

INSTITUT ZA MATEMATIKU I INFORMATIKU PRIRODNO-MATEMATIČKI FAKULTET

Letnja praksa

### Aplikacija za upravljanje preduzećem

#### Mentor Student

#### dr Marina Svičević Andrija Lazić 53/2019

**Sadržaj**

[Uvod 3](#_Toc174279147)

[Opis stručne prakse, projekta i ciljevi 3](#_Toc174279148)

[Zadaci 3](#_Toc174279149)

[Dizajniranje baze podataka 3](#_Toc174279150)

[Dizajniranje API-a 4](#_Toc174279151)

[Logs 4](#_Toc174279152)

[Exception middleware 4](#_Toc174279153)

[Socket-i 4](#_Toc174279154)

[Dizajniranje frontend-a 5](#_Toc174279155)

[Login 5](#_Toc174279156)

[Registracija 5](#_Toc174279157)

[Kalendar rada 6](#_Toc174279158)

[Token interceptor 6](#_Toc174279159)

[Čet 6](#_Toc174279160)

[Testiranje 7](#_Toc174279161)

[Zaključak 7](#_Toc174279162)

# Uvod

Ovaj seminarski rad se fokusira na iskustva stečena tokom letnje prakse, sa posebnim naglaskom na

samostalnom radu na projektu za upravljanje preduzećem. Tokom prakse, imao sam priliku da radim na

razvoju, implementaciji i testiranju rešenja koje optimizuje poslovne procese i poboljšava efikasnost unutar

organizacije.

# Opis stručne prakse, projekta i ciljevi

Naziv projekta tokom stručne prakse bio je " **CompanyMenagmentProject**".

Osnovna namena projekta " CompanyMenagmentProject" bila je razvoj sveobuhvatnog softverskog alata koji omogućava lako i efikasno upravljanje različitim aspektima radnog okruženja. Konkretno, alat bi trebalo da obezbedi:

- način komunikacije između radnika i menadžera

- informacioni sistem za upravljanje nabavkama, kalendara rada...

Ovaj projekat se izvodio od 01.06.2024. - 25.6.2024.

# Zadaci

Moje uloge u projektu su bile:

- Dizajniranje aplikacije

- Implementacija rešenja

- Izrada testova

Ceo proces razvoja smo započeli sa određivanjem korisničkih zahteva i na osnovu njih smo izabrali odgovarajuće tehnologije:

1. **Angular 17**: Framework zasnovan na typescript-u i bibliotekama Zone.js i RxJS. Omogućava nam brz i olakšan razvoj frontend dela aplikacije.
2. **.NET 8**: Moderan i skalabilan framework zasnovan na c#-u. Predstavlja osnovu za razvoj backend dela naše aplikacije. Poseduje moćne alate za obradu podataka, poslovne logike i komunikaciju sa frontendom i bazom podataka.
3. **MS SQL Server**: Sql baza podataka koja nam je omogućila efikasno skladištenje, upravljanje i brzu pretragu informacija.
4. **Xunit**: Framework koji omogućava unit i E2E testiranje .net aplikacija.

## Dizajniranje baze podataka

Zatim sam odlučio da iskoristim Data base fist pristup u kojem ću prvo kreirati bazu podataka sa odgovarajućim tabelama i stornim procedurama.

Nakon toga je potrebno da u .NET projektu import-ujemo Entity Framework i izvršimo reverse enginering

pomuću Scaffold funkcije. Ovime smo SQL tabele pretvorili u modele podataka na backend-u čime smo uštedeli vreme na razvoja.

## Dizajniranje API-a

Razvijen u .NET framework-u i uz pomoć layerd arhitekture koja nam omogućava da organizujemo softverski sistem u odvojene slojeve, pri čemu svaki sloj ima specifičnu odgovornost.

Ceo .Net framework se zasniva dependency injection-u odnosno paternu koji treba da obezbedi povećanu fleksibilnost, lakše testiranje i bolju organizaciju koda.

