DISLINKT

XWS PROJEKAT

Razviti poslovnu društvenu mrežu za povezivanje poslodavaca i osoba koji traže posao.

Učesnici/Korisnici sistema

- Neregistrovani korisnik - pretražuje i razgleda postove javnih profila.
- Registrovani korisnik - pored pretrage i pregleda postova, ima mogućnost kačenja sopstvenih postova. Takođe ima mogučnost pretraživanja i kreiranja ponuda za posao kao i dopisivanje sa svojim pratiocima i konekcijama.
- Administrator - ima kompletan pristup i nadzor sistemu.

Moduli sistema

- **Front-End Dislinkt-a** - Obezbeđuje interfejs i funcionalnosti neophodne korisnicima i administratorima.
- **Back-End Dislinkt-a** - Sadži kompletnu poslovnu logiku aplikacije koja se zasniva na mikroservisnoj arhitekturi.
- **Agensta aplikacija** (Front-End i Back-End) - predstavlja web aplikaciju za postavljanje ocena i komentara odredjenih poslovnih kompanija (monolitna apliklacija). Koristi API koji nudi Back-End Dislinkt-a kako bi promovisala otvorene ponude za posao.

Funkcionalnosti

Dislinkt

1. Obezbediti sledeće funkcionalnosti **neregistrovanim korisnicima**:
    1. Pretragu javnih profila.
    2. Pregled postova na javnim profilima.
    3. Registracija na sistem (obavezan jedinstven *username*).

2. Obezbediti sledeće funkcionalnosti **registrovanim korisnicima**:
    1. Sve funkcionalnosti kojima raspolaže neregistrovani korisnik.
    2. Logovanje na sistem.
    3. Preporučivanje drugih profila kao potencijalne konekcije. **Za implementaciju neophodno je koristiti graf bazu**.
    4. Preporučivanje otvorenih ponuda za posao na osnovu prethodnog radnog iskustva i veština. **Za implemntaciju neophodno je koristiti graf bazu**.
    5. Zapraćivanje drugih profila (uspostavljanje konekcija):
        a. Zapraćivanje treba da omogući:
            i. prikaz postova prilikom njihovog kreirana od strane zaprećenog profila,
            ii. razmenjivanje poruka između zaprećenih profila,
            iii. pregled ranije odbjavljenih postova.
        b. Zapraćivanje javnog profila je uvek omogućeno.
        c. Zapraćivanje privanog profila mora da se odobri od strane privatnog profila.
    6. Objavljivanje postova čiji sadržaj može da obuhvati tekst, sliku i linkove. Post se trajno nalazi na korisnikovom profilu. Prilikom kreiranja posta, neophodno je podeliti ga sa ostalim pratiocima. Post omogućava sledeće opcije: like, **dislike**, pisanje komentara.
    7. Razmenjivanje poruka sa drugim zapraćenim profilima.
    8. Kreiranje i pretragu ponuda za posao. Ponuda treba da obuhvati poziciju, opis posla i dnevnih aktivnosti, preduslove koje kandidat mora da ispuni.
    9. Ažuriranje i podešavanje profila:
        a. Dodavanje i izmena ličnih podataka: ime, email, broj telefona, pol, datum rođenja, *username* i biografija.
        b. Podešavanje radnog iskustva i obrazovanja.
        c. Podešavanje veština i interesovanja.
        d. Podešavanje privatnosti profila:
            i. Da li je profil javan ili privatan.
            ii. Blokiranje drugih profila, nakon čega sa istima nije moguće vršiti bilo kakav vid interakcije.
        e. Podešavanje notifikacija za zapraćene profile, poruke i postove.

