



Univerzitet u Sarajevu
Elektrotehnički fakultet u Sarajevu
Odsjek za računarstvo i informatiku



REFACTORING

The Smart Café

Članovi tima:

- Emina Efendić
- Hana Mahmutović
- Tajra Selimović

Refactoring je proces poboljšanja strukture i kvalitete postojećeg koda, bez mijenjanja njegovog vanjskog ponašanja. Cilj refactoringa je olakšati održavanje, poboljšati čitljivost i razumljivost koda, te smanjiti dupliranje i kompleksnost. Pri izradi naše web aplikacije u ASP.NET-u, uspješno smo primijenile nekoliko važnih refaktoriranja kako bismo postigle čist i strukturiran kod. Ovdje je pregled nekih od tih refaktoriranja:

1. Automatsko generisanje koda: S velikom pažnjom smo koristile alate za automatsko generisanje koda kako bismo smanjile potencijalne greške i osigurale dosljednost. Time smo uštedjele vrijeme i poboljšale kvalitetu koda, jer se automatski generisani kod često temelji na dobro utvrđenim obrascima.
2. Uklanjanje nepotrebnih datoteka: Da bismo poboljšale sigurnost i spriječile pristup korisnika određenim datotekama, odlučile smo izbrisati neke automatski generisane datoteke kao što su "create", "edit" i "delete" datoteke u MenuPage-u, AdminPanel-u, BartenderPanelu i ModifyOrder-u. To je odličan pristup jer onemogućava korisnicima pristup tim funkcionalnostima putem browsera.
3. Izdvajanje istog koda u metode: Kroz naš AdminPanelController, primijenile smo princip DRY (Don't Repeat Yourself) tako što smo kreirale metodu *MostUsedDrink* koju smo pozivale na više mjesta, konkretno u metodama *MostSoldDrink* i *Clone*, umjesto ponavljanja istog koda. Ovo poboljšava čitljivost i održavanje, jer svaka promjena u toj metodi automatski se odražava na sva mjesta gdje je metoda pozvana.
4. Ispravna terminologija: Pridržavale smo se sintaksnih pravila i odabirale deskriptivna imena za klase, metode i attribute koji odgovaraju stvarnim pojmovima iz stvarnog života. Ovo olakšava razumijevanje koda i smanjuje potrebu za dodatnim komentarima ili objašnjenjima.
5. Izbjegavanje dugih funkcija: Svjesne važnosti čitljivosti i održavanja koda, izbjegavale smo duge funkcije. One su sklonije greškama, teško ih je razumjeti i teško je pratiti njihov tok izvršavanja. Razbijanje dugih funkcija na manje, jednostavnije funkcije poboljšava čitljivost, olakšava testiranje i omogućava bolju modularnost.

Odlučile smo pratiti pravila i postići da naš kod bude "clean code" koji olakšava buduće nadogradnje, dodavanje novih karakteristika i ispravljanje grešaka, a također omogućava da naša aplikacija bude prilagodljiva.