985789>

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

Download date / Datum preuzimanja: **2025-01-11**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli

Rene Borić

**WEB APLIKACIJA ZA UPRAVLJANJE
LJUDSKIM RESURSIMA**

Završni rad

Pula, kolovoz, 2023.

Sadržaj

1. Uvod	1
2. Korištene tehnologije	2
2.1. React.js	2
2.2. Firebase	4
3. Implementacija projekta	6
3.1. Implementacija	6
3.1.1 React bootstrap	6
3.1.2. React router.....	7
3.2. Struktura aplikacije.....	8
4. Upute.....	17
4.1. Registracija.....	17
4.2. Početak korištenja funkcionalnosti.....	17
4.2.1. Kreiranje zadatka	17
4.2.2. Vođenje evidencije plaće.....	21
4.2.3. Evidencije slobodnih dana	26
4.2.4 Uređivanje informacija na profilu	28
4.2.5. Evidencija važnih dokumenata	30
4.2.6. Kreiranje pravila	30
5. Zaključak.....	33
6. Literatura	34
7. Popis slika	35
Sažetak.....	36
Abstract	37

1. Uvod

Upravljanje ljudskim resursima je vitalni segment svake organizacije. U današnjem dinamičnom i konkurentnom poslovnom okruženju, uspješno upravljanje ljudskim resursima postalo je ključni faktor koji određuje uspjeh i konkurentsku prednost tvrtke. Upravo zato, organizacije sve više prepoznaju važnost implementacije softvera za upravljanje ljudskim resursima kako bi poboljšale učinkovitost i produktivnost svojih HR procesa.

Softveri za upravljanje ljudskim resursima (HR softveri) predstavljaju tehnološke alate dizajnirane za automatizaciju i integraciju ključnih HR funkcija unutar organizacije. Ovi softveri obuhvaćaju širok spektar zadataka, uključujući regrutiranje, zapošljavanje, praćenje radnog vremena, upravljanje plaćama i pogodnostima, obuku i razvoj zaposlenika, evaluaciju performansi te upravljanje informacijama o zaposlenicima.

U posljednjem desetljeću, HR softveri su doživjeli značajan napredak zahvaljujući brzom razvoju informacijske tehnologije. Tradicionalni, papirnati procesi zamijenjeni su digitalnim platformama koje omogućavaju jednostavnu pohranu, upravljanje i pristup HR podacima. Također, moderni HR softveri često uključuju napredne analitičke alate koji organizacijama omogućuju dublji uvid u podatke o zaposlenicima, pružajući osnovu za donošenje informiranih odluka i strategija za upravljanje ljudskim resursima.

Kroz ovaj rad, detaljno ćemo opisati funkcionalnosti ove aplikacije i kako se one integriraju u cjelokupni proces upravljanja ljudskim resursima. Osim toga, analizirat ćemo prednosti i koristi koje ova aplikacija pruža organizacijama, kao i potencijalne izazove koji se mogu pojaviti prilikom implementacije i korištenja.

2. Korištene tehnologije

2.1. React.js

React.js je popularni JavaScript okvir koji se koristi za razvoj korisničkog sučelja. Razvijen od strane Facebooka, React je postao vrlo popularan među programerima zbog svoje efikasnosti, fleksibilnosti i reaktivnosti. Jedna od ključnih karakteristika React-a je njegov pristup komponentnom modelu. Korisničko sučelje se sastoji od malih, ponovno iskoristivih komponenti koje se mogu lako kombinirati kako bi se izgradile kompleksne aplikacije. Svaka komponenta u React-u ima svoje stanje (*eng. state*) i može generirati izlaz (*eng. output*) koji se automatski ažurira kada se stanje promijeni. Ovo omogućuje brzo i efikasno ažuriranje sučelja bez potrebe za potpunim osvježavanjem stranice.

Još jedna značajna osobina React-a je virtualni DOM (*eng. Document Object Model*). React održava virtualnu reprezentaciju stvarnog DOM-a i automatski vrši efikasne promjene samo na dijelovima koji su se promijenili, umjesto da ažurira cijelu stranicu. Ovo omogućuje brže izvođenje aplikacija i bolju performansu. Nadalje, React podržava jednosmjerno povezivanje podataka (*eng. one-way data binding*), što znači da se podaci prenose kroz hijerarhiju komponenti u jednom smjeru. Navedeno olakšava praćenje i upravljanje stanjem aplikacije, što čini kod jednostavnijim za održavanje.

React je također vrlo prilagodljiv i ima bogat ekosustav podrške. Postoje brojne biblioteke i alati koji podržavaju React razvoj, kao što su React Router za upravljanje rutama, Redux za upravljanje stanjem aplikacije i mnogi drugi. Zahvaljujući ovim karakteristikama, React se često koristi za izgradnju modernih web aplikacija, mobilnih aplikacija i čak desktop aplikacija. Također je vrlo popularan u programerskoj zajednici, što znači da postoji obilje resursa i dokumentacije za učenje i rješavanje problema. Kada je u pitanju razvoj korisničkog sučelja u React-u, postoji rasprava između korištenja klasa i funkcionalnih komponenti, a oba pristupa imaju svoje prednosti i razlike, što će biti navedeno u nastavku.

	Klase	Funkcije
Sintaksa	Klase koriste klasičnu sintaksu nasljeđivanja i imaju definiranu strukturu s konstruktorom, metodama i životnim ciklusom. Ovo može biti korisno u složenijim scenarijima gdje je potrebna detaljnija organizacija koda.	Funkcionalne komponente su jednostavne JavaScript funkcije koje su kraće i konciznije. One koriste destrukuiranje i kuke (<i>eng. hooks</i>) za upravljanje stanjem i efektima. Funkcionalne komponente imaju jednostavniju i lakše čitljivu sintaksu.
Upravljanje stanjem	Klase koriste stanje (<i>eng. state</i>) koje se definira u konstruktoru, ažurira se pomoću <code>this.setState</code> metode i pristupa se putem <code>this.state</code> svojstva. Ovo je pogodno za složene scenarije s više stanja i kompleksnijim logikama upravljanja stanjem.	Funkcionalne komponente koriste kuke (<i>eng. hooks</i>) poput <code>useState</code> za upravljanje stanjem. Kuke nude jednostavniju sintaksu i pružaju slične mogućnosti upravljanja stanjem kao i klase. One su posebno dobre za jednostavnije scenarije s jednim stanjem i jednostavnijim logikama.
Životni ciklus	Klase imaju bogatiji skup metoda životnog ciklusa koji se automatski pozivaju u određenim fazama životnog ciklusa komponente. Ovo je korisno za izvršavanje određenih akcija pri montiranju, ažuriranju ili demontiranju komponente.	Funkcionalne komponente koriste kuke poput <code>'useEffect'</code> za postizanje slične funkcionalnosti životnog ciklusa. Kuke su fleksibilne i pružaju kontrolu nad specifičnim aspektima životnog ciklusa. Ovo omogućava bolju kontrolu i optimizaciju performansi.
Performanse	Klase koriste cijeli objekt stanja i ponovno renderiraju cijelu komponentu kada se stanje promijeni. To može dovesti do nepotrebnog re-renderiranja i potencijalno pogoršati performanse.	Funkcionalne komponente s kookama (<i>hooks</i>) mogu imati bolje performanse jer se mogu optimizirati i reagirati samo na promjene specifičnih dijelova stanja. Ovo rezultira manje re-renderiranja komponenti i poboljšanim performansama.

Općenito, funkcionalne komponente s kukama (*hooks*) postaju sve popularniji pristup u React-u zbog svoje jednostavnosti, čitljivosti i boljih performansi. Klase i dalje imaju svoje mjesto, posebno u starijim kodovima ili u situacijama gdje je potrebna složenija logika koju kuke (*hooks*) možda ne mogu pružiti. Važno je razumjeti razlike između klasa i funkcionalnih komponenti kako biste mogli odabrati pristup koji najbolje odgovara vašim potrebama. U oba slučaja, važno je napisati čist i čitljiv kod koji je lako održavati i razumjeti.

Zaključno, React.js je moćan i fleksibilan JavaScript okvir za izgradnju korisničkog sučelja. Njegove karakteristike poput komponentnog modela, virtualnog DOM-a i jednosmjernog povezivanja podataka omogućuju brži i efikasniji razvoj aplikacija.

2.2.Firebase

Firestore je platforma u oblaku koju je razvio Google. Ona pruža širok spektar alata i usluga koji olakšavaju razvoj, implementaciju i upravljanje web i mobilnih aplikacija. Platforma je popularana među programerima jer omogućuje brz i jednostavan način za izgradnju visokokvalitetnih aplikacija.