10. Upravljanje API tokenima koji omogućavaju pristup funkcionalnostima koje se odnose na kreiranje ponude za posao za eksterne aplikacije. **Token ne sme da autorizuje pristup ostalim funkcionalnostima**

4. Obezbediti sledeće funkcionalnosti **administratorima**:
    1. Pregled svih događaja koji su se desili u sistemu.

5. Monitoring:
    1. Tracing: Neophodno je implementirati tracing nad svim servisima mikroservisne aplikacije i prikazati u nekom alatu za vizualizaciju.
    2. Logging: Neophodno je implementirati agregraciju logova nad svim servisima mikroservisne aplikacije i prikazati u nekom alatu za vizualizaciju.
    3. Metrics:
        a. Metrike operativnog sistema host mašine na kojoj će mikroservisna aplikacija biti podignuta. Minimum treba obezbedtiti informacije o iskorišćenju procesora, RAM memorije, file sistema i protok mrežnog saobraćaja.
        b. Metrike kontejnera koji se koriste u mikroservisnoj aplikaciji. Minimum treba obezbedtiti informacije o iskorišćenju procesora, RAM memorije, file sistema i protok mrežnog saobraćaja.
        c. Mektrike Web saobraćaja u mikroservisnoj aplikaciji:
            i. Ukupan broj HTTP zahteva u prethodnih 24 sata.
            ii. Broj uspešnih HTTP zahteva u prethodnih 24 sata (2xx, 3xx).
            iii. Broj neuspešnih zahteva u prethodnih 24 sata (4xx, 5xx).
            iv. Broj jedinstvenih posetilaca (ista IP adresa, timestamp i web browser).
            v. Broj neuspešnih zahteva sa statusom 404 sa njihovim *endpoint*-ovima u prethodnih 24 sata.
            vi. Ukupan protok saobraćaja izražen u GB.
        d. Metrike je neophodno prikazati u nekom alatu za vizualizaciju.

Agentska apllikacija

6. Obezbediti sledeće funkcionalnosti u **agentskoj aplikaciji**:
    1. Registracija korisnika.
    2. Registovani korisnik ima mogućnost da kreira zahtev za registraciju kompanije. Prilikom registracije nepohodno je ostaviti kontakt informacije same kompanije, kratak opis delatnosti, kulture itd. Admin mora da potvrdi posalti zahtev za registraciju kompanije. Nakon uspešne potvrde, korisnik dobija rolu vlasnika kompanije kojom ima pravo da ažurira opis sadržaja kompanije i postavlja otvorene pozicije za posao, ali nema pravo da ostavlja komentare, plate, ocene i proces selekcije kandidata (interjvua).
    3. Kreiranje komentara, ostavljanje iznosa plata za određenu poziciju, procesa intervjua od strane registrovanih korisnika.
    4. Objavljivanje otvorenih ponuda za posao od strane vlasnika kompanije koje koriste API Delinkt aplikacije namenjen za ponude za posao.

Način realizacije projekta i ocenjivanje

Projekat se realizuje timski, pri čemu timovi broje do 4 člana. Studenti treba da:

- Razviju model podataka neophodan za realizaciju kompletnih funkcionalnosti (Analizirati koji podaci se koriste u sistemu kao i koje međuzavisnosti postoje).
- Definišu neophodne komunikacije kako bi ceo sistem funkcionisao na adekvatan način (pri čemu treba voditi računa o tome da je Dislinkt neophodno razviti kao skup mikroservisa).
- Realizuju sve navedene funkcionalnosti vodeći računa o svim graničnim slučajevima, odnosno omogućiti pravilno funkcionisanje aplikacije (**za sve slučajeve koji nisu pokriveni u specifikaciji studentima se daje mogućnost da ih reše na način koji je njima najprikladniji**).

Za ocenu 6 neophodno je implmentirati:

- Funkcionalnosti: 1.1, 1.2, [illegible], 2.9.a, 2.9.b, 2.9.c, [illegible], 2.6
- [illegible]

Za ocenu 7 neophodno je implementirati:

- Sve stavke navedene za ocenu 6.
- Funkcionalnosti: 2.8, 2.10, 6.1, 6.2, 6.3, 6.4
- Za sihronu komunikaciju između servisa neophodno je koristiti gRPC

Za ocenu 8 neophodno je implementirati:

- Sve stavke naveden za ocenu 7.
- Funkcionalnosti: [illegible]d, 2.9.e, 2.7.
- Svi servisi moraju da se izvršavju u kontejnerima i definisati *docker compose* za njihovo pokretanje.
- Primeniti SAGA obrazce za sihronizaciju podataka između mikroservisa.

Za ocenu 9 neophodno je implementirati:

- Sve stavke navedene za ocenu 8.
- Funkcionalnosti: [illegible], 2.4.

