

A decorative graphic consisting of blue circuit-like lines with small circles at the ends, extending horizontally from the left and right sides of the central black box.

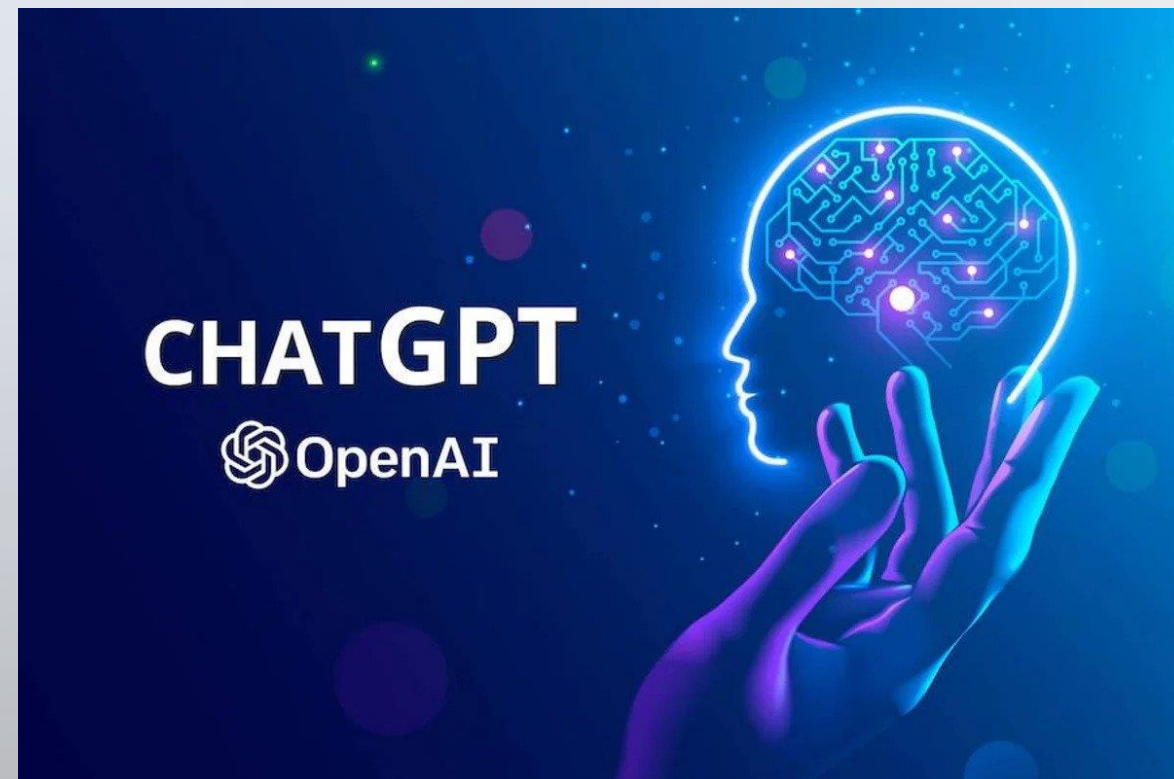
# PROJEKT MENADŽER

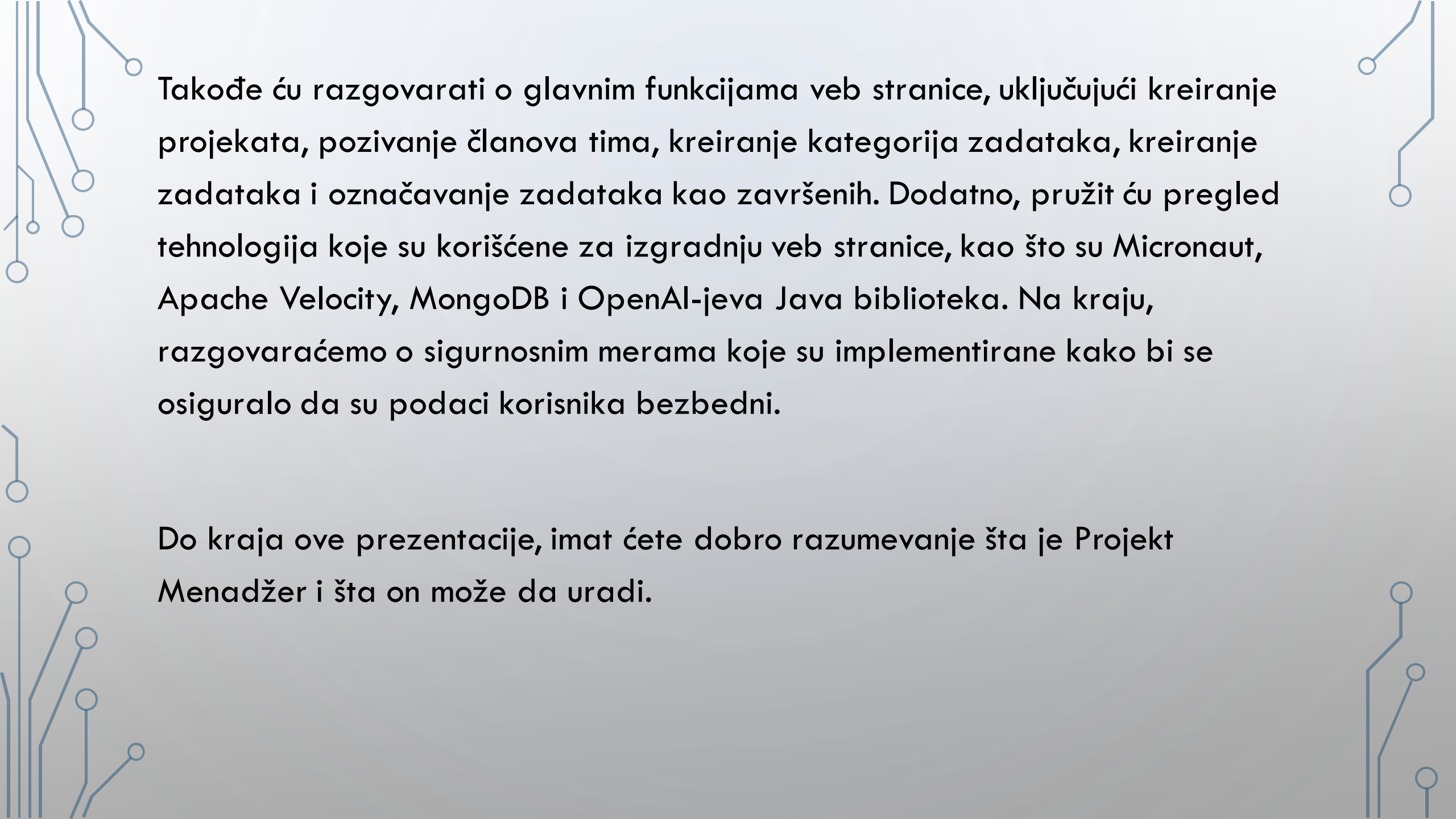
KORISTEĆI AI

# UVOD U PROJEKT MENADŽER

U ovoj prezentaciji upoznat ću vas sa Projekt Menadžerom, veb stranica osmišljena da pomogne menadžerima i njihovim timovima da efikasnije sarađuju. Projekt Menadžer je platforma bazirana na web-u koja se može koristiti za upravljanje širokim spektrom projekata, od IT do umetničkih pa čak i do uređenja vrtova. Bez obzira da li radite sami ili sa timom, Projekt Menadžer vam može pomoći da ostanete organizovani i na pravom putu.