Firestore je platforma za razvoj aplikacija u oblaku koja pruža nerekacijsku bazu podataka. Nerekacijske baze podataka, ili NoSQL baze podataka, se razlikuju od tradicionalnih relacijskih baza podataka jer ne koriste tabličnu strukturu za organizaciju podataka. Umjesto toga, koriste se različiti modeli podataka poput dokumenta, grafa ili ključ-vrijednost para. Firestore koristi model dokumenta i pruža mogućnost pohrane podataka u JSON formatu. Ova fleksibilnost omogućava developerima da brzo i jednostavno spremaju, pretražuju i ažuriraju podatke. Osim toga, Firestore nudi i druge značajke kao što su autentifikacija korisnika, hosting, analitika i real-time komunikacija. Firestore se ističe svojom jednostavnošću upotrebe, skalabilnošću i brzinom razvoja aplikacija, što ga čini popularnim izborom među razvojnicima.

Iako je MongoDB bila razmatrana kao opcija za aplikaciju, odlučeno je da će se koristiti Firebase iz nekoliko ključnih razloga. Prvo, iako MongoDB nudi moćan NoSQL pristup za rukovanje nestrukturiranim podacima, Firebase je donio više integriranih alata i usluga na jednoj platformi. To je omogućilo brži razvoj jer se nije moralo istraživati i integrirati dodatne alate kao što bi se moralo učiniti s MongoDB-om. Firebase-ov sustav autentifikacije predstavlja ključnu prednost koja je olakšala upravljanje korisničkim pristupom i povećala sigurnost aplikacije. Sustav autentifikacije je omogućio jednostavno i brzo postavljanje autentifikacije korisnika. Firebase također podržava provjeru autentičnosti korisnika na serveru, čime se osigurava da samo ovlašteni korisnici imaju pristup osjetljivim podacima. S obzirom na jednostavno sučelje i dokumentaciju koja je bila lako dostupna, izabran je Firebase. ¹

¹ Comparing Firebase vs MongoDB. Dostupno na <https://www.mongodb.com/firebase-vs-mongodb>

3. Implementacija projekta

3.1. Implementacija

Kao što je već ranije spomenuto, projekt je izrađen u React.js frameworku, ali su također korištene dodatne biblioteke za olakšavanje implementiranja određenih dijelova projekta.

3.1.1 React bootstrap

React-Bootstrap je biblioteka koja pruža gotove komponente i stilove temeljene na Bootstrapu za razvoj React aplikacija. Bootstrap je popularni CSS okvir koji omogućava brz i jednostavan dizajn responzivnih web stranica. Integracija Reacta s Bootstrapom može biti izazovna jer React koristi JSX sintaksu i pristupačnost kroz virtualni DOM. React-Bootstrap rješava ove probleme pružajući predefinirane komponente koje su već povezane s Bootstrap stilovima i funkcionalnostima.

Nadalje, React-Bootstrap olakšava rad s Reactom jer programerima pruža pristup gotovim komponentama koje već imaju stilove i ponašanje definirano. Ove komponente se mogu koristiti kao građevni blokovi za brzu izgradnju sučelja bez potrebe za izradom svih stilova i logike od početka. Također, React-Bootstrap pruža prilagodljive opcije za konfiguriranje komponenti kako bi se zadovoljile specifične potrebe projekta.

Korištenje React-Bootstrapa pomaže u poboljšanju produktivnosti i ubrzava razvoj React aplikacija jer programeri ne moraju trošiti vrijeme na izradu osnovnih komponenti i stilova. Osim toga, React-Bootstrap ima veliku zajednicu i dokumentaciju koja olakšava učenje i rješavanje problema. Kombinacija Reacta i Bootstrapa putem React-Bootstrapa omogućuje programerima da izgrade moderne i responzivne aplikacije brzo i efikasno.

3.1.2. React router

React Router je biblioteka koja olakšava upravljanje rutama i navigacijom u React aplikacijama. Izgrađen je na temelju komponentnog pristupa Reacta i omogućava programerima da definiraju i upravljaju različitim rutama u svojoj aplikaciji.

React Router pruža nekoliko korisnih komponenti poput `<BrowserRouter>`, `<Switch>`, `<Route>` i `<Link>` koje omogućavaju kreiranje dinamičke navigacije i prikazivanje različitih komponenti ovisno o trenutnoj ruti. Korištenjem React Routera, programeri mogu organizirati i strukturirati svoje aplikacije u logičke sekcije i pojednostaviti upravljanje rutama.

Jedna od glavnih prednosti React Routera je da omogućava prikazivanje različitih komponenti na temelju ruta bez potrebe za ponovnim učitavanjem stranice. To pruža bolje korisničko iskustvo i brži prijelaz između različitih dijelova aplikacije. React Router također podržava razne napredne značajke poput ugniježđenih ruta, dinamičkog prikaza ruta, zaštićenih ruta i sl. Ove značajke omogućavaju programerima da izgrade kompleksne i skalabilne aplikacije s fleksibilnom navigacijom i upravljanjem rutama.

Korištenje React Routera olakšava rad s Reactom jer pruža intuitivne i moćne alate za upravljanje rutama i navigacijom u aplikaciji. Programeri mogu lako definirati i organizirati rute, omogućiti navigaciju između različitih dijelova aplikacije te implementirati napredne funkcionalnosti povezane s rutama.

3.2. Struktura aplikacije

Aplikacija koja će se opisivati kroz ovaj rad je osmišljena kao alat namijenjen korištenju u firmama. Proces registracije u aplikaciji počinje s registracijom same firme. Nakon što je firma registrirana, dodjeljuje se jedinstveni kod koji će se koristiti za registraciju radnika kao korisnika ili administratora u toj firmi.

Kod se generira prilikom registracije putem unosa željene e-mail adrese. Nakon registracije, stvara se novi dokument u Firebase bazi podataka za firmu, kao i novi dokument u kolekciji "mail". Kolekcija "mail" koristi se za aktiviranje Firebase ekstenzije nazvane "Trigger Email".

Prilikom registracije, korisnik će dobiti generirani kod putem e-mail poruke na adresu koju je unio. Ova poruka omogućava daljnje korake u postupku registracije. Također, stvaranje novog dokumenta za firmu osigurava da se podaci o registriranoj firmi pravilno pohrane i upravljaju.

Korištenje ekstenzije "Trigger Email" u Firebase-u omogućava automatsko slanje e-mail poruka i aktiviranje funkcionalnosti na temelju unaprijed definiranih događaja. Na taj način, možete jednostavno integrirati i automatizirati slanje e-mail poruka u svoju aplikaciju putem Firebase platforme. Ovaj način registracije omogućuje korisnicima jednostavan pristup generiranom kodu i osigurava da sve potrebne radnje budu izvršene kako bi registracija bila uspješna.

```

async function handleSubmit(e) {
  e.preventDefault();
  setLoading(true);
  await addDoc(collection(db, "firm"), {
    firmname: firmNameRef.current.value,
    id: randomNumber,
  });
  await addDoc(collection(db, "mail"), {
    to: emailRef.current.value,
    message: {
      subject: "Firm registering",
      text: `Thank you for registering your firm. Here is the firm number: ${randomNumber}`,
      html: `
        <h1>Thank you for registering your firm!</h1>
        <p>Below is the firm number:</p>
        <h2>${randomNumber}</h2>
        <p>We appreciate your interest in our platform.</p>
      `,
    },
  });
  setLoading(false);
  toast.success(
    "Firm successfully registered. Firm code has been sent to your e-mail address."
  );
  setTimeout(() => {
    navigate("/");
  }, 5000);
}

```

Slika 1 - Prikazani kod za registriranje firme

Radnici i administratori mogu se registrirati u aplikaciji koristeći dobiveni kod za svoju specifičnu firmu. Ovaj pristup omogućava vašoj aplikaciji da pruži personalizirano iskustvo za svaku firmu, čime se osigurava sigurnost i kontrola pristupa informacijama.

Kreiranje zadataka je jedna od glavnih karakteristika aplikacije. Ovu funkcionalnost imaju samo *admini* firme, koji su odgovorni za vođenje projekata i upravljanje zadacima. *Admini* koriste formu za kreiranje zadataka kako bi unijeli sve relevantne informacije. Ova forma obično uključuje polja za naziv zadatka, opis zadatka i odabir zaduženih radnika u firmi koji će biti odgovorni za izvršenje tog zadatka. Također, moguće je postaviti i prioritet zadatka kako bi se jasno odredila njegova važnost u odnosu na ostale.

