Univerzitet u Sarajevu

Elektrotehnički fakultet Sarajevo

Računarstvo i informatika

ITShop

Dokumentacija implementacije

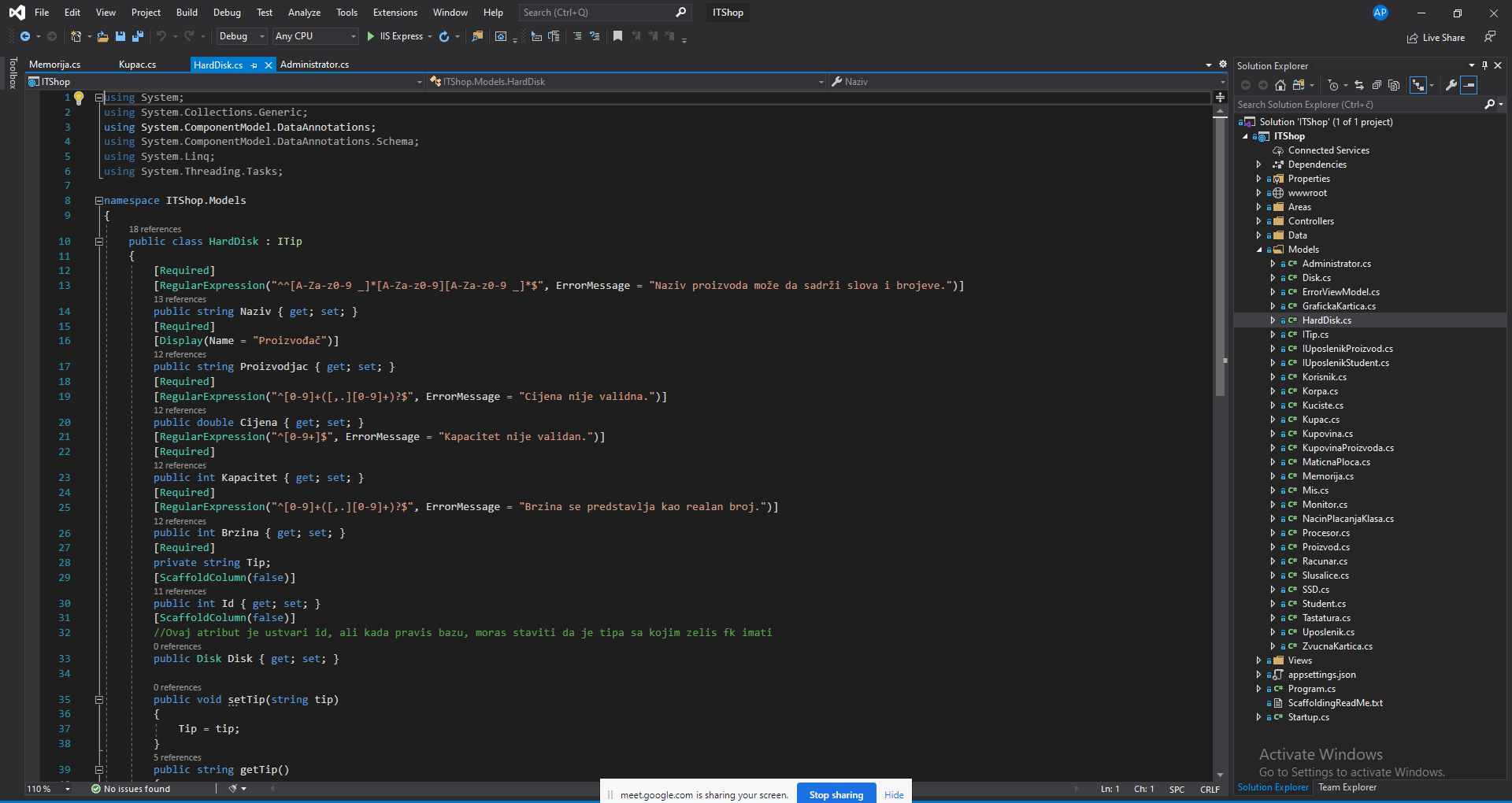
Predmet: Objektno orijentisana analiza i dizajn

Članovi grupe: Ahmed Pašić, Sabahudin Spahić, Elma Šeremet

Juni, 2020.