Jedna od značajnih karakteristika Projekt Menadžera je integracija sa ChatGPT-om, AI asistentom koji vam može pomoći da dodelite zadatke pravim članovima tima. Sa AI asistentom mozete izbeći dodeljivanje zadataka članovima čije zanimanje ne spada u tu oblast, također se možete uveriti da nijedna grupa ljudi nije opterećena sa previše zadataka. Kasnije u prezentaciji detaljnije ćemo se pozabaviti funkcionalnostima ChatGPT-a.





Takođe ću razgovarati o glavnim funkcijama veb stranice, uključujući kreiranje projekata, pozivanje članova tima, kreiranje kategorija zadataka, kreiranje zadataka i označavanje zadataka kao završenih. Dodatno, pružit ću pregled tehnologija koje su korišćene za izgradnju veb stranice, kao što su Micronaut, Apache Velocity, MongoDB i OpenAI-jeva Java biblioteka. Na kraju, razgovaraćemo o sigurnosnim merama koje su implementirane kako bi se osiguralo da su podaci korisnika bezbedni.

Do kraja ove prezentacije, imat ćete dobro razumevanje šta je Projekt Menadžer i šta on može da uradi.

# REGISTRACIJA I PRIJAVA

Sajt projekt menadžer ima sistem za autentifikaciju koji omogućava korisnicima da kreiraju i prijave se na svoj nalog. To je neophodan sistem kako bi se projekti i privatne informacije razdvajali od drugih korisnika i kako bi se dodale funkcionalnosti poput prilagođavanja slike profila i sistema pozivnica.


# KREACIJA PROFILA


Nakon što korisnik otvori sajt, biće upućen na ekran za prijavu. S te strane korisnici mogu da se prijave ili kreiraju svoj nalog. Prilikom kreiranja naloga, korisnik mora da pruži:


- Puno Ime
- Profesija
- Email
- Snažna lozinka koja ima najmanje 8 karaktera, uključuje jedno veliko slovo, jedno malo slovo, jedan broj i jedan specijalni znak.


Pre registracije naloga, takođe proveravamo ispravnost imena korisnika, e-pošte, naziva posla i lozinke kako bi se obezbedila pravilna dužina i format.


## Registracija

 Ime

 Prezime

 Profesija

 Email

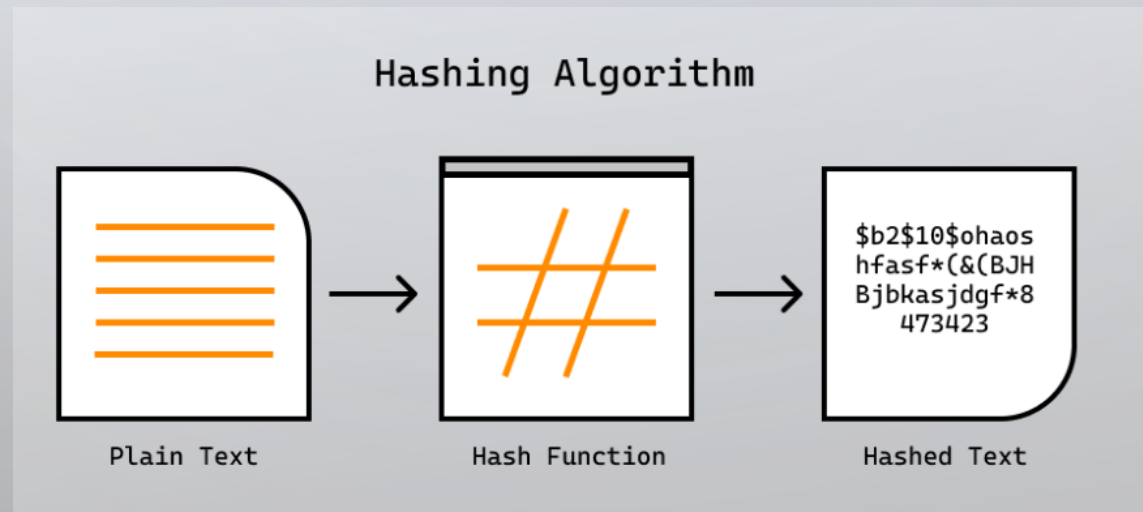
 Šifra

Register

Već imate profil? Prijavite se [ovde](#).

# MD5

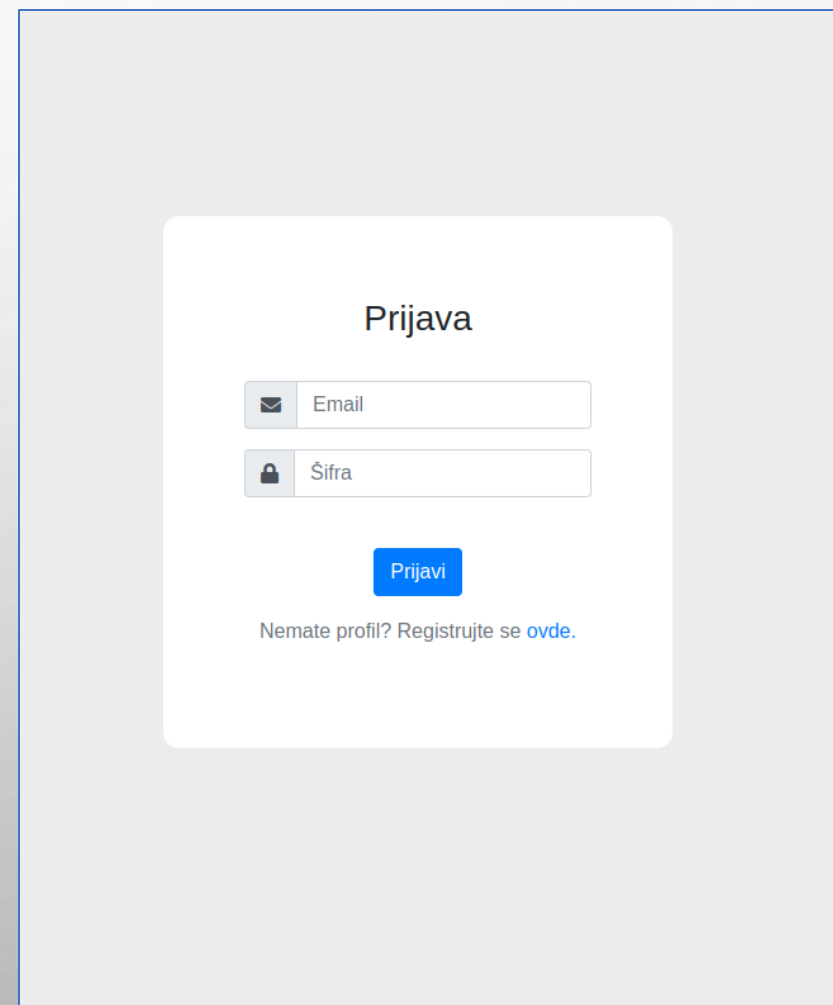
Nakon registracije naloga, pre nego što se lozinka skladišti u bazi podataka, kriptuje se korišćenjem MD5 algoritma tako da u slučaju napada na sigurnost šteta bude minimalna. Iako je lozinka kriptovana i ne može se vraćati u originalni oblik, koristimo isti algoritam za poređenje kriptovane lozinke i one koja je uneta prilikom prijave.