Ovaj proces omogućuje *adminima* da detaljno definiraju zadatke i dodijele ih odgovarajućim radnicima. Kroz jasno definirane nazive i opise, *admini* pružaju radnicima potrebne smjernice za izvršenje zadatka. Dodatno, odabirom zaduženih radnika, osigurava se da svaki zadatak bude pridružen odgovarajućem članu tima koji će biti odgovoran za njegovo izvršenje.

S druge strane, radnici u firmi nemaju mogućnost kreiranja novih zadataka (*eng, task*), već samo mogu komentirati postojeće zadatke. Ova funkcionalnost omogućuje radnicima da izraze svoje mišljenje, postavljaju pitanja ili pruže dodatne informacije vezane uz zadatke. Komentari omogućuju otvorenu komunikaciju i suradnju između radnika i *admina*, što doprinosi boljoj organizaciji i izvršenju zadataka. Kreiranje zadataka putem forme omogućuje strukturiranje i jasno definiranje radnih zadataka unutar firme. Ova karakteristika omogućuje bolju organizaciju, raspodjelu radnih obveza te transparentnost u pogledu odgovornosti i prioriteta zadatka.

```
const createTask = async (e) => {
  e.preventDefault();

  try {
    setLoading(true);
    await addDoc(collection(db, "tasks"), {
      title: titleRef.current.value,
      description: descriptionRef.current.value,
      assignedUsers: assignedUsers,
      importance: importance,
      firm_id: parseInt(currentFirm),
      creator: currentUser.uid,
      archive: false,
      comments: [],
      isActive: false,
    });
    await createTimeline("create");
    toast.success("Task successfully created!");

    setTimeout(() => {
      navigate(0);
    }, 1000);
  } catch (e) {
    toast.error("Unknown error for action: Create task.");
    console.error(e);
  }

  setLoading(false);
};
```

Slika 2 - Prikazani kod za kreiranje zadataka

Uz karakteristiku kreiranja zadataka, aplikacija također sadrži neke HRM funkcije koje doprinose cjelokupnom upravljanju ljudskim resursima u tvrtki. Jedna od tih funkcija je pregled plaća radnika. Administrativni tim ima mogućnost uređivanja plaća radnika i podnošenja obračuna plaća za određeni mjesec. Nakon podnošenja obračuna, radnicima se na njihovim profilima prikazuje kartica s informacijama o isplaćenoj plaći za taj mjesec. Ova funkcija pruža transparentnost i olakšava administraciju plaća unutar tvrtke.

Informacije koje se koriste za obračun uključuju mjesečnu plaću, koja može biti fiksna ili temeljena na radnom satu, ovisno o postavkama koje je *admin* odabrao. Prilikom obračuna plaća, postoji mogućnost uzeti u obzir prekovremene sate koje su radnici ostvarili tijekom određenog razdoblja. To omogućuje pravedan obračun ukupne isplate radnicima s obzirom na njihovu osnovnu plaću i radne sate iznad redovnog radnog vremena. Dodatno, aplikacija pruža i polje za unos bonusa ukoliko je radnik zaslužio dodatnu nagradu za svoj rad ili postignuća. Ova funkcionalnost omogućuje administrativnom timu da pravedno i transparentno dodijeli bonuse radnicima na temelju njihovih zasluga ili ostvarenih ciljeva.

Nakon što je obračun plaća podnesen, radnicima se na njihovim profilima prikazuje kartica s detaljima isplaćene plaće za taj određeni mjesec. Ova kartica pruža radnicima transparentan uvid u informacije o njihovim primanjima, uključujući osnovnu plaću, prekovremene sate, bonuse i sve druge relevantne podatke o isplati.

Ova funkcionalnost olakšava radnicima praćenje i razumijevanje njihovih primanja te pruža transparentnost u pogledu obračuna plaća unutar tvrtke. Kroz ove informacije o plaćama, uključujući fiksnu plaću ili rad plaćen po satu, prekovremene sate i polje za bonuse, aplikacija omogućuje precizan i pravedan obračun plaća radnicima unutar tvrtke, pridonoseći transparentnosti i jednostavnosti administracije plaća.

```

const createPayroll = async (uid, user) => {
  try {
    setLoading(true);
    const overtimeHours =
      showOvertimeInfo && payrollUserOveritimeHoursRef.current
        ? payrollUserOveritimeHoursRef.current.value
        : 0;
    const overtimeSalary =
      showOvertimeInfo && payrollUserOvertimeSalaryRef.current
        ? payrollUserOvertimeSalaryRef.current.value
        : 0;

    const bonusSalary =
      showBonusInfo && payrollUserBonusRef.current
        ? payrollUserBonusRef.current.value
        : 0;

    await addDoc(collection(db, "payroll"), {
      firm_id: currentFirm,
      user_id: uid,
      date: payrollNextDateRef.current.value,
      salary: payrollUserSalaryRef.current.value,
      hours: payrollUserHoursRef.current.value,
      overtime_salary: overtimeSalary,
      overtime_hours: overtimeHours,
      total:
        payrollUserSalaryRef.current.value *
        payrollUserHoursRef.current.value +
        overtimeSalary * overtimeHours +
        parseInt(bonusSalary),
      status: "active",
      bonus: bonusSalary,
    });
    await setDoc(
      doc(db, "user", uid),
      {
        payrollActive: true,
      },
      { merge: true }
    );
    await createTimeline("createPayroll", user);
    navigate(0);
  } catch (error) {
    console.error(error);
  } finally {
    setLoading(false);
  }
};

```

Slika 3 - Prikazani kod za kreiranje obračuna

Aplikacija također prati zahtjeve za slobodnim vremenom koje podnose radnici. Radnici imaju mogućnost podnošenja zahtjeva za slobodnim danom ili odmorom putem aplikacije, a admini imaju ovlasti za prihvatanje ili odbijanje tih zahtjeva. Kada radnik podnese zahtjev za slobodnim vremenom, admini dobivaju obavijest o tom zahtjevu putem aplikacije ili putem obavijesti. Admini mogu pregledati detalje zahtjeva, uključujući datum, razlog zahtjeva i potrebno trajanje slobodnog vremena. Na temelju raspoloživosti resursa, radnog opterećenja ili drugih faktora, admini donose odluku o prihvatanju ili odbijanju zahtjeva. Ova funkcija omogućuje učinkovito praćenje i upravljanje radničkim slobodnim vremenom na jednostavan način. Radnici mogu jednostavno podnijeti zahtjev putem aplikacije umjesto da se oslanjaju na tradicionalne obrasce ili verbalnu komunikaciju. To poboljšava efikasnost i smanjuje mogućnost pogrešaka ili nesporazuma.

Za admina, ova funkcionalnost pruža pregled nad zahtjevima za slobodno vrijeme svih radnika na jednom mjestu. Ovo olakšava donošenje odluka o rasporedu, osigurava pravednu raspodjelu slobodnog vremena te smanjuje potencijalne konflikte između radnika. Također, admini mogu koristiti informacije o zahtjevima za slobodno vrijeme za planiranje resursa i upravljanje opterećenjem timova ili odjela.

Zaključno, praćenje zahtjeva za slobodnim vremenom omogućuje bolje upravljanje radničkim rasporedima, poboljšava transparentnost i olakšava administraciju slobodnog vremena unutar tvrtke.

```
const timeoffSendRequest = async () => {
  try {
    setLoading(true);
    const userRef = doc(db, "user", currentUser.uid);
    const userDoc = await getDoc(userRef);
    await addDoc(collection(db, "requests"), {
      type: typeRef.current.value,
      from: fromRef.current.value,
      until: untilRef.current.value,
      user: userDoc.data().firstname + " " + userDoc.data().lastname,
      status: "Requested",
      notes: notesRef.current.value,
      firm_id: currentFirm,
    });
    toast.success("Timeoff request sent successfully!");
  } catch (error) {
    toast.error("Unknown error.");
  }
  setLoading(false);
};
```

Slika 4 - Prikazani kod za kreiranje zahtjeva za slobodne dane

Aplikacija također omogućuje učitavanje (*eng. upload*) dokumenata koji su javno vidljivi svim radnicima u firmi. Ova funkcija olakšava dijeljenje važnih dokumenta, poput internih politika, procedura ili obavijesti, pružajući centraliziranu i pristupačnu platformu za pristup tim dokumentima. U suvremenom poslovnom okruženju, učinkovita komunikacija i razmjena informacija ključne su za uspjeh tvrtki. Stoga, implementacija aplikacije koja omogućuje učitavanje (*eng. upload*) dokumenata koji su javno vidljivi svim radnicima u firmi predstavlja značajan korak prema poboljšanju internih komunikacijskih procesa.