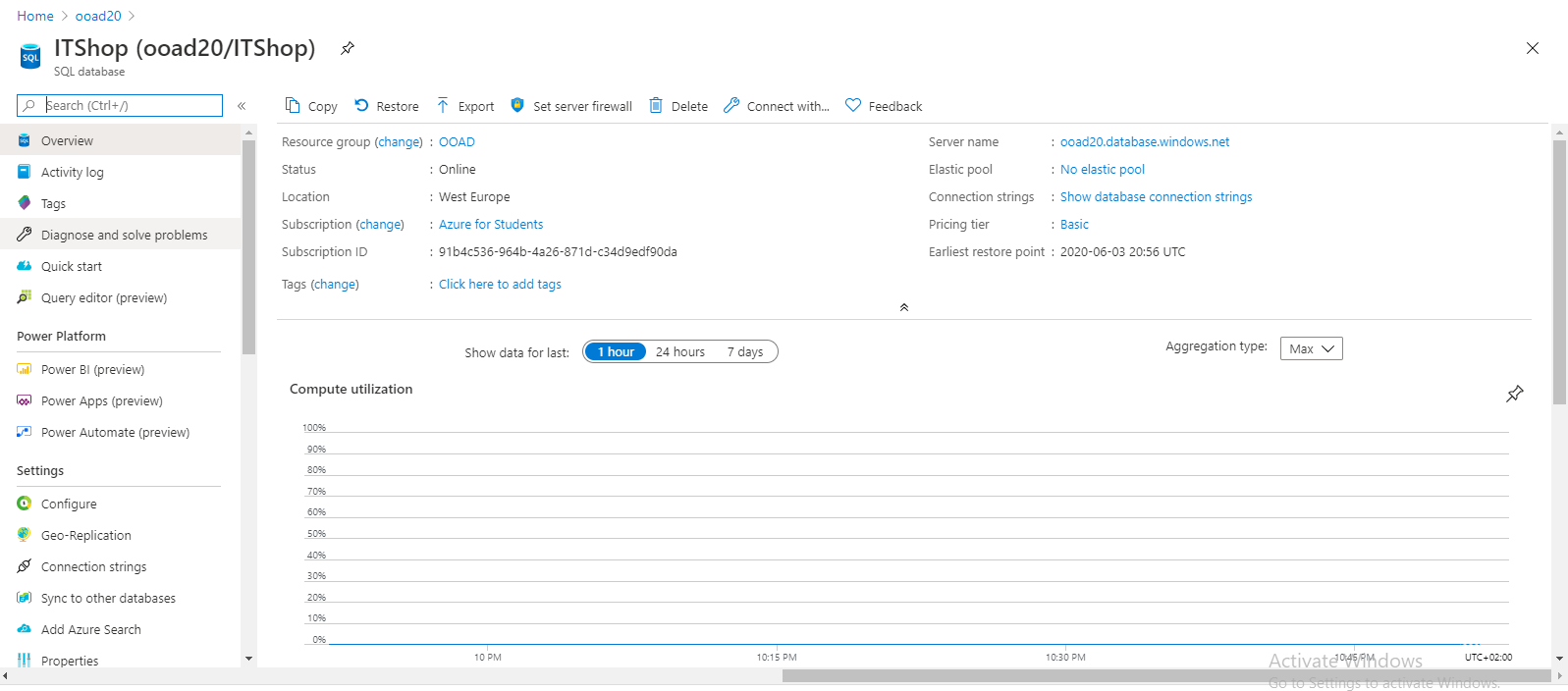
*Model*

Nakon kreiranja ASP.NET MVC projekta započeli smo sa implementacijom modela tako što smo kreirali klase u Models folderu na osnovu našeg dijagrama klasa. Nakon što smo kreirali modele, bilo je neophodno kreirati i specifičnu klasu OOADContext čija je uloga da mapira klase koje će se nalaziti u bazi podataka.



*Kreiranje baze na Azure*

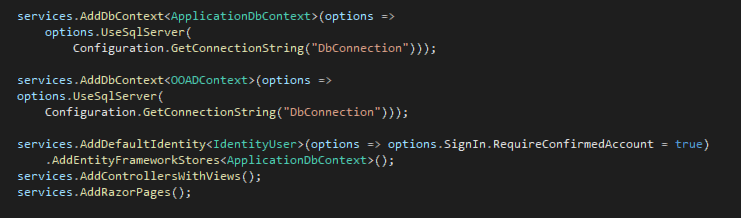
Odlučili smo se da čuvamo našu bazu podataka na platformi Azure. Prvo smo napravili server koji se zove OOAD20 i u njemu smo napravili bazu podataka ITShop.