# PRIJAVA

Sistem autentifikacije koristi Micronautov JWT (JSON Web Token) sistem za bezbednost. JWT omogućava korisnicima da se sigurno prijave, a jednom kada se prijave, JWT token se čuva u njihovim kolačićima i automatski se koristi za svaku zahtev koji zahteva autentifikaciju. Ovaj token koristimo za identifikaciju korisnika i praktično sve u bazi podataka se može pratiti po korisniku.

Autentifikacioni sistem može biti unapređen dodavanjem hlađenja na autentifikacione krajnje tačke radi zaštite od brutalne sile, dodavanjem provere e-pošte, 2FA (dvofaktornog autentifikacionog sistema) i brisanjem naloga, ali nisam uspeo da implementiram sve te funkcije zbog ograničenog vremena.

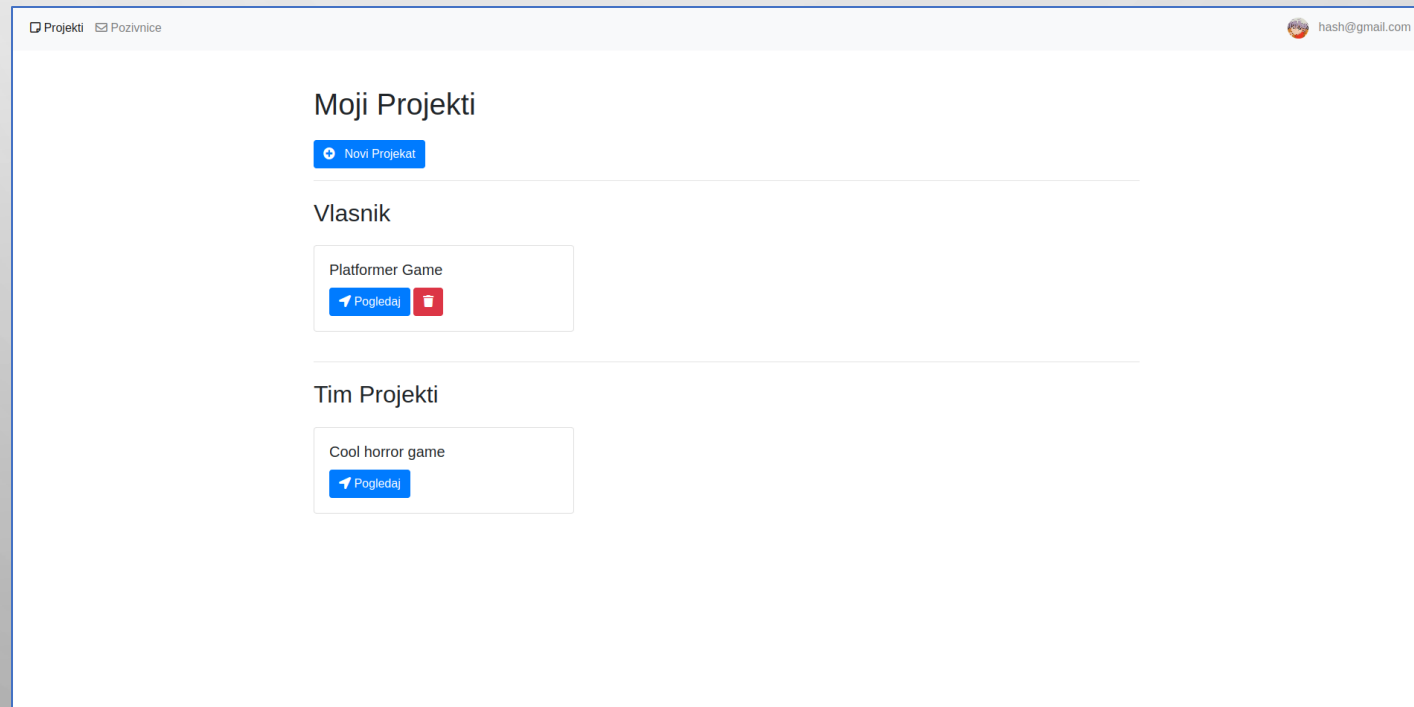


The image shows a login form titled "Prijava" (Login) centered on a light gray background. The form is a white rounded rectangle containing two input fields: "Email" with an envelope icon and "Šifra" (Password) with a lock icon. Below these fields is a blue "Prijavi" (Login) button. At the bottom of the form, there is a link that says "Nemate profil? Registrujte se [ovde](#)."



# KREACIJA PROJEKTA

Kada se korisnik autentikuje, doći će na početnu stranicu. Tu mogu videti sve projekte koje su kreirali i sve projekte kojima pripadaju. Početna stranica će biti prazna, ali korisnik može kliknuti na dugme "Novi projekat" i započeti proces kreiranja projekta. Da bi kreirali projekat, korisniku je potrebno samo ime projekta i malo volje! Korisnik takođe može brisati projekte koje je kreirao, ali ne i projekte kojima je samo član.

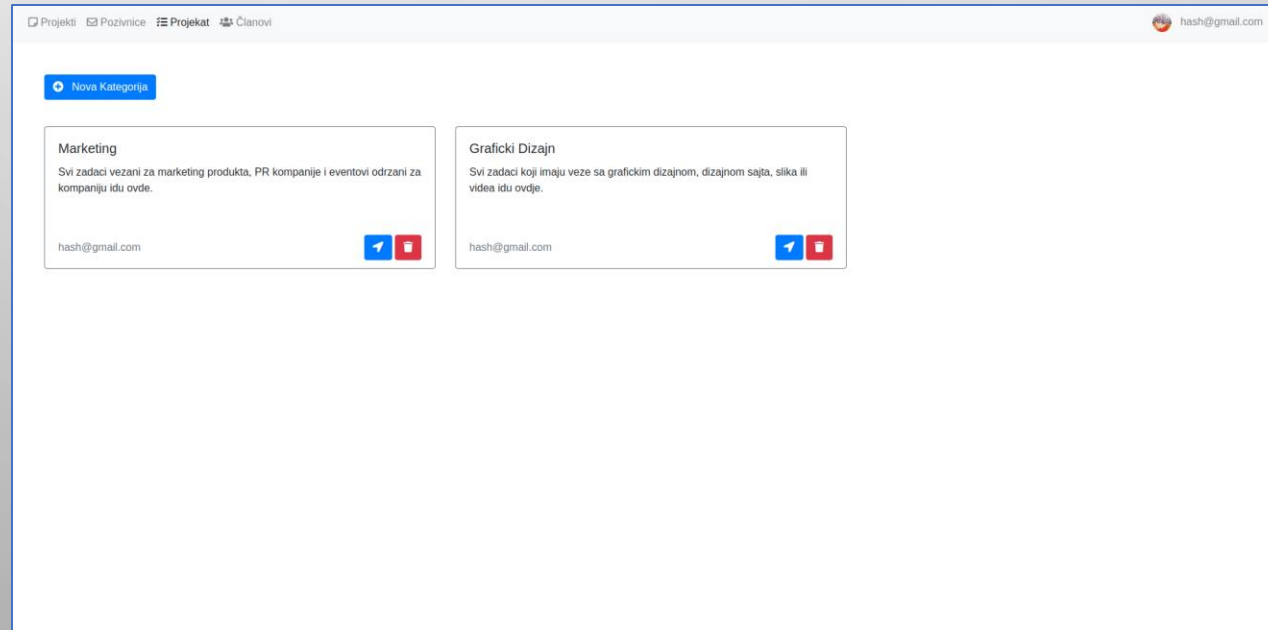


Korisnici mogu pozivati druge na svoj projekat klikom na dugme "Članovi" u navigacionoj traci i klikom na plus dugme. Tada samo treba da unesu imejl osobe koju žele pozvati. Nakon slanja poziva, novi poziv će se pojaviti u sekciji "Pozivnice" kod pozvanog korisnika, a korisnik koji je poslao poziv takođe može pratiti poslate pozive u svojoj sekciji "Pozivnice".