Jedna od glavnih prednosti prethodno navedenog, očituje se u olakšanom dijeljenju važnih dokumenata unutar tvrtke. Interni dokumenti kao što su politike, procedure ili obavijesti često zahtijevaju brz i jednostavan pristup svim radnicima. Ova funkcija također poboljšava efikasnost i produktivnost unutar organizacije. Umjesto da radnici troše vrijeme na traženje odgovarajućih dokumenata putem e-pošte ili lokalnih mrežnih mapa, mogu jednostavno pristupiti aplikaciji i preuzeti potrebne dokumente. To štedi vrijeme, smanjuje frustraciju i omogućuje radnicima da se brzo usredotoče na svoje zadatke.

```
const uploadFile = async (file) => {
  try {
    const storage = getStorage();
    const storageRef = ref(storage, "files/" + file.name);

    await uploadBytes(storageRef, file);

    const downloadURL = await getDownloadURL(storageRef);
    const fileData = {
      name: file.name,
      url: downloadURL,
      createdAt: new Date(),
      fileType: "document",
      firm_id: currentFirm,
    };

    await addDoc(collection(db, "files"), fileData);

    toast.success("File uploaded and stored successfully.");
    setTimeout(() => {
      navigate(0);
    }, 1000);
  } catch (error) {
    toast.error("Error uploading file: ", error);
  }
};
```

Slika 5 - Prikazani kod za učitavanje dokumenata

Nadalje, *admini* imaju mogućnost kreiranja pravila koja se moraju poštovati unutar firme. Ova funkcija omogućuje postavljanje i implementaciju pravila koja se odnose na radno vrijeme, ponašanje na radnom mjestu, etiku ili bilo koja druga pravila koja su važna za tvrtku. To pomaže u održavanju dosljednosti i usklađenosti s internim pravilima i regulativama. Uz mogućnost kreiranja i primjene pravila unutar aplikacije, administratori dobivaju moćan alat za održavanje reda, dosljednosti i usklađenosti s internim pravilima i regulativama unutar firme. Ova funkcija omogućuje tvrtki da jasno definira očekivanja i standarde ponašanja za sve radnike.

Pravila koja se mogu postaviti putem aplikacije obuhvaćaju različite aspekte radnog okruženja. Na primjer, mogu se utvrditi pravila vezana uz radno vrijeme, poput dolaska i odlaska s posla, dužine pauza i slično. To pomaže u uspostavljanju dosljednosti u radnom vremenu svih zaposlenika te u osiguravanju da se svi pridržavaju definiranih smjernica.

Etika je još jedan važan aspekt koji se može regulirati putem ove funkcije. Administratori mogu postaviti pravila koja se odnose na etičko ponašanje, integritet i poštivanje pravila o povjerljivosti podatak. Ova pravila su posebno važna u industrijama koje zahtijevaju visoku razinu povjerljivosti i sigurnosti informacija.

Dodatno, ova funkcija omogućuje tvrtki da implementira pravila koja su specifična za njezinu industriju ili regulativu kojoj mora zadovoljiti. Na primjer, tvrtke koje posluju u financijskom sektoru mogu postaviti pravila o poštivanju zakona o financijskom poslovanju ili pravila o sprečavanju pranja novca. Ovo osigurava da svi zaposlenici budu svjesni i pridržavaju se važećih propisa i regulativa.

Ove dodatne HRM funkcije unutar aplikacije omogućuju sveobuhvatan pristup upravljanju ljudskim resursima unutar tvrtke. Od praćenja plaća i odmora radnika do organizacije i dijeljenja dokumenata, aplikacija pruža alate koji olakšavaju administrativne procese i promiču uspješno upravljanje ljudskim resursima.

```

const createRule = async (e) => {
  e.preventDefault();

  try {
    await addDoc(collection(db, "rules"), {
      text: ruleTextRef.current.value,
      title: ruleTitleRef.current.value,
      firm_id: currentFirm,
    });
    toast.success("Rule successfully created!");

    setTimeout(() => {
      navigate(0);
    }, 1000);
  } catch (e) {
    toast.error("Unknown error for action: Create rule.");
    console.error(e);
  }
};

```

Slika 6 - Prikazani kod za kreiranje pravila

4. Upute

4.1. Registracija

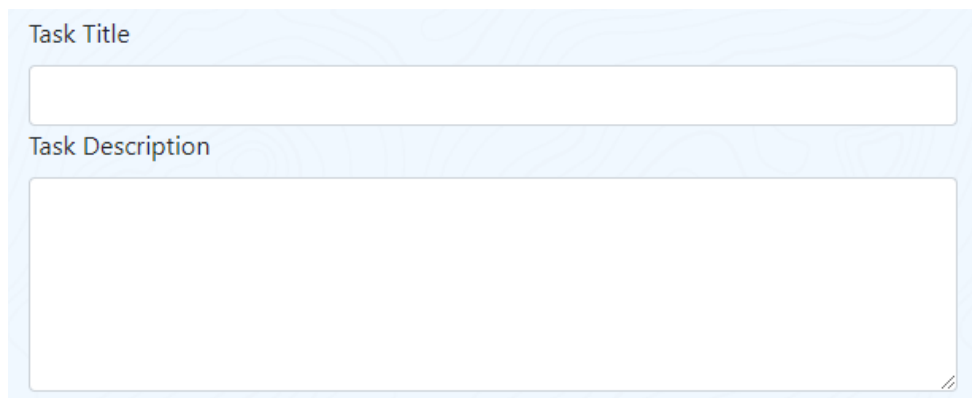
Za početak rada u aplikaciji potrebno je registrirati firmu. U ovom slučaju, aplikacija ne zahtijeva visoku razinu sigurnosti, već se fokusira na prikazivanje osnovnih HRM funkcija. Registracija firme se obavlja jednostavno. Na početnom zaslonu aplikacije nalazi se opcija "Registracija firme" u koju se unosi ime firme i e-mail adresu na koju osoba želi primiti registracijski kod. Nakon što je kod zatražen, potrebno je provjeriti e-mail i zabilježiti zaprimljeni kod. Nakon dobivenog goda, može se nastaviti s registracijom korisnika.

Postoje dvije vrste registracija korisnika: za admin korisnike i obične radnike. Nakon što korisnik dobije registracijski kod, vraća se u aplikaciju i pronalazi opciju "Registracija korisnika". Zatim odabire odgovarajuću opciju, ovisno o tome je li admin korisnik ili obični radnik. Unosi tražene osobne podatke, uključujući ime, prezime i e-mail adresu, te unosi primljeni registracijski kod. Kada sve ispravno unese, klikne na gumb "Register". Na taj način uspješno registrira svoj korisnički račun u aplikaciji i može pristupiti glavnim HRM funkcijama koje su mu dodijeljene u skladu s ulogom koju ima u organizaciji.

4.2. Početak korištenja funkcionalnosti

4.2.1. Kreiranje zadataka

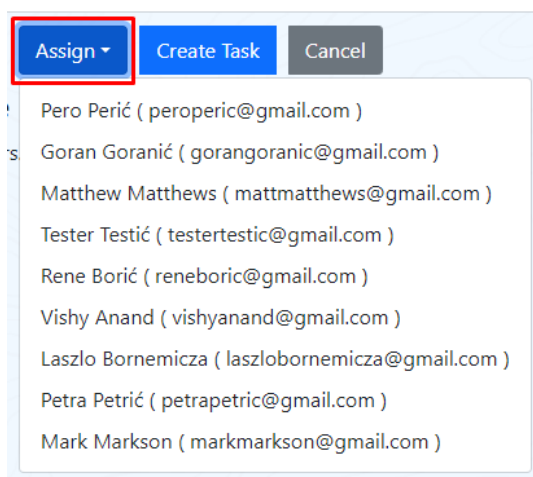
Kreiranje zadataka (*eng. task*) je jedna od glavnih funkcionalnosti ove HRM aplikacije. Samo korisnik s admin ulogom može kreirati zadatke. Pri kreiranju zadatka, korisnik će biti tražen da unese naziv zadatka i opis.



The image shows a form for creating a task. It has two main sections: 'Task Title' and 'Task Description'. Each section contains a large, empty text input field. The form is styled with a light blue background and rounded corners.