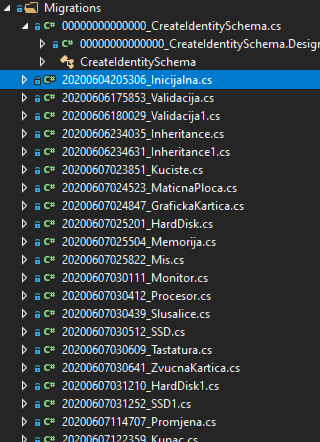
Kako bi se konektovali sa našom bazom, morali smo u datoteci appsettings.json staviti naš ConnectionStrings.



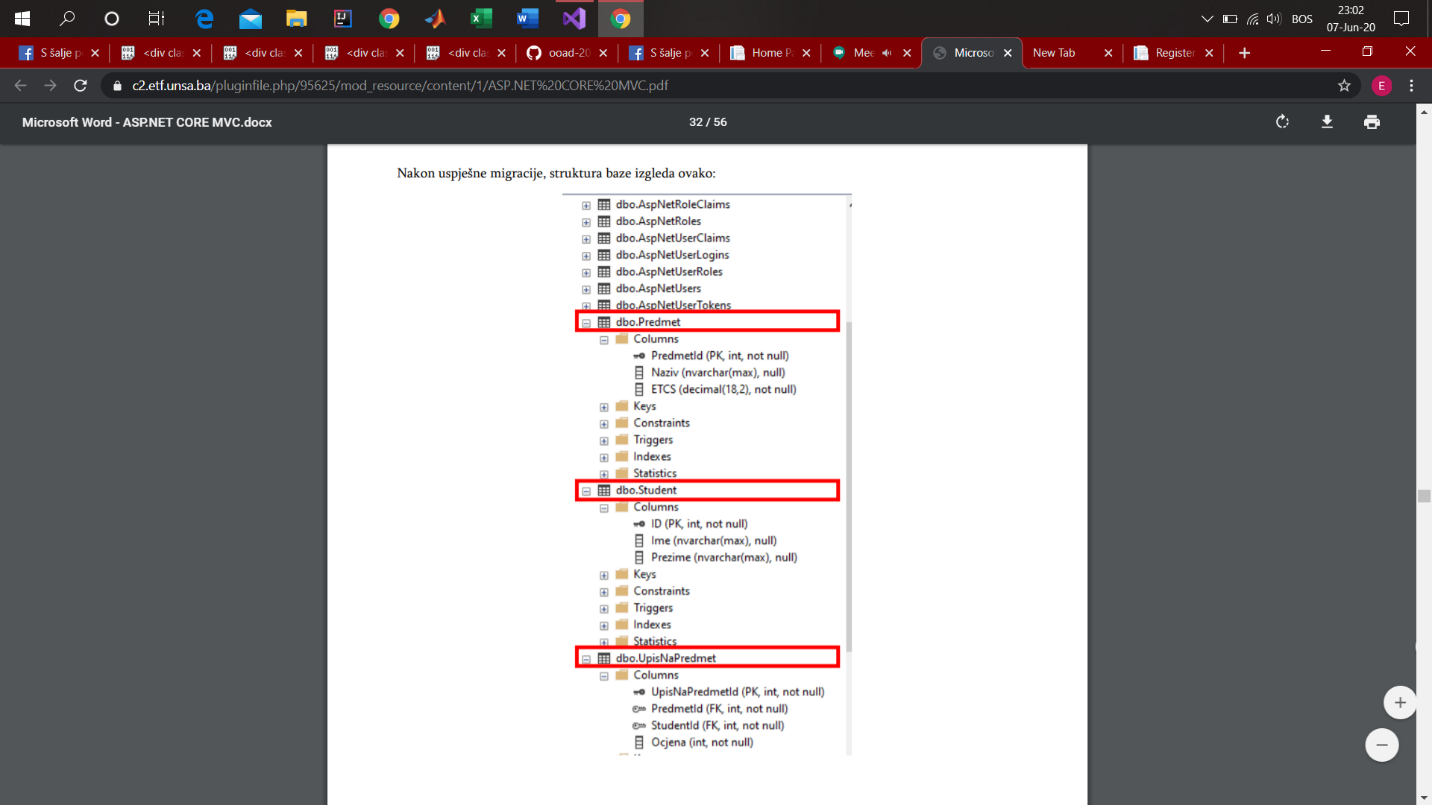
Pošto je pored našeg konteksta, aplikacija automatski generisala context koji se odnosi na Identity, odnosno na korisnike, nephodno je i u fileu Startup.cs izvršiti korekciju korištene konekcije (DefaultConnection) u „DbConnection“.