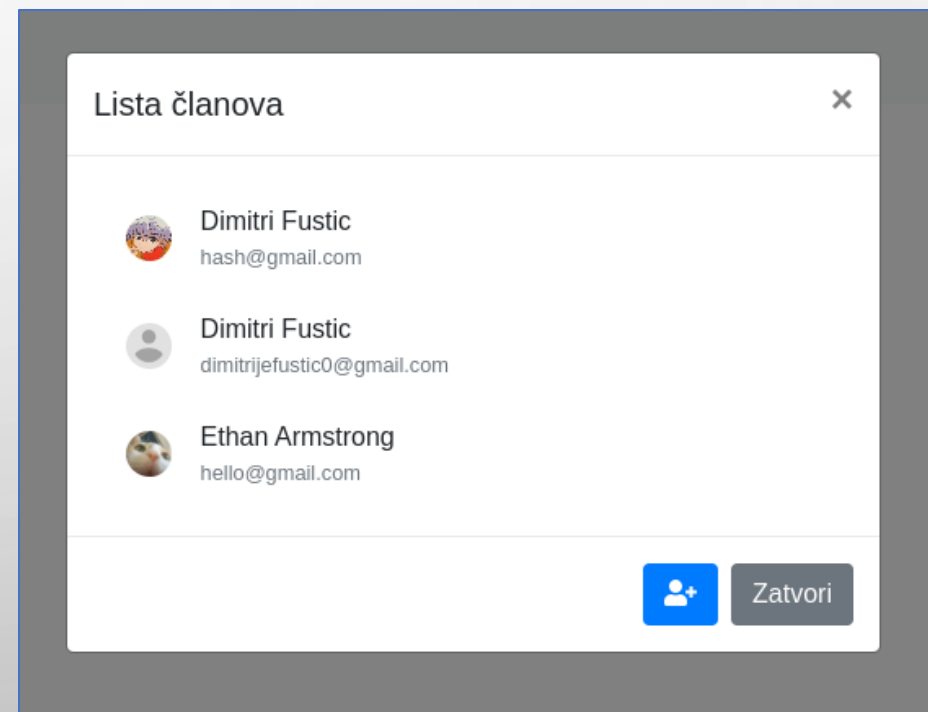
Primljene Pozivnice		
Za Prezentaciju	dimitrijefustic0@gmail.com	<input checked="" type="checkbox"/>
Poslate Pozivnice		
Vijesti Web Sajt	hash2@gmail.com	<input type="checkbox"/>
projekat jedan	hash2@gmail.com	<input type="checkbox"/>

# KATEGORIJE

Kada korisnik kreira svoj projekat i pozove članove, može početi kreirati zadatke. Veliki projekti mogu imati mnogo zadataka koji zahtevaju različite veštine i spadaju u različite kategorije. Kako ne bi došlo do mešanja, na primer, grafičkog dizajna i programiranja za vašu novu hit indie igru, možete kreirati posebne kategorije. Kategorije su grupe zadataka dizajnirane da ih odvoje. Možete na primer kreirati kategoriju "grafički dizajn" i tamo postaviti sve zadatke vezane za dizajn, a drugu kategoriju pod nazivom "oglašavanje" i tamo postaviti sve zadatke vezane za oglašavanje. Takođe možete brisati ove kategorije po potrebi.




Nakon što pozvani korisnik prihvati poziv, biće dodan u projekte i biće vidljiv u sekciji "Članovi". Mogao sam ovaj sistem poboljšati mnogo više dodavanjem dugmeta za odbijanje poziva, uklanjanjem korisnika iz projekta i dodavanjem specijalnih uloga projektima, ali nisam imao dovoljno vremena.



# ZADACI

Nakon što korisnik kreira željenu kategoriju, može početi sa pisanjem zadatka. Prilikom kreiranja zadatka, potrebno je navesti ime zadatka, opis i listu korisnika kojima želite da dodelite taj zadatak. Kada se zadatak kreira, on će se prikazati na tabli sa oznakom "u toku". Nakon što se zadatak završi, može se označiti kao završen i po potrebi obrisati.

 Novi Zadatak

## Lista Zadataka

### Koncert Poster

Dodjeljen:

hash2@gmail.com

Trebaju se napraviti poster za buduci koncert odrzan na kampusu.

[Završi](#) [Obriši](#)

[Izrada](#)

### Web Dizajn

Dodjeljen:

hash2@gmail.com

hash@gmail.com

Napraviti dizajn web sajta koristeći photoshop. Također se trebaju dizajnirati ikone.

[Završi](#) [Obriši](#)

[Izrada](#)

### Logo

Dodjeljen:

hash@gmail.com

Napraviti jednostavan logo za sutdentski form veb sajt.

[Završi](#) [Obriši](#)

[Izrada](#)

# AI ZADACI?

AI integracija. Želio sam da ovaj projekat bude poseban, a budući da je AI danas veoma popularan, odlučio sam se da dodam malo AI-a. Nisam imao dovoljno vremena da treniram sopstveni model za neki manji cilj, pa sam se odlučio za korišćenje ChatGPT turbo 3.5.

The screenshot shows a web application interface with a modal window titled "Kreacija Zadataka" (Task Creation). The modal has a close button (X) in the top right corner. It contains a text input field with the value "PHP CRUD Web Aplikacija". Below this is a text area containing the text "Napraviti jednostavnu CRUD web aplikaciju koristeći PHP", which is underlined in red. Below the text area is a section labeled "Predaj Zadatak:" (Submit Task:). This section contains a text input field with the value "hash2@gmail.com" and a dropdown menu showing "hash@gmail.com". At the bottom of the modal are three buttons: "AI Predaja" (AI Submit) in yellow, "Zatvori" (Close) in grey, and "Kreiraj Zadatak" (Create Task) in blue. The background of the application is dark grey and shows a list of tasks with columns for "Zadatak" (Task), "Web Dizajn", and "Logo".