Ceo istem se sastoji iz četiri sloja:

1. Main sloj: Predstavlja najviši sloj aplikacije sa kojim korisnik komunicira preko kontrolera. U ovom sloju se kreiraju svi potrebni dependency-i i konfiguracija aplikacije. Ovaj sloj treba da komunicira samo sa BLL slojem.
2. BLL sloj (Biznis logic layer): Sloj koji sadrži servise i logiku za obradu podata. Direktno komunicira samo sa DAL slojem.
3. DAL sloj (Data access layer): Sloj koji je zadužen za komunikaciju sa bazom podataka. Za komunikaciju se koriste Linq upiti koji se pri izvršavanju pretvaraju u SQL-upite. Moguće je pozivati storne procedure ukoliko su nam potrebne bolje performanse ili kompleksniji upiti.
4. Domain sloj: Sadrži sve modele i abstrakcije.

## Logs

Logovanje događaja i grešaka se vrši pomoću Serilog biblioteke koja nam omogućava razvoj logera po našim potrebama. U našem primeru se logovi čuvaju u CSV fajlu što nam omogućava uvid u sva dešavanja u apkaciji.

A screenshot of a computer

Description automatically generated

## Exception middleware

Globalni middleware koji nam olakšava upravljanje exception-a na svim nivoima aplikacije. Kada na nekom od slojeva dođe do bacanja exception-a isti se prosleđuje middleware-u koji ga obrađuje i obaveštava korisnika o grešci.

## Socket-i

Za komunikaciju između korisnika su korišćeni socket-i pomoću biblioteke SignalR. Socket-i nam omogućavaju real time komunikaciju između korisnika i servera koja nam je potrebna za izradu četa.

## Dizajniranje frontend-a

Angular u sebi sadrži brojne alate i sisteme koji nam olakšavaju razvoj aplikacije.

Razvoj započinjemo instalacijom biblioteka kao što su PrimeNG, AngularMaterials i Bootstrap. Ove biblioteke u sebi sadrže komponente i alate koje ćemo iskoristiti za izradu naših komponenata.

Jedan Angular projekat se sastoji iz sledećih elemenata:

* Moduli: Konfiguracione klase u kojima su definisane komponente, servisi, importovane biblioteke... Od angulara 17 više nisu obavezni ali se i dalje mogu koristiti za organizaciju koda i lazyloading.
* Komponente: Osnovni blok gradnje. Sastoji se od HTML-a, css-a i logike. Od Angulara 17 postaju standalone i više nije obavezno da se nalaze u sklopu modula.

Pored ovih onsovnih elemenata imamo mogućnost i da koristimo servise, guard-ove i interceptor-e.

## Login

Kada se korisnik loguje sa back-a se vracaju JWT token i Reset token.

JWT sadrži podatke o korisniku i validan je za kratki vremenski period. Koristi se za autentifikaciju korisnika prilikom slanja HTTP zahteva.

Reset token se koristi za resetovanje JWT tokena kada isti istekne. Salje se samo prilikom resetovanja JWT i čuva se u cookie-u čime je obezbeđena njegova sigurnost.

## Registracija

Samo Admin može da registruje nove korisnike i menja postojeće.

A screenshot of a computer

Description automatically generated

## Kalendar rada

Obični radnici mogu da pristupe svom kalendaru rada i provere da li je došlo do promene rasporeda u radu i da li se nekog dana radi ili ne.

Samo Admin i Menadžer mogu da menjaju podatke vezane za kalendar.

A screenshot of a computer

Description automatically generated

## Token interceptor

Middleware koje presreće HTTP zahtev pre nego što se pošalje na back i dodaje JWT token u header zahteva. Ovim smanjujemo nepotreban kod i olakšavamo upravljanje JWT tokenom.

## Čet

Razvijen pomoću SignalR biblioteke. Koristi socket-e za komunikaciju u realnom vremenu.

A screenshot of a computer

Description automatically generated

## Testiranje

Za testiranje je korišćen XUnit framework koji nam je omogućio dva načina testiranja:

- E2E (End to End) testiranje: pozivanje API-a preko HTTP zahteva i pregled validnosti rezultata(Testiramo ceo sistem)

- Unit testiranje: testiranje pojedinačnih elemenata u sistemu.

A screenshot of a computer

Description automatically generated

# Zaključak

Tokom stručne prakse, imao sam priliku da se upoznam sa konceptima razvoja jedne aplikacije od početka do kraja. Ovo iskustvo je obuhvatalo različite faze razvoja, uključujući planiranje, dizajn, implementaciju i testiranje.

Naučio sam da koristim Angular i .NET framework na višem nivou i efikasno koristim njihove alate. Jednako bitno je i znanje iz Xunit framework-a koji nam je omogućio da pišemo testove i time sebi olakšamo dalji razvoj aplikacije.