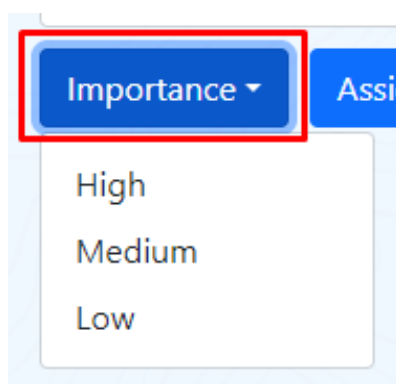
Slika 7 - Ispunjavanje forme za kreiranje zadatka

Admin ima mogućnost dodijeliti određene radnike za taj zadatak, ali to nije obavezno. Također, admin može odrediti važnost zadatka, koja može biti mala, srednja ili visoka.



The image shows a dropdown menu for assigning a task. The menu is open, showing a list of employee names and their email addresses. The 'Assign' button is highlighted with a red box. The list includes: Pero Perić (peroperic@gmail.com), Goran Goranić (gorangoranic@gmail.com), Matthew Matthews (mattmatthews@gmail.com), Tester Testić (testertestic@gmail.com), Rene Borić (reneboric@gmail.com), Vishy Anand (vishyanand@gmail.com), Laszlo Bornemicza (laszlobornemicza@gmail.com), Petra Petrić (petrapetric@gmail.com), and Mark Markson (markmarkson@gmail.com).

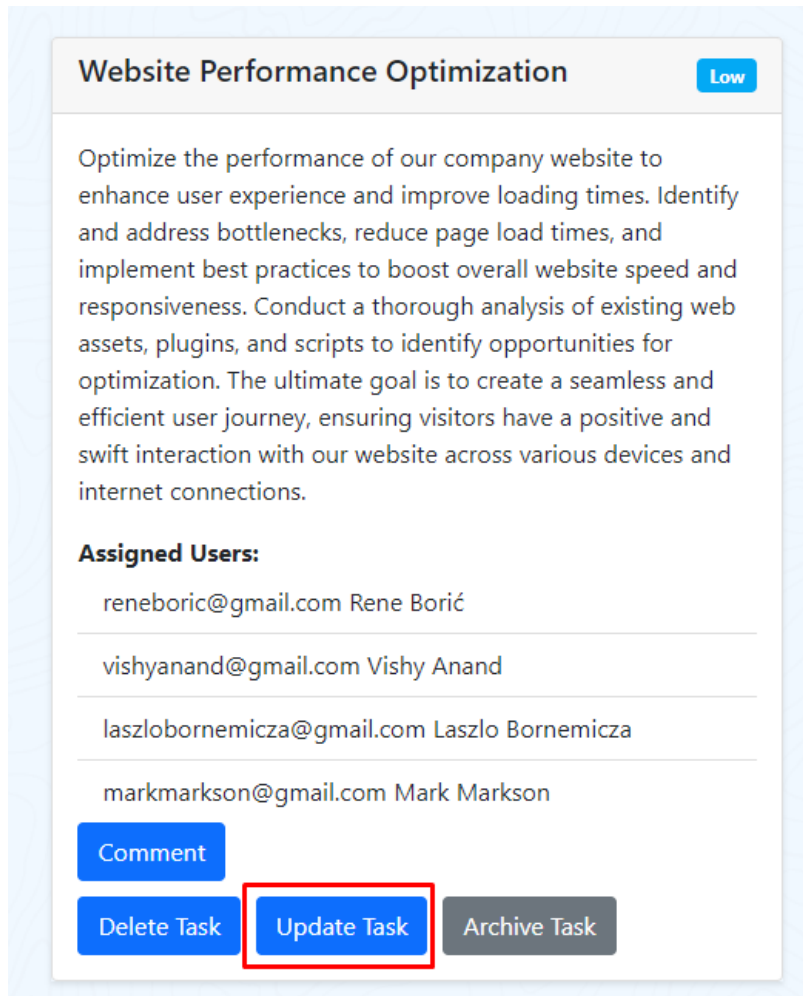
Slika 8 - Izbornik za dodjeljivanje radnika na zadatak



The image shows a dropdown menu for selecting the importance of a task. The menu is open, showing three options: High, Medium, and Low. The 'Importance' button is highlighted with a red box. The 'Assign' button is partially visible to the right.

Slika 9 - Izbornik za dodjeljivanje vrste važnosti na zadatka

Nakon što je zadatak kreiran, on će biti vidljiv svim korisnicima te firme. U budućnosti, ako je potrebno, admin može izmijeniti zadatke, ažurirajući naziv, opis ili druga polja.



Website Performance Optimization Low

Optimize the performance of our company website to enhance user experience and improve loading times. Identify and address bottlenecks, reduce page load times, and implement best practices to boost overall website speed and responsiveness. Conduct a thorough analysis of existing web assets, plugins, and scripts to identify opportunities for optimization. The ultimate goal is to create a seamless and efficient user journey, ensuring visitors have a positive and swift interaction with our website across various devices and internet connections.

Assigned Users:

- reneboric@gmail.com Rene Boric
- vishyanand@gmail.com Vishy Anand
- laszlobornemicza@gmail.com Laszlo Bornemicza
- markmarkson@gmail.com Mark Markson

[Comment](#)

[Delete Task](#) [Update Task](#) [Archive Task](#)

Slika 10 - Kreirani zadatak

Website Performance Optimization ✕

Task Title

Website Performance Optimization

Task Description

Optimize the performance of our company website to enhance user experience and improve loading times. Identify and address bottlenecks, reduce page load times, and implement best practices to boost overall website speed and responsiveness. Conduct a thorough analysis of existing web assets, plugins, and scripts to identify opportunities for optimization. The ultimate goal is to create a seamless and efficient user journey, ensuring visitors have a positive and swift interaction with our website across various devices and internet connections.

Importance ▾

Low

Close Save Changes

Slika 11 - Modul za ažuriranje zadatka

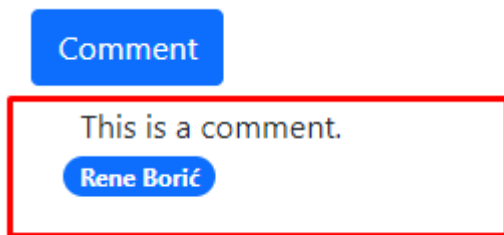
Admin također ima mogućnost brisanja zadatka ili arhiviranja. Kada je zadatak arhiviran, bit će premješten u poseban prozor gdje se nalaze svi arhivirani zadaci za tu firmu.

Aplikacija pruža funkcionalnost komentiranja za zadatke. Svi korisnici, ne samo admin, imaju mogućnost komentirati zadatke. Ovo omogućuje komunikaciju i suradnju između članova tima vezano uz određeni zadatak.

Comment

@ Comment..

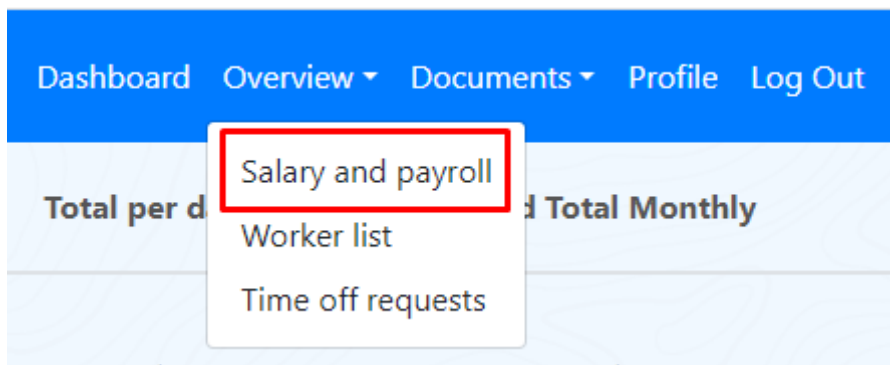
Slika 12 - Opcija komentiranja na zadatak



Slika 13 - Prikazani komentar

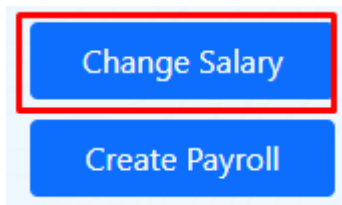
4.2.2. Vođenje evidencije plaće

Jedna od najvažnijih funkcionalnosti ove HRM aplikacije je postavljanje i vođenje evidencije plaća radnika u firmi. Pristup tom prozoru ima samo admin korisnik. Kada se otvori prozor za vođenje evidencije plaća, prikazuje se lista svih radnika u firmi s početno postavljenim zadanim iznosom plaće za svakog od njih.



Slika 14 - Navigacijska traka

Prva funkcionalnost vođenja evidencije je postavljanje plaće, koja može biti fiksna ili se može odrediti po satu, za svakog radnika. Ova funkcionalnost omogućuje *adminu* da ima pregled nad radnicima i okvirno zna kolika bi trebala biti njihova plaća.