Kreacija Zadataka

PHP CRUD Web Aplikacija

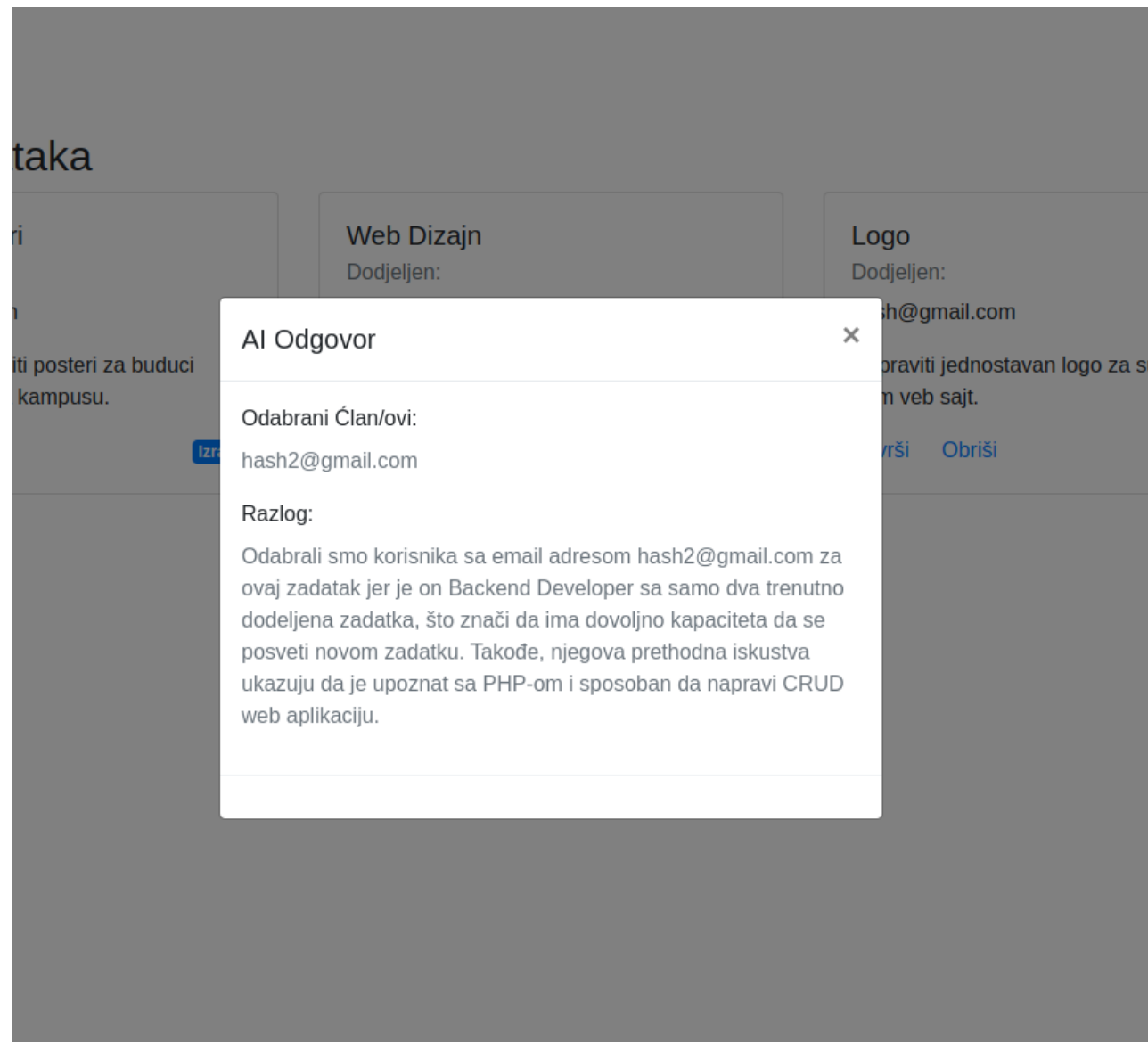
Napraviti jednostavnu CRUD web aplikaciju koristeći PHP

Predaj Zadatak:

hash2@gmail.com  
hash@gmail.com

AI Predaja Zatvori Kreiraj Zadatak

Kako neki timovi istovremeno rade sa stotinama zadataka sa desetinama ljudi, prilikom kreiranja zadatka možete kliknuti na dugme za AI. Ovo dugme će prikupiti sve informacije o ljudima u vašem timu - koje vrste zadataka su u prošlosti završavali, koliko imaju trenutno aktivnih zadataka i kojim se profesijama bave. Sve ove informacije će biti proslijeđene ChatGPT turbo 3.5, nakon čega će ChatGPT odrediti najbolje kandidate za zadatak koji trenutno kreirate. Kako biste koristili ovu funkciju, morate navesti opis novog zadatka. Ovaj sistem radi zato što web stranica pamti sve završene zadatke i ko ih je završio, tako da se ove informacije mogu koristiti za AI kasnije. Zbog toga tvrdim da moj projekat spada u oba takmičarska kategorije, jer skladišti i vizualizuje podatke, ali i poboljšava kvalitet života korišćenjem inteligentnih sistema.



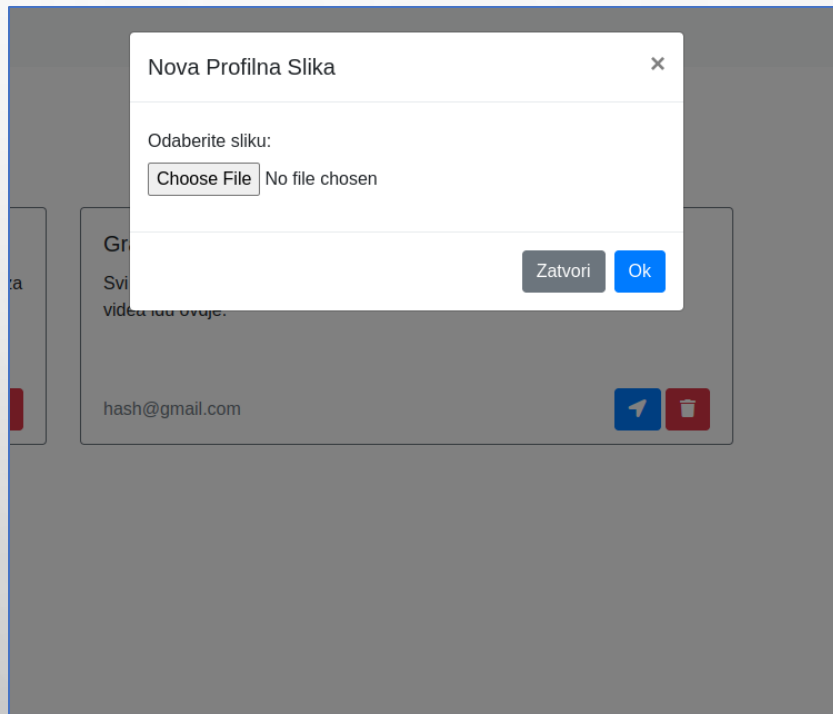
# PROFILNE SLIKE

Verovatno ste već primetili profilnu ikonu i imejl u gornjem desnom uglu vašeg ekrana.

Ovo predstavlja vaš profil.

Na početku, vaša profilna slika će biti siva avatar, ali možete je promeniti tako što ćete kliknuti na vaše ime, pojaviće se padajući meni, a zatim odaberite dugme "Promijeni Sliku". Zatim će se pojaviti novi meni koji će vas pitati da učitajte svoju profilnu sliku. I to je to! Vaša profilna slika je vidljiva vama na vašoj traci i u navigaciju i drugim ljudima u delu "članovi" bilo kog projekta.

Sada možete pokazati svoju mačku čak i svojim kolegama sa posla!

