Interview Skripta: .NET / SQL / React

OSNOVE .NET / C# PITANJA

1. Koja je razlika izmedu class i struct u C#?
- class je referentni tip, smjesten na heap, podrzava nasljedivanje.
- struct je vrijednosni tip, smjesten na stack, nema nasljedivanja.

2. Razlika izmedu ref, out, i in parametara:
- ref: mora biti inicijaliziran prije poziva.
- out: mora biti inicijaliziran unutar metode.
- in: samo za citanje.

3. Sto je garbage collector u .NET-u?
- Automatizirani sustav upravljanja memorijom koji cisti neiskoristene objekte.

4. SOLID principi (nabroj i objasni):
- S - Single Responsibility Principle: Klasa treba imati samo jednu odgovornost.
- O - Open/Closed Principle: Kod treba biti otvoren za prosirenje, zatvoren za izmjene.
- L - Liskov Substitution Principle: Objekti podklasa moraju biti zamjenjivi s objektima nadklasa.
- I - Interface Segregation Principle: Klijenti ne trebaju ovisiti o suceljima koja ne koriste.
- D - Dependency Inversion Principle: Ovisnosti trebaju biti prema apstrakcijama, a ne prema konkretnim k

5. Razlika izmedu IEnumerable i IQueryable:
- IEnumerable izvrsava upite u memoriji, koristi se za kolekcije u aplikaciji.
- IQueryable prevodi upit u SQL i izvrsava ga na bazi podataka.

6. Asinhroni kod u C# (async/await):
```
public async Task<string> GetDataAsync() {
    var response = await httpClient.GetStringAsync("url");
```

```
    return response;
}
```

7. Dependency Injection u .NET-u:

```
services.AddScoped<IMyService, MyService>();
```

8. Razlika izmedu Task i Thread:

- Thread: fizicka nit koja se izvrsava na procesoru.
- Task: apstrakcija koja koristi niti ispod haube, pogodna za asinhroni kod i upravljanje resursima.

9. Sto je async void i zasto ga treba izbjegavati?

- async void se koristi samo za event handlere. Inace je losa praksa jer ne mozete await-ati i ne mozete uh

10. Sto su Extension metode?

- Omogucuju dodavanje metoda postojecim tipovima bez mijenjanja izvornog koda:

```
public static class StringExtensions {
    public static bool IsCapitalized(this string input) {
        return !string.IsNullOrEmpty(input) && char.IsUpper(input[0]);
    }
}
```

11. Sto je LINQ i kako se koristi?

- LINQ (Language Integrated Query) omogucuje upite nad kolekcijama:

```
var result = list.Where(x => x > 10).ToList();
```

12. Sto je null-coalescing operator ?? i ??=?

- a ?? b: vraca a ako nije null, inace b.
- x ??= y: ako je x null, dodjeljuje y.

13. Sto je var, a sto dynamic?

- var: staticki tip koji se odreduje u vrijeme kompajliranja.
- dynamic: tip se odreduje u vrijeme izvodenja, sporiji, rizicniji.

14. Kako se hendla iznimka u C#?

```
try {
    // kod koji moze baciti iznimku
} catch(Exception ex) {
    // rukovanje greskom
} finally {
    // izvrsava se uvijek
}
```

SQL PITANJA

1. Razlika izmedu WHERE i HAVING:
- WHERE: koristi se prije grupiranja.
- HAVING: koristi se nakon GROUP BY za filtriranje grupa.

2. Normalizacija baza podataka:
- Proces uklanjanja redundancije i povecanja integriteta podataka.
- 1NF, 2NF, 3NF - najosnovniji oblici.

3. SQL za drugi najveci iznos:

```
SELECT MAX(Salary) FROM Employees
WHERE Salary < (SELECT MAX(Salary) FROM Employees);
```

4. Indexi u SQL-u:
- Pomazu u brzom pronalasku redova.
- Negativna strana: usporavanje INSERT, UPDATE i DELETE operacija.

5. Primarni i strani kljucevi:
- PRIMARY KEY: jedinstveni identifikator unutar tablice.
- FOREIGN KEY: referenca na primarni kljuc druge tablice.

6. Transakcije i ACID:

- A - Atomicity
- C - Consistency
- I - Isolation
- D - Durability

7. Razlika DELETE vs TRUNCATE:

- DELETE: moze imati WHERE, sporije, logira se.
- TRUNCATE: brise sve redove bez logiranja, brze.

.NET + Entity Framework

1. Kako koristis EF Core migracije?

dotnet ef migrations add InitialCreate

dotnet ef database update

2. Lazy vs Eager loading:

- Lazy: navigacijski podaci se ucitavaju kad su potrebni.
- Eager: Include() se koristi za odmah ucitavanje podataka.

3. Fluent API vs Data Annotations:

- Fluent API: koristi se u OnModelCreating().
- Anotacije: npr. [Required], [MaxLength], [ForeignKey].

REST API PITANJA

1. Statusni kodovi:

- 200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, 404 Not Found, 500 Internal Server Error

2. REST vs SOAP:

- REST je laksi, koristi HTTP metode i JSON.

- SOAP koristi XML, kompleksniji je.

3. Autentikacija i autorizacija:

- JWT tokeni

- OAuth2

REACT PITANJA

1. Sto je JSX?

- Sintaksa koja omogucava pisanje HTML-a unutar JavaScript-a.

2. Komponenta kao funkcija vs klasa:

- Funkcijska komponenta koristi useState, useEffect, itd.

- Klasna koristi this.state, componentDidMount...

3. useState primjer:

const [count, setCount] = useState(0);

4. Props vs State:

- props dolaze izvana (od roditelja), state je lokalno stanje komponente.

5. Lifecycle metode (React Hooks):

- useEffect(() => { ... }, [dependencies])

6. Sto je virtual DOM?

- Abstraktni prikaz stvarnog DOM-a. React koristi virtual DOM za brzo rerenderiranje.

DODATNI SAVJETI ZA RAZGOVOR:

- Pripremi se za bijelu plocu: nacrtaj relacije baza, flow komponente, API komunikaciju.
- Pitaj o:
  - CI/CD procesima (koriste li Azure DevOps?)
  - Kako izgleda jedan "tipican" projekt
  - Koje softvere koriste u testiranju transformatora