Programski jezik C# nudi uslugu migriranja što u biti znači da mi možemo dizajnirati izgled baze podataka tako što prvo napravimo sve potrebne klase u Models folderu.

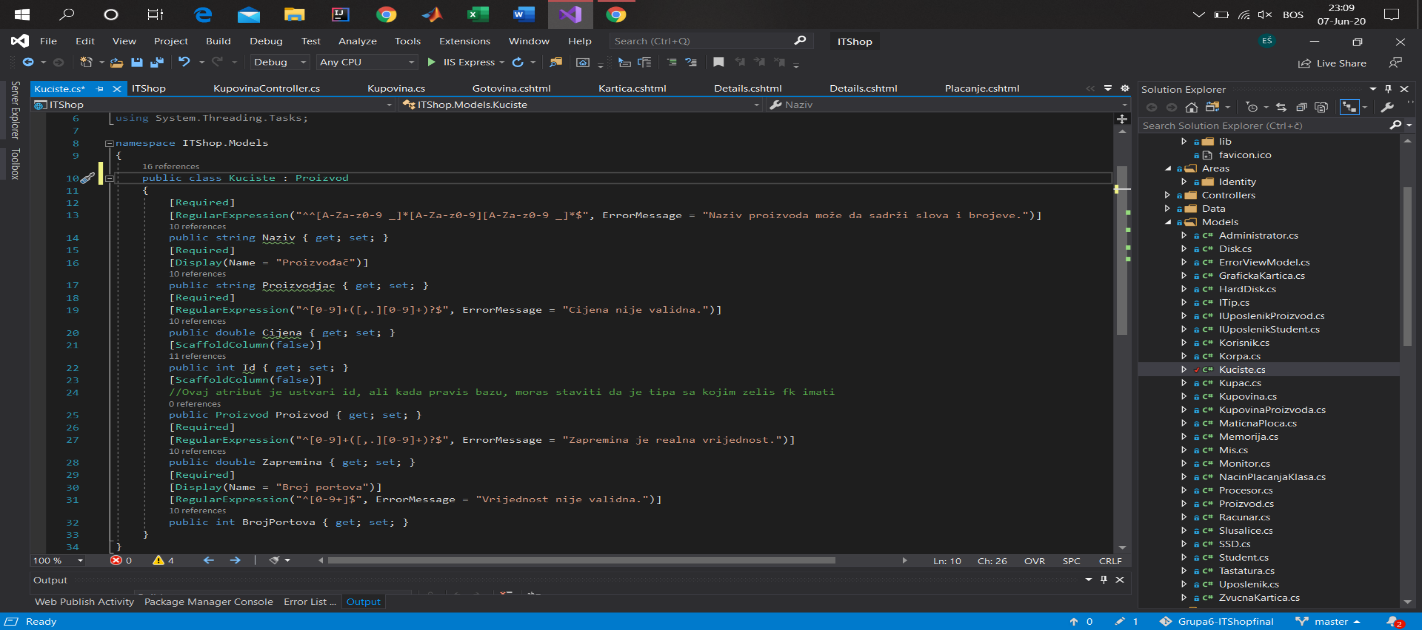


Nakon uspješne migracije, izgled naše baze podataka možemo vidjeti u ServerExplorer-u.



*Validacija modela*

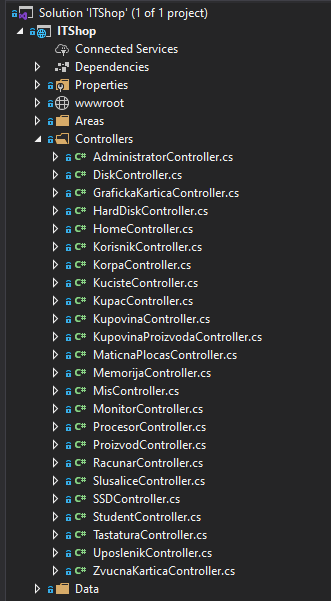
Još jedna važna osobina ASP.NET MVC i EntityFramework-a jesu dodatna definisanja samog modela. Pod tim se misli na dodatno definisanje osobina atributa kao što su: obavezno polje, display name, dužina, format, itd. Nakon kreiranja baze to je bio naš sljedeći korak. Neke od primjera validacije su prikazane na sljedećoj slici:



Nakon što smo završili sa izmjenama na modelu, vršimo update baze podataka, i to tako što dodajemo novu migraciju i vršimo update baze u konzoli.