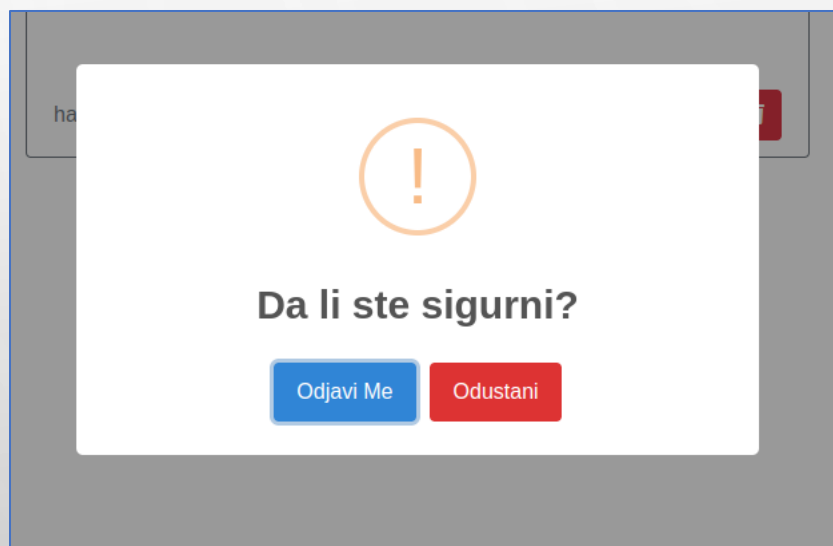




# CLOUDINARY

Za hosting vaše profilne slike koristim uslugu nazvanu Cloudinary. Ovo je sjajna usluga za male programere i velike kompanije da bi olakšala hosting slika. Koristio sam njihovu jednostavnu Java biblioteku da integrišem njihovu uslugu u projekat sa samo oko 50 linija koda. Nakon što učitate sliku, ona se šalje Cloudinary-u, nakon čega se skladišti na njihovim serverima, a zatim ja dobijem url slike, a taj url se čuva u bazi podataka i od sada je vaša nova profilna slika!

```
_id: ObjectId("64481f4af447e0547c72be1b")
firstName: "Dimitri"
lastName: "Fustic"
jobTitle: "Backend Developer"
email: "hash@gmail.com"
password: "653b18af931fe779d09cafc079522a54d2be56cfb79e8af8b436aeac66f24147"
profilePictureUrl: "https://res.cloudinary.com/drz3o1ast/image/upload/v1683214172/fzlrupv7..."
```



## ODJAVA

Možete se odjaviti sa svog profila u bilo koje vreme tako što ćete kliknuti na vašu profilnu sliku i odabrati "Odjava" iz padajućeg menija. To je korisno ako želite da zaštitite svoj profil. Dodatno, sistem će vas automatski odjaviti nakon određenog perioda neaktivnosti.

# BACK-END TEHNOLOGIJE

Backend je u potpunosti napisan u Javi. Zašto Java, možda se pitate? Pa, najviše iskustva imam u Javi i pošto je vreme bilo od suštinskog značaja, nisam želeo da mi izbor programskog jezika predstavlja ograničenje. Za veb deo projekta korišćen je Micronaut framework. Prvobitno sam pokušao da to uradim u SpringBoot-u pošto ga obično koristim, ali integrisanje Spring-ove sigurnosti je bilo previše komplikovano, a nisam želeo da jednostavno preuzmem već konfigurisan i napravljen Spring Security projekat sa Github-a, pa sam se umesto toga odlučio da naučim Micronaut od nule i koristim njegov JWT cookie sistem za sigurnost. Da, dobro ste čuli. Nemam apsolutno nikakvog iskustva sa Micronaut-om pre početka ovog projekta, a iskustvo koje imam u razvoju backend-a je vrlo malo i ograničeno kada su u pitanju baze podataka i sistemi za autentifikaciju. Jedini razlog zbog kojeg sam uspeo da napravim ovaj projekat je Božja milost.

```
aut:
sources:
staticPaths:
- public:public
location:
name: project_Manager
urity:
uthentication: cookie
oken:
jwt:
  signatures:
    secret:
      generator:
        secret: ${JWT_GENERATOR_SIGNATURE_SECRET:pleaseChangeThisSecretForAM
Redirect configuration
edirect:
  login-failure: "/login/authFailed"
  unauthorized:
    url: "/login/auth"

Allow access to resources in public folder
ntercept-url-map:
- pattern: "/public/images/**"
  http-method: "GET"
  access:
    - isAnonymous()

- pattern: "/public/js/**"
  http-method: "GET"
  access:
    - isAnonymous()

- pattern: "/public/css/**"
  http-method: "GET"
  access:
    - isAnonymous()

Allow usage of static files in /resources/public
ter:
> micronaut: > security: > authentication: > cookie
```

```

import java.util.List;

no usages
@Secured(SecurityRule.IS_AUTHENTICATED)
@Controller("/project")
public class ProjectController {

    7 usages
    private final ProjectService projectService;

    no usages
    @Inject
    public ProjectController(ProjectService projectService) { this.projectService = projectService; }

    no usages
    @Post("/new")
    @Produces(MediaType.APPLICATION_JSON)
    public HttpResponseMessage<> createProject(@Nullable @Body Project project, Principal principal) {
        if (project != null) {
            project.setOwnerEmail(principal.getName());
            project.setTeamMemberEmails(List.of( @"null"));
            project.setProjectImage(null);
            return projectService.createProject(project);
        }
        return HttpResponseMessage.badRequest();
    }

    no usages
    @Post("/delete")
    @Produces(MediaType.APPLICATION_JSON)
    public HttpResponseMessage<> deleteProject(@Nullable @Body ProjectID projectID, Principal principal) {
        if (projectID != null) {
            return projectService.deleteProject(projectID.getProjectId(), principal.getName());
        }
        return HttpResponseMessage.badRequest();
    }

    no usages
    @Get("/owner")
    @Produces(MediaType.APPLICATION_JSON)
    public HttpResponseMessage<> getOwnerProjects(Principal principal) {

itri > project_manager > controller > dashboard > ProjectController

```

Micronaut deo stvari je prilično jednostavan. Imamo kontrolere koji predstavljaju HTTP endpointe, a zatim imamo servise koji su zapravo klase poslovne logike. Nakon što se zahtev prosledi kontroleru, prosleđuje se servisnoj klasi da obavi sve poslove biznis logike.

# DATABAZA MONGO-DB

Odlučio sam se za upotrebu MongoDB baze podataka. Planirao sam korišćenje nosql baze podataka jer sam planirao drugačiju strukturu baze. Trenutno struktura baze izgleda ovako: prilikom registracije, kreira se dokument korisnika u kolekciji korisnika. Dokument projekta se kreira u kolekciji projekata, kada želimo da preuzmemo sve projekte od korisnika, pretražujemo bazu podataka za ownerEmail -> email. Dokument kategorije se kreira u kolekciji grupa, kada želimo da preuzmemo sve kategorije za određeni projekat, jednostavno pretražujemo kolekciju po projectId -> id. Razumete šta želim da kažem, koristimo strane ključeve za povezivanje svih podataka. Prvobitno sam želeo da sve podatke čuvam u jednoj kolekciji poput objekta, ali sam odustao jer je bilo previše nepregledno. Ovaj način skladištenja podataka je lep i kompaktan i lako se može prevesti u sql tabelarnu bazu podataka.

```
>
{
  "_id": ObjectId("64481f4af447e0547c72be1b"),
  "firstName": "Dimitri",
  "lastName": "Fustic",
  "jobTitle": "Backend Developer",
  "email": "hash@gmail.com",
  "password": "653b18af931fe779d09cafc079522a54d2be56cfb79e8af8b436aeac",
  "profilePictureUrl": "https://res.cloudinary.com/drz3o1ast/image/uplo..."
}

{
  "_id": ObjectId("6448405809552b096be75d7d"),
  "firstName": "Dimitri",
  "lastName": "Fustic",
  "jobTitle": "Backend Developer",
  "email": "hash2@gmail.com",
  "password": "653b18af931fe779d09cafc079522a54d2be56cfb79e8af8b436aeac",
  "profilePictureUrl": "https://res.cloudinary.com/drz3o1ast/image/uplo..."
}

{
  "_id": ObjectId("644c079d0a545a3ee6713bd2"),
  "firstName": "Dimitri",
  "lastName": "Fustic",
  "jobTitle": "Developer",
  "email": "dimitrijefustic@gmail.com",
  "password": "653b18af931fe779d09cafc079522a54d2be56cfb79e8af8b436aeac",
  "profilePictureUrl": null
}

{
  "_id": ObjectId("64525023ef76b13d5325118d"),
  "firstName": "Ethan",
  "lastName": "Armstrong",
  "jobTitle": "Student",
  "email": "eathenx@gmail.com",
  "password": "e03703c15739c069890c2fde0fcc3c279f9094c76ddd33c051ce7b97",
  "profilePictureUrl": null
}

{
  "_id": ObjectId("645250a8ef76b13d5325118e"),
  "firstName": "Ethan",
  "lastName": "Armstrong",
  "jobTitle": "Student",
  "email": "hello@gmail.com",
  "password": "653b18af931fe779d09cafc079522a54d2be56cfb79e8af8b436aeac",
  "profilePictureUrl": "https://res.cloudinary.com/drz3o1ast/image/uplo..."
}
```