Slika 15 - Gumb za mijenjanje okvirne plaće

Rene ×

First name

Last name

Hourly ▾

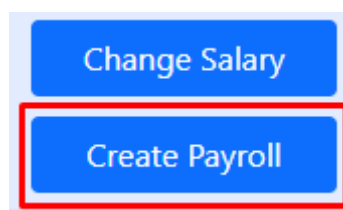
Salary /h

Hours per day

Close Save Changes

Slika 16 - Modul za mijenjanje okvirne plaće

Druga funkcionalnost je kreiranje obračuna (*eng. payroll*) za određenog korisnika. Prilikom kreiranja obračuna, *admin* odabire datum kada bi taj *obračun* trebao biti isplaćen. Zatim se unose radni sati koje je radnik odradio, plaća po satu, a po potrebi se unose i prekovremeni sati i plaća za prekovremeni rad. Na kraju, *admin* ima mogućnost dodavanja bonusa. Nakon što su svi podaci uneseni, kreira se obračun, a na listi se prikazuje datum za isti.



Slika 17- Gumb za kreiranje obračuna

Editing next payroll for Rene ×

Date of the next payroll

Hours

Salary

Add overtime

Overtime hours

Overtime salary

Add bonus

Bonus

Total: 2012

Slika 18 - Modul za kreiranje obračuna

Klikom na određeni datum na listi, *admin* ima mogućnost brisanja obračuna ako je slučajno napravio pogrešku pri unosu i želi započeti ispočetka ili ga može podnijeti (*eng. submit*). Važno je napomenuti da se obračun može podnijeti samo na dan koji odgovara odabranom datumu, inače će gumb za podnošenje biti onemogućen.

Slika 19 - Kreirani obračun za zadani datum

Ova funkcionalnost omogućuje *adminu* precizno vođenje evidencije plaća radnika u firmi te olakšava proces isplate plaća i praćenje finansijskih podataka vezanih uz plaće.

Payroll information for Rene Borić ×

Date:

27/07/2023

Hours:

134

Salary:

13

Overtime Hours:

7

Overtime Salary:

20

Bonus:

130

Total:

2012

Delete payroll

Close window

Submit Payroll

Slika 20 - Modul za podnošenje obračuna

Kada je obračun podnesen za određenog korisnika, taj korisnik može pristupiti detaljima tog obračuna putem svog profila. Na profilu korisnika, obračun će biti prikazan kao kartica s relevantnim informacijama. Korisnik će moći vidjeti detalje poput datuma obračuna, isplaćene plaće, radnih sati, plaće po satu, mogućih prekovremenih sati te bonusa, ako je dodijeljen. Na profilu će biti prikazani svi obračuni koje je taj radnik ikada primio. To omogućuje radniku da ima sve informacije vezane za svoje plaće na jednom mjestu.

Payroll

Payroll information for 27/07/2023
Total: 2012\$
Hours: 134
Salary: 13
Overtime hours: 7
Overtime salary: 20
Bonus: 130

Slika 21 - Kartica sa informacijama obračuna

Ova funkcionalnost omogućuje radniku jednostavan pristup njegovoj povijesti plaća. Radnik može pregledati i provjeriti sve detalje svojih prethodnih obračuna, pratiti promjene u plaćama tijekom vremena te imati cjelovit uvid u svoje primanja. Navedeno, olakšava praćenje i upravljanje financijskim aspektima njihovog radnog odnosa s firmom.

4.2.3. Evidencije slobodnih dana

Svaki korisnik u firmi ima mogućnost u aplikaciji na svom profilu poslati zahtjev za slobodne dane. Na profilu korisnika postoji posebna sekcija nazvana "Time off request", odnosno zahtjev za slobodne dane. U toj sekciji korisnik može odabrati vrstu slobodnih dana (poput godišnjeg odmora) te odrediti datum početka i završetka tog perioda slobodnih dana.

Korisnik također ima mogućnost upisivanja dodatnih bilješki ili napomena koje su povezane s tim zahtjevom za slobodne dane. Na primjer, moguće je unijeti razlog slobodnih dana ili bilo koje posebne zahtjeve ili informacije koje bi bile relevantne za taj period odmora.

Time Off Request

Timeoff
Type
Personal leave

From
07/21/2023

Until
07/29/2023

Notes
Family event attendance. Important occasion, need time to celebrate with close relatives. Will ensure pending tasks are completed before leave. Grateful for your approval. Thank you.

Submit request

Slika 22 - Forma za ispunjavanje zahtjeva za slobodne dane

Ova funkcionalnost omogućuje korisnicima jednostavan način slanja zahtjeva za slobodne dane putem aplikacije, umjesto da se izravno obrate upravi ili osoblju za upravljanje ljudskim resursima. Time se pojednostavljuje postupak zahtjeva za slobodnim danima i omogućuje brzu i učinkovitu komunikaciju između korisnika i uprave ili osoblja HR-a.

Nakon što je zahtjev za godišnji odmor poslan, *admin* ima pristup posebnom prozoru ili sučelju gdje može pregledati sve zahtjeve za godišnji odmor radnika u toj firmi. Prozor za pregled zahtjeva omogućuje *adminu* jednostavan i pregledan prikaz svih primljenih zahtjeva.

Pending Requests		Accepted requests	Terminated requests		
Name	Status	Reason	Notes	Date	Actions
Rene Borić	Requested	Personal leave	Family event attendance. Important occasion, need time to celebrate with close relatives. Will ensure pending tasks are completed before leave. Grateful for your approval. Thank you.	2023-07-21 - 2023-07-29	Y X

Slika 23 - Lista trenutnih zahtjeva za slobodne dane

U ovom posebnom prozoru, *admin* može vidjeti sve detalje zahtjeva za godišnji odmor, uključujući korisnika, vrstu slobodnih dana i datume zahtjeva.

Adminu je omogućeno jednostavno prihvaćanje ili odbijanje zahtjeva korisnika za godišnji odmor putem sučelja. Na temelju raspoloživosti resursa, poslovnih potreba ili drugih faktora, *admin* može donijeti odluku o prihvatu ili odbijanju zahtjeva.

Prihvaćeni zahtjevi će biti prikazani u posebnom prozoru gdje se nalaze svi prihvaćeni zahtjevi radnika za godišnji odmor. Na isti način, odbijeni zahtjevi će biti prikazani na zasebnom prozoru. Ova strukturirana organizacija omogućuje *adminu* jednostavan pristup svim prihvaćenim ili odbijenim zahtjevima radnika.

Pending Requests		Accepted requests	Terminated requests	
Name	Status	Reason	Notes	Date
Rene Borić	Accepted	Personal leave	Family event attendance. Important occasion, need time to celebrate with close relatives. Will ensure pending tasks are completed before leave. Grateful for your approval. Thank you.	2023-07-21 - 2023-07-29

Slika 24 - Lista prihvaćenih zahtjeva za slobodne dane

Pending Requests		Accepted requests	Terminated requests	
Name	Status	Reason	Notes	Date
Laszlo Bornemicza	Declined	Annual leave	I am requesting annual leave to take some much-needed time off and recharge. The past year has been filled with dedicated efforts and hard work, and I believe that a short break will help me return with increased productivity and creativity. During my absence, I will ensure that all pending tasks are completed or delegated to colleagues. Your approval for this annual leave would be invaluable to help me maintain a healthy work-life balance. Thank you for considering my request.	2023-07-14 - 2023-07-25

Slika 25 - Lista odbijenih zahtjeva za slobodne dane

Ova funkcionalnost olakšava proces upravljanja zahtjevima za godišnji odmor, omogućujući *adminu* učinkovito donošenje odluka i praćenje statusa zahtjeva svih radnika u firmi.

4.2.4 Uređivanje informacija na profilu

Svaki korisnik ima mogućnost uređivanja informacija koje se odnose na njihov profil kako bi osigurali ažurirane i točne podatke. U okviru svojeg profila, korisnik može pristupiti opciji uređivanja i promijeniti različite informacije o sebi.

Među informacijama koje korisnik može mijenjati su:

1. Ime i prezime: Korisnik može izmijeniti svoje ime i prezime ako je došlo do promjene ili je bilo netočno uneseno.
2. Spol: Ako korisnik želi ispraviti spol u opcijama
3. Adresa: Korisnik može unijeti ili izmijeniti svoju adresu prebivališta
4. Nacionalnost: Ako korisnik želi promijeniti informaciju o svojoj nacionalnosti, ima mogućnost ažuriranja te informacije.
5. Edukacija: Korisnik može unijeti podatke o svojoj edukaciji, kao što su završeni stupanj ili diplome.
6. Datum zapošljenja: Korisnik može upisati datum zapošljenja
7. Broj telefona: Korisnik može unijeti ili promijeniti broj telefona kako bi osigurao ispravne kontakt informacije.
8. Preferirani jezik: Korisnik može odabrati svoj preferirani jezik za komunikaciju.
9. Preferirani način komunikacije: Korisnik može odabrati svoj preferirani način komunikacije, kao što su e-mail, telefonski poziv, SMS ili slično.