Za ocenu 10 neophodno je implementirati sledeće:

- Sve stavke navedene za ocenu 9.
- Funcionalnosti: 4.1
- *Monitoring* Delinkt mikroservisne aplikacije pomoću *Prometheus*, *Fluentbit*, *Grafana* i *Jaeger* alata.

Bezbednost u sistemima elektronskog poslovanja

Dislinkt rešenje je neophodno obezbediti integracijom bezbednosnih kontrola u njegove module, kao i uvođenjem bezbednosnih alata koji su opisani u ovom poglavlju.

PKI

Implementirati alat za podršku infrastrukture javnih ključeva. Specifikacija projektnog zadatka je definisana kroz skup funkcionalnih i nefunkcionalnih zahteva za rad sa sertifikatima i ključevima. Potrebno je dizajnirati i implementirati PKI vođeni ovim zahtevima.

Bezbednost modula

Potrebno je obezbediti čitav Dislinkt sistem. Svaku bezbednosnu kontrolu treba integrisati prateći *best practice* konfiguraciju i šablone bezbednog dizajna (višeslojna odbrana, najmanja privilegija, jednostavan dizajn, itd.).

Zaštita podataka

Osetljivi podaci sa kojim aplikacija radi treba da budu obezbeđeni u skladištu, u transportu i tokom upotrebe. Identifikovati osetljive podatke, definisati i implementirati prikladne bezbednosne kontrole. Podaci čije skladištenje se ne može izbeći treba da budu šifrovani ili heširani ukoliko je to prikladno. Poruke u internoj komunikaciji treba da imaju očuvanu poverljivost, integritet i neporecivost, kao i da budu zaštićene od *replay* napada. Komunikacija između veb-čitača i servera treba da bude zaštićena sigurnom konfiguracijom HTTPS protokola. Sertifikate generisati putem PKI alata.

Kontrolna pristupa

Moduli sistema treba da podrže prikladne mehanizme za autentifikaciju i autorizaciju. Mehanizmi autentifikacije treba da podrže bezbednu registraciju, prijavu na sistem pomoću lozinke, prijavu na sistem bez unosa lozinke (tzv. *passwordless* prijava), odjavu, promenu lozinke i oporavak naloga. Autorizacija podrazumeva kontrolu pristupa po RBAC modelu.

Zadatak za 9

Potrebno je da svaki student iz tima kreira model pretnji za proizvoljno odabran mikroservis. Model pretnji podrazumeva sagledavanje ranjivosti, pretnji, napada, kontrola za njihovo sprečavanje i negativnih uticaja uz dijagrame arhitekture servisa i dijagrame tokova podataka. Pored dijagrama, potrebno je odgovoriti na sledeća pitanja:

- Odakle napadač može da sprovede napad i pod kojim uslovima?
- Kakve napade može da sprovede i koje su posledice?

- Na koji način se napadi mogu sprečiti?

Zadatak za 10

Za najvišu ocenu je neophodno realizovati **sve prethodne stavke** i jednu od celina definisanih u ovom poglavlju.

Single sign-on

Potrebno je omogućiti single sign-on (u daljem tekstu SSO) prijavu na kompletan sistem. Mehanizam za SSO se mora implementirati konfigurisanjem gotovih rešenja, poput Active Directory ili Keycloak i njihovom integracijom sa ostatkom sistema.

Penetration testing

Sprovesti penetraciono testiranje modula sistema upotrebom bar dva alata iz grupe: Nmap, Nikto, dirbuster, sqlmap, OWASP ZAP, Burp Suite. Kao rezultat penetracionog testiranja, alati nude generisan izveštaj. Potrebno je priložiti izveštaj pentesting alata i regulisati ranjivosti.

Two-factor authentication

Potrebno je mogućiti dvofaktorsku prijavu na sistem, gde bi se od korisnika pored lozinke zahtevalo još nešto što *“korisnik zna ili poseduje”*. Mehanizam se može implementirati pomoću TOTP (Time-based One Time Password) koji bi generisao Google Authenticator ili Microsoft Authenticator.

Zadatak za dodatne poene

Za dodatnih 5 poena integrisati postojeći sistem sa nekim od alata za statičku analizu koda (SonarCloud, SonarCube, …). Analizirati i rešiti ranjivosti u skladu sa preporukama alata.