# PRIKAZIVANJE FRONT END-A

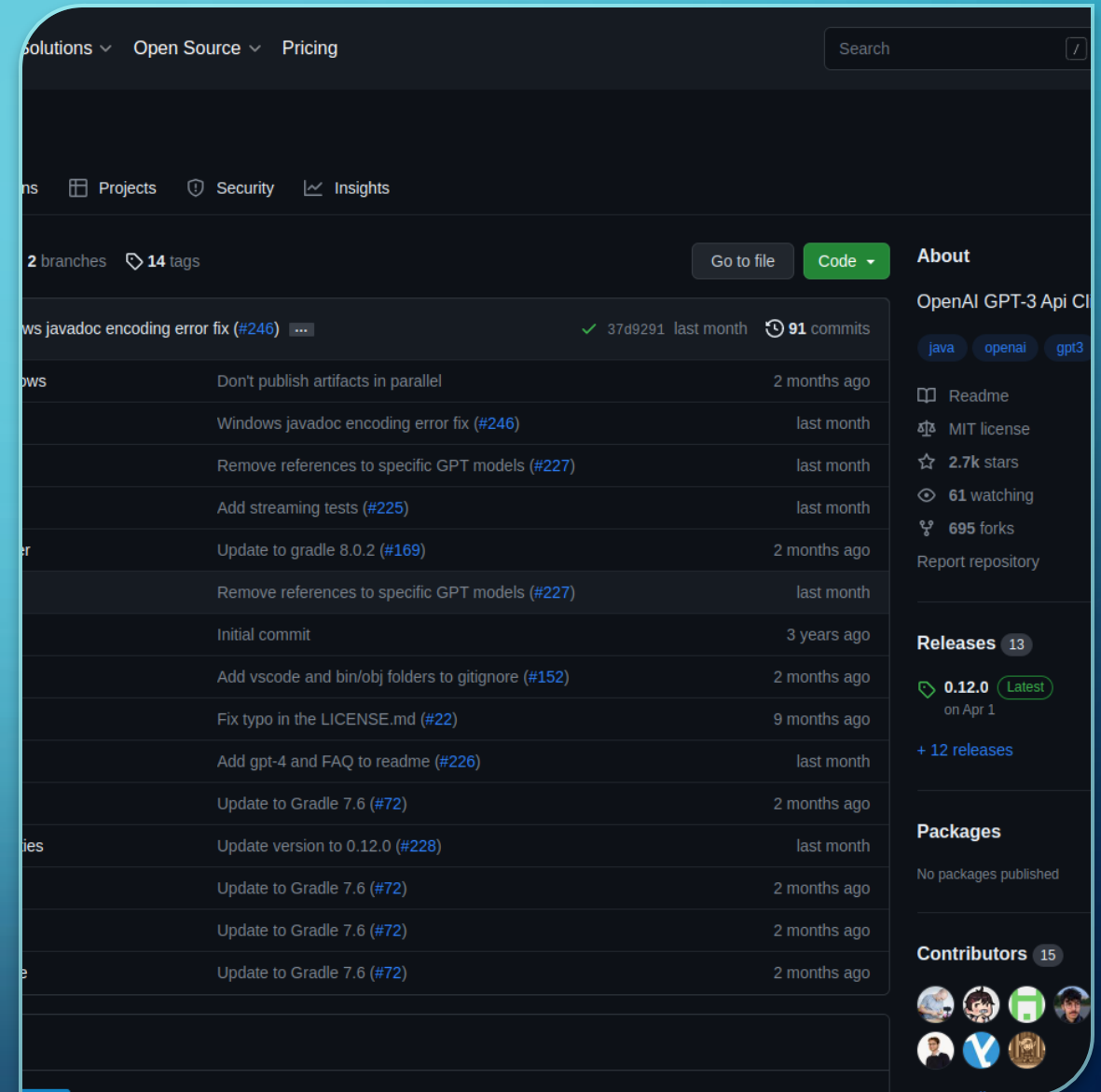
Frontend kod sam uključio u back-end kod unutar resources/views zato što sam želeo da svi moji kodovi budu na jednom mestu. Folder views koristi se za Tymeleaf, Apache Velocity itd. Ali nisam baš koristio nijednu od ovih tehnologija. Ove tehnologije koriste se za generisanje dinamičkih fajlova pre nego što se pošalju korisniku, slično kao u PHP-u. Nisam ih koristio jer sam imao problema sa namještanjem thymeleaf-a i nisam imao još vremena da naučim apache velocity. Umesto toga, koristio sam JavaScript na frontend-u da uzmem sadržaj sa backend-a i prikažem ga korisniku. Ipak, koristio sam .vm fajlove da korisnicima pokažem HTML pomoću kontrolera.

```
c(.      Ip>M[DT      i 7Pe P[| "L !yQ[~~d
'c}[mv<,D`m~  C`xb      |1<E TT@%L }u60'
[L+x(D+<)Vsm#gN`zPvM5 oz  / A,9 x'b,
7G  9  +g;i6) P(      \? jd:jjPoqg<d~@41
Je P  l1)duKd0 lM7      lYqheSyt"AR43
vw3      zc      R?8Y  %  +t  ?5ad$>d/9i0{H
DOz'4! hbRt ,*      )tz +S Ua#>a
      oj  w4  <no code>      5 pID6( Ou028
IPDm] [$nQ      .<F  _/p_QfK-qUj
*>s UqB Xpy      =xLc[++ 7^ 7D9bQ0: -6{
8o ~  6@9 ;Ku \? 2>      9 M7W'| #Lr2S?0
3'E  V^S=SU?  U      ~k      ;_prKBY_yIPr}
Z*- j=TeK PO9] L%      cs ' Oi lN's7\X9
C3#"L$p  x34oNW]V      \PuA'9Dz2F Sm'6Hu
^ m4y$G<t8LcF  0      [ w`#|bTk-k3
65  L'x7=VF62=  7  7UT^NKh0  77
```

# OPEN AI BIBLIOTEKA

Takođe, koristio sam još dvije vanjske biblioteke za backend. Jedna je za hosting slike o kojoj sam već ranije govorio, a druga je openai na github-u:

<https://github.com/TheoKanning/openai-java> Ova biblioteka mi omogućava jednostavno interagovanje sa openai-jevim AI modelima, bez pisanja vlastitog http klijenta i čitanja odgovora. Jednostavna je za upotrebu i implementaciju, i nije mi trebalo više od jedne java klase da je pokrenem.