Ove opcije uređivanja omogućuju korisniku da održava svoje osobne podatke ažurnima i prilagođenima te omogućuju bolju komunikaciju i upravljanje informacijama u okviru aplikacije.

Profile

Email address

reneboric@gmail.com

First name

Rene

Last name

Borić

Gender

Male Female

Address

Ul. Jurja Žakna 23

Nationality

Croatian

Education

Faculty of informatics Pula

Employment start date

06/12/2020

Phone number

385912773721

Preferred language

English

Communication Preference (E-mail, phone, sms or text messages, in-person meetings, etc.)

E-mail

Save

Slika 26 - Forma za mijenjanje informacija o korisniku na profilu

Sve informacije vezane za profile korisnika, *admin* ima na pregled u posebnom prozoru pod nazivom "Worker list". U tom prozoru, *admin* ima pregled svih korisnika u firmi i njihovih profilnih informacija.

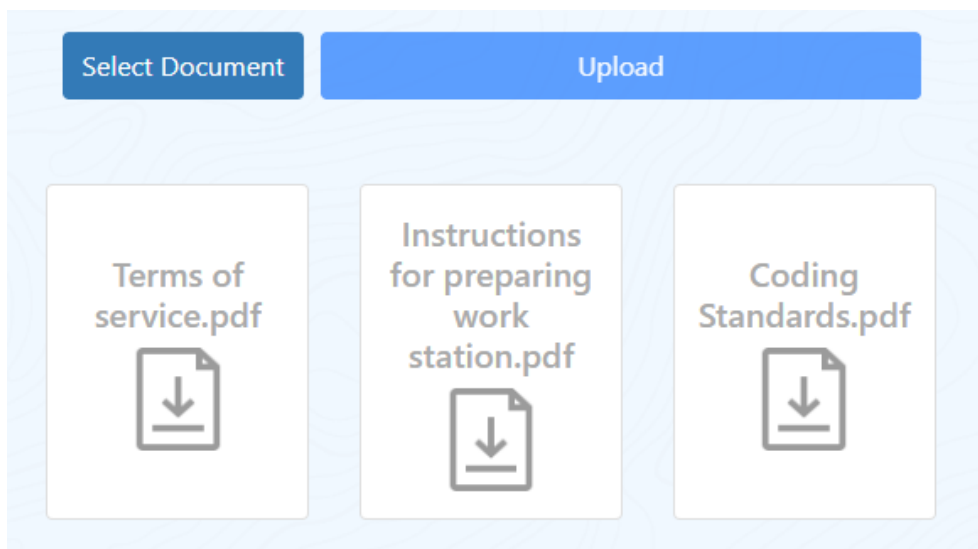
Ova pregledna tablica omogućuje *adminu* brz i jednostavan pristup svim korisnicima u organizaciji te njihovim profilnim podacima. Ovaj "Worker list" prozor olakšava administrativne zadatke povezane s upravljanjem ljudskim resursima u organizaciji.

4.2.5. Evidencija važnih dokumenata

Admini imaju mogućnost učitavanja (*eng. upload*) važnih dokumenata za firmu putem aplikacije. Ovi dokumenti mogu uključivati interna pravila, obrasce, politike, upute ili druge dokumente koji su relevantni za radnu okolinu i postupke u firmi. Ova funkcionalnost omogućuje *adminima* jednostavno i sigurno dijeljenje dokumenata s ostalim korisnicima.

Osim pregleda, korisnici će imati mogućnost preuzimanja tih dokumenata na svoje uređaje. Na taj način, korisnici mogu imati pristup i koristiti dokumente izvan aplikacije, na primjer, isprintati ih ili ih koristiti kao referencu prilikom obavljanja određenih zadataka ili postupaka.

Ova funkcionalnost olakšava dijeljenje važnih informacija unutar organizacije, osiguravajući da svi korisnici imaju pristup ažuriranim i relevantnim dokumentima koji su potrebni za njihov rad.



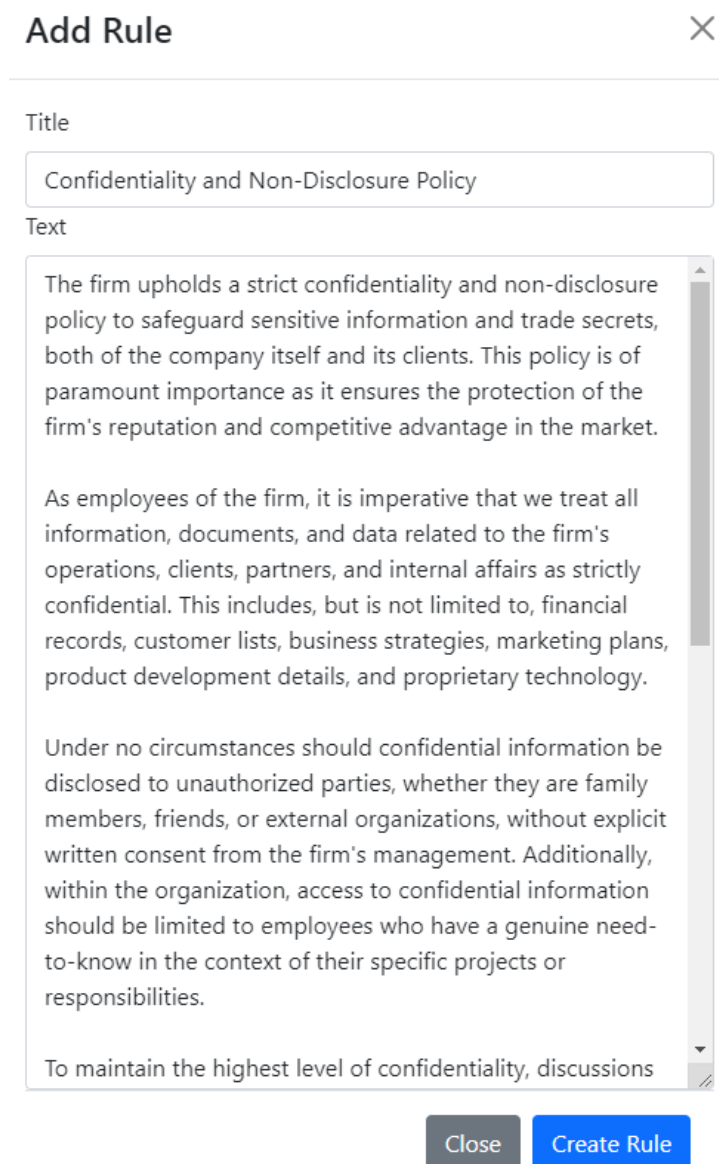
Slika 27 - Učitavanje te pregled učitanih dokumenata

4.2.6. Kreiranje pravila

Ova funkcionalnost omogućuje *adminima* da definiraju i uspostave pravila koja se moraju poštivati na radnom mjestu ili unutar organizacije. Pravila mogu obuhvaćati različite aspekte kao što su kodeks ponašanja, sigurnosne smjernice, politike privatnosti ili bilo koja druga pravila koja su relevantna za radno okruženje.

Kroz sučelje za kreiranje pravila, *admini* mogu unijeti tekst pravila i opisati njihovu svrhu, primjenu i očekivane standarde ili postupke. Također, mogu dodati dodatne informacije ili primjere kako bi pravila bila jasnija i razumljivija za sve korisnike.

Nakon što su pravila kreirana, ona će biti dostupna svim korisnicima putem aplikacije. Korisnici će moći pročitati pravila, razumjeti njihov sadržaj i pridržavati se navedenih smjernica. Pravila će biti dostupna kao referenca i podsjetnik za pridržavanje standarda i očekivanja na radnom mjestu.



Add Rule ✕

Title

Confidentiality and Non-Disclosure Policy

Text

The firm upholds a strict confidentiality and non-disclosure policy to safeguard sensitive information and trade secrets, both of the company itself and its clients. This policy is of paramount importance as it ensures the protection of the firm's reputation and competitive advantage in the market.

As employees of the firm, it is imperative that we treat all information, documents, and data related to the firm's operations, clients, partners, and internal affairs as strictly confidential. This includes, but is not limited to, financial records, customer lists, business strategies, marketing plans, product development details, and proprietary technology.