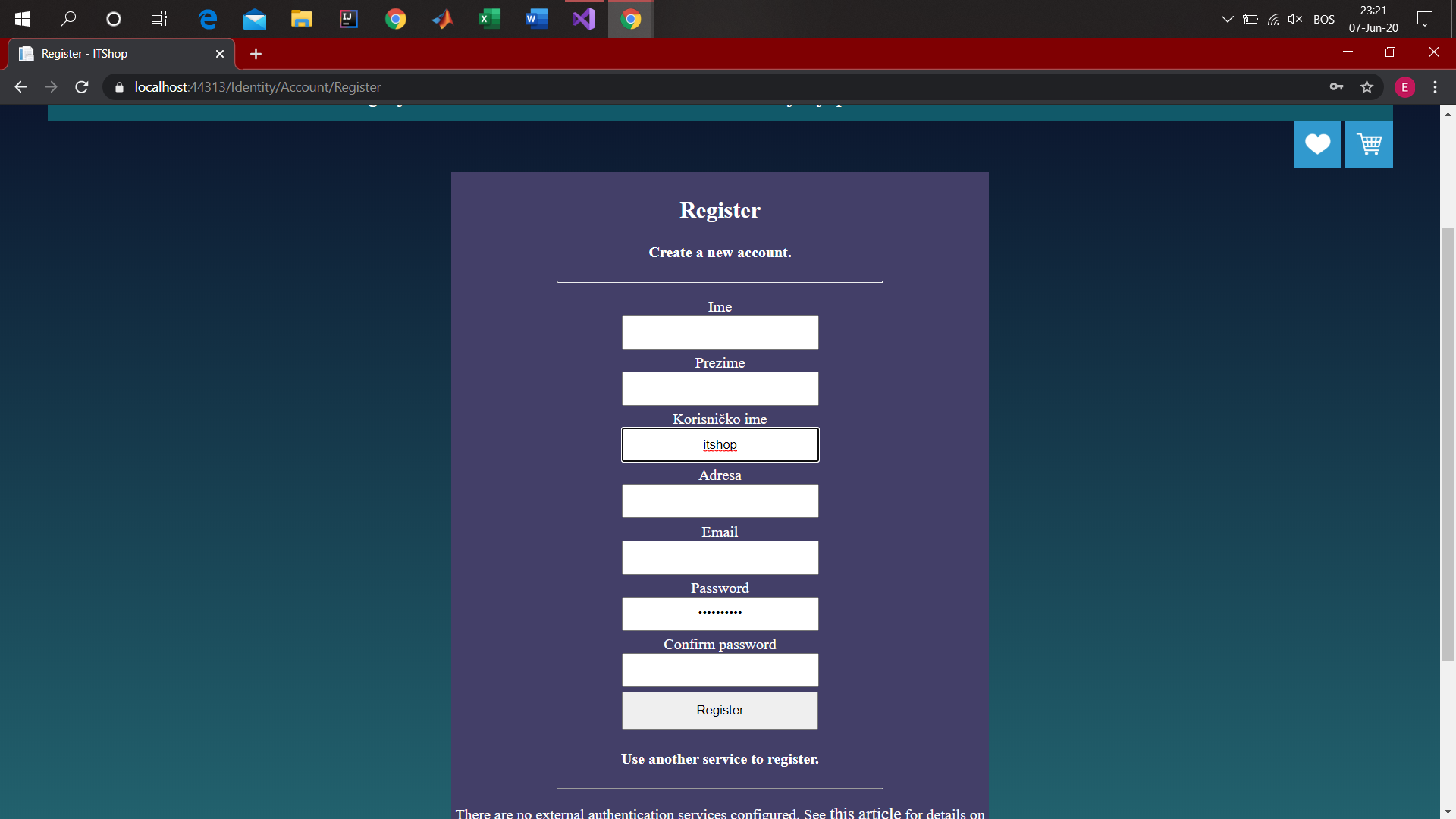
*Kreiranje kontrolera i pogleda*