# FRONT-END TEHNOLOGIJE

```
function showSpinner() {
  document.querySelector('.spinner-container').classList.remove('d-none');
}

function hideSpinner() {
  document.querySelector('.spinner-container').classList.add('d-none');
}

function errorAlert(text) {
  Swal.fire({
    icon: 'error',
    title: "Neuspjeh",
    text: text,
    confirmButtonColor: '#3085d6',
    confirmButtonText: 'Ok',
    allowOutsideClick: false,
    allowEscapeKey: false
  });
}

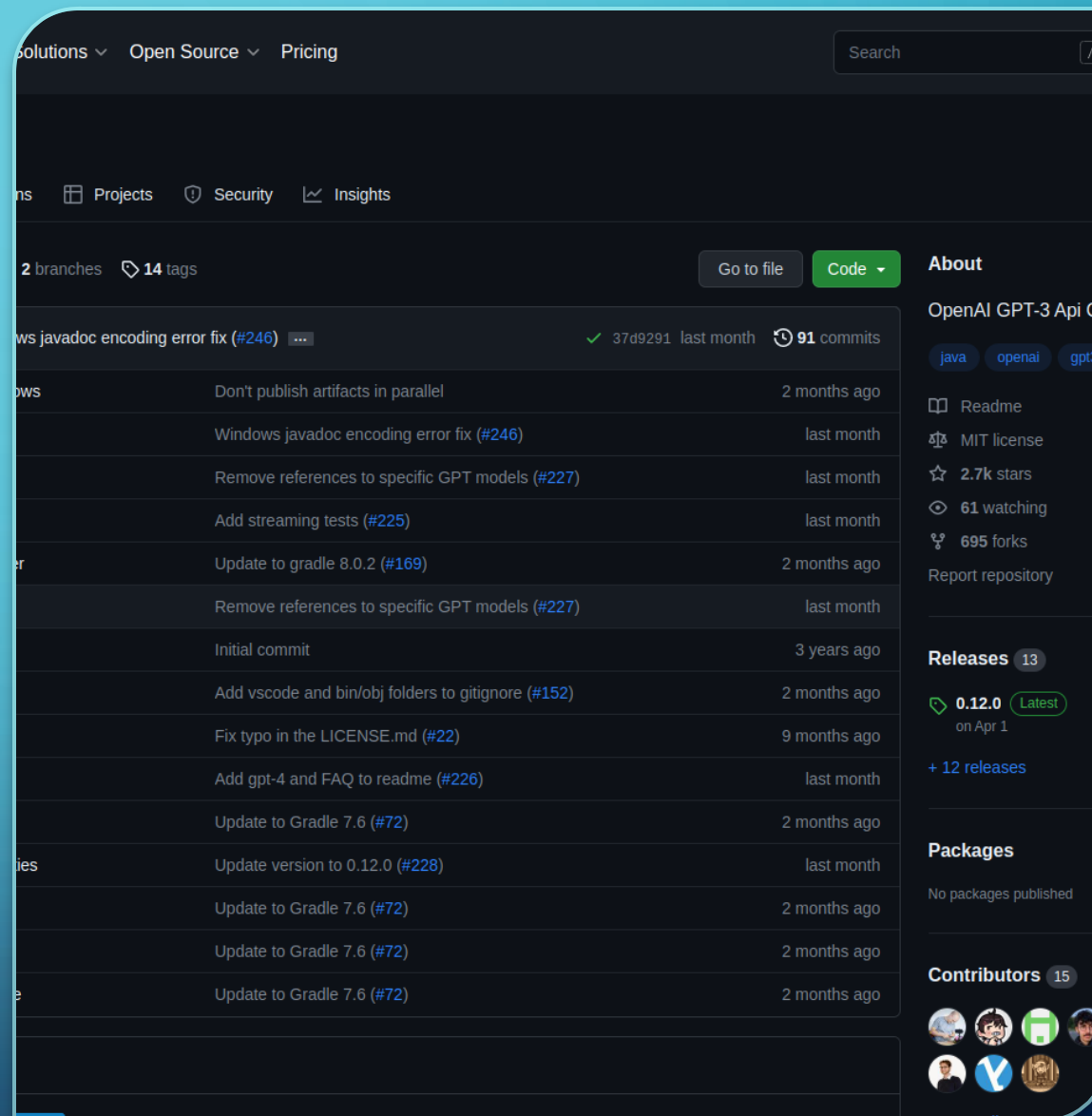
function successAlert(text) {
  Swal.fire({
    icon: 'success',
    title: "Uspjeh",
    text: text,
    confirmButtonColor: '#3085d6',
    confirmButtonText: 'Ok',
    allowOutsideClick: false,
    allowEscapeKey: false
  }).then((result) => {
    if (result.isConfirmed) {
      location.reload();
    }
  });
}

function toggleLogoutWarning() {
  Swal.fire({
    title: 'Da li ste sigurni?',
    icon: 'warning',
  });
}
```

Front-End je prilično jednostavan i malen. Za prikazivanje svih sadržaja iz back-end-a, kao i za prikazivanje kruga za učitavanje koji se pokaže kada se vrši neka akcija, koristio sam vanilla javascript. Takođe sam koristio Bootstrap 4 za dizajn veb stranice i napisao samo oko 50 linija CSS-a. Loš sam u dizajniranju front-end-a i nikada nisam mnogo radio na tome, pa mi je bootstrap stvarno pomogao ovde. Sajt je responsive i izgleda dobro na mobilnim uređajima i izgleda dobro na gotovo svakoj veličini ekrana, kažem na skoro svakoj jer verovatno izgleda prilično loše na velikim ekranima poput televizora. Za HTML sam koristio običan stari HTML5 i dobro je obavio posao. Jedina druga biblioteka koju sam koristio bila je biblioteka sweet alert za dodavanje lepo izgledajućih alertova.



Ceo proces dobijanja sadržaja iz pozadine obavlja se putem funkcije fetch u javascriptu, nakon čega se sve prikazuje. Upotreba nekog front-end frameworka poput React ili Angular bila bi idealna, ali nisam imao vremena da naučim nešto od toga. Možda se takođe pitate kako sam dodao funkcionalnost dugmadima, na primer, označavanje zadatka kao završenog? Pa kada javascript dobije sadržaj sa back-end-a (back-end odgovara u json formatu što sam zaboravio da pomenem), takođe stavlja id zadatka unutar kontejnera zadatka, tako da kada god se neka funkcija mora obaviti u vezi sa zadatkom, sve informacije koje veb sajt treba da bi uradila tu funkciju su već smeštene u front-endu.



# KRAJ

Projektni menadžer može koristiti bilo ko za bilo koji projekat. Ne mora biti kompanija koja ga koristi za IT ili grafički dizajn, može biti pojedinac koji ga koristi za organizovanje svog dana, pejzažeri, građevinski radnici, modni dizajneri, restorani, prodavnice i slično. S obzirom da koristi ChatGPT, njegove mogućnosti preporuka zadataka su neograničene i jedino što vas ograničava je vaša mašta (i možda manje idealan dizajn interfejsa haha).