Under no circumstances should confidential information be disclosed to unauthorized parties, whether they are family members, friends, or external organizations, without explicit written consent from the firm's management. Additionally, within the organization, access to confidential information should be limited to employees who have a genuine need-to-know in the context of their specific projects or responsibilities.

To maintain the highest level of confidentiality, discussions

Close Create Rule

Slika 28 - Modul za kreiranje pravila

Confidentiality and Non-Disclosure Policy

The firm upholds a strict confidentiality and non-disclosure policy to safeguard sensitive information and trade secrets, both of the company itself and its clients. This policy is of paramount importance as it ensures the protection of the firm's reputation and competitive advantage in the market.

As employees of the firm, it is imperative that we treat all information, documents, and data related to the firm's operations, clients, partners, and internal affairs as strictly confidential. This includes, but is not limited to, financial records, customer lists, business strategies, marketing plans, product development details, and proprietary technology.

Under no circumstances should confidential information be disclosed to unauthorized parties, whether they are family members, friends, or external organizations, without explicit written consent from the firm's management. Additionally, within the organization, access to confidential information should be limited to employees who have a genuine need-to-know in the context of their specific projects or responsibilities.

To maintain the highest level of confidentiality, discussions or sharing of confidential information should never take place in public areas such as elevators, cafeterias, public transportation, or social gatherings, where unauthorized individuals may overhear or see sensitive material.

Furthermore, employees are responsible for taking appropriate measures to secure both physical and digital documents that contain confidential information. Utilizing password protection and secure storage systems when necessary is a critical step in ensuring the protection of sensitive data.

Upon termination of employment, employees must continue to adhere to this policy and refrain from disclosing or utilizing any confidential information acquired during their tenure.

By complying with this Confidentiality and Non-Disclosure Policy, we collectively contribute to safeguarding the firm's interests and upholding the trust placed in us by our clients and stakeholders. Any violation of this policy will result in disciplinary action, up to and including termination of employment. Let us all be committed to maintaining the highest standards of confidentiality and professionalism at all times.

Delete rule

Slika 29 - Kreirano pravilo

Ova funkcionalnost pomaže u osiguravanju dosljednosti i usklađenosti s pravilima unutar organizacije te olakšava komunikaciju i implementaciju važnih smjernica. Također, omogućuje *adminima* da lako ažuriraju ili mijenjaju pravila kako se potrebe ili zahtjevi organizacije mijenjaju.

5. Zaključak

U današnjem poslovnom okruženju, upravljanje ljudskim resursima je ključni faktor koji određuje uspjeh i konkurentsku prednost organizacija. Uspješno upravljanje HR procesima postaje sve važnije, a softveri za upravljanje ljudskim resursima (HR softveri) postaju neophodan alat za postizanje učinkovitosti i produktivnosti.

Prednosti HR softvera su mnogobrojne. Omogućuju smanjenje administrativnih zadataka, poboljšanje komunikacije unutar organizacije, povećanje transparentnosti HR procesa te učinkovitije praćenje radne snage. Osim toga, HR softveri doprinose boljem iskorištavanju resursa, smanjenju troškova i povećanju zadovoljstva zaposlenika.

Međutim, važno je napomenuti da implementacija HR softvera može donijeti izazove. Potrebno je osigurati adekvatnu obuku zaposlenika za korištenje sustava i osigurati sigurnost podataka. Također, odabir odgovarajućeg HR softvera koji odgovara potrebama i veličini organizacije ključan je za uspješnu primjenu.

Uzimajući u obzir sve navedeno, HR softveri su neprocjenjiv alat za moderno upravljanje ljudskim resursima. Njihova integracija u organizacijske procese omogućava postizanje konkurentске prednosti, povećanje produktivnosti i ostvarenje ciljeva organizacije. Stoga, organizacije koje prepoznaju važnost i potencijal HR softvera bit će sposobne da brže i efikasnije odgovaraju na promjenjive poslovne zahtjeve i ostvare održiv rast u dinamičnom poslovnom okruženju.

6. Literatura

1. ReactJS, dostupno na: <https://legacy.reactjs.org/>
2. Firebase, dostupno na: <https://firebase.google.com/>
3. React router, dostupno na: <https://reactrouter.com/en/main>
4. React bootstrap, dostupno na: <https://react-bootstrap.netlify.app/docs/getting-started/why-react-bootstrap>
5. Comparing Firebase vs MongoDB: <https://www.mongodb.com/firebase-vs-mongodb>
6. GitHub repozitorij: <https://github.com/rboric/react-hrm-app>

7. Popis slika

Slika 1 - Prikazani kod za registriranje firme	9
Slika 2 - Prikazani kod za kreiranje zadataka	10
Slika 3 - Prikazani kod za kreiranje obračuna	12
Slika 4 - Prikazani kod za kreiranje zahtjeva za slobodne dane	13
Slika 5 - Prikazani kod za učitavanje dokumenata	14
Slika 6 - Prikazani kod za kreiranje pravila	16
Slika 7 - Ispunjavanje forme za kreiranje zadataka	18
Slika 8 - Izbornik za dodjeljivanje radnika na zadatak	18
Slika 9 - Izbornik za dodjeljivanje vrste važnosti na zadatak	18
Slika 10 - Kreirani zadatak	19
Slika 11 - Modul za ažuriranje zadatka	20
Slika 12 - Opcija komentiranja na zadatak	20
Slika 13 - Prikazani komentar	21
Slika 14 - Navigacijska traka	21
Slika 15 - Gumb za mijenjanje okvirne plaće	21
Slika 16 - Modul za mijenjanje okvirne plaće	22
Slika 17- Gumb za kreiranje obračuna	22
Slika 18 - Modul za kreiranje obračuna	23
Slika 19 - Kreirani obračun za zadani datum	23
Slika 20 - Modul za podnošenje obračuna	24
Slika 21 - Kartica sa informacijama obračuna	25
Slika 22 - Forma za ispunjavanje zahtjeva za slobodne dane	26
Slika 23 - Lista trenutnih zahtjeva za slobodne dane	27
Slika 24 - Lista prihvaćenih zahtjeva za slobodne dane	27
Slika 25 - Lista odbijenih zahtjeva za slobodne dane	27
Slika 26 - Forma za mijenjanje informacija o korisniku na profilu	29
Slika 27 - Učitavanje te pregled učitanih dokumenata	30
Slika 28 - Modul za kreiranje pravila	31
Slika 29 - Kreirano pravilo	32

Sažetak

Cilj ovog završnog rada je izrada web aplikacije za Upravljanje Ljudskim Resursima (HRM) korištenjem tehnologija kao što su ReactJS i Firebase, s naglaskom na olakšavanje procesa upravljanja osobljem u organizacijama. HRM je ključna komponenta uspješnog poslovanja, zahtijeva učinkovito vođenje zaposlenika kroz njihov cijeli radni vijek. U ovom radu fokus je na razvoju alata koji pruža ključne funkcionalnosti za jednostavnije praćenje, organizaciju i upravljanje ljudskim resursima. Kroz implementaciju ove web aplikacije, korisnicima će biti omogućeno praćenje osnovnih radnih informacija, kao što su prisutnost i radno vrijeme, uz dodatne funkcije poput kreiranja i praćenja zahtjeva za odmorom i dopustom te kreiranje zadataka. Kombinacija ReactJS-a za korisničko sučelje i Firebase platforme za backend osigurava brzu, skalabilnu i pouzdanu platformu za upravljanje ljudskim resursima.

U zaključku, ova web aplikacija pružit će svestran alat za olakšavanje HR procesa unutar organizacija, omogućujući bolje upravljanje osobljem i poboljšanu komunikaciju unutar organizacije.

Ključne riječi: ReactJS, Firebase, web aplikacija, upravljanje ljudskim resursima

Abstract

The aim of this final project is to develop a web application for Human Resources Management (HRM) using technologies such as ReactJS and Firebase, with a focus on streamlining personnel management processes within organizations. HRM is a pivotal component of successful business operations, requiring effective management of employees throughout their entire careers. This work concentrates on creating a tool that provides essential functionalities for simpler tracking, organization, and management of human resources. Through the implementation of this web application, users will be able to monitor basic work-related information, such as attendance and working hours, along with additional features like creating and tracking leave and vacation requests, as well as task management. The combination of ReactJS for the user interface and the Firebase platform for the backend ensures a fast, scalable, and reliable platform for human resource management.

In conclusion, this web application will provide a versatile tool for streamlining HR processes within organizations, enabling improved personnel management and enhanced communication within the organization.

Keywords: ReactJS, Firebase, web application, human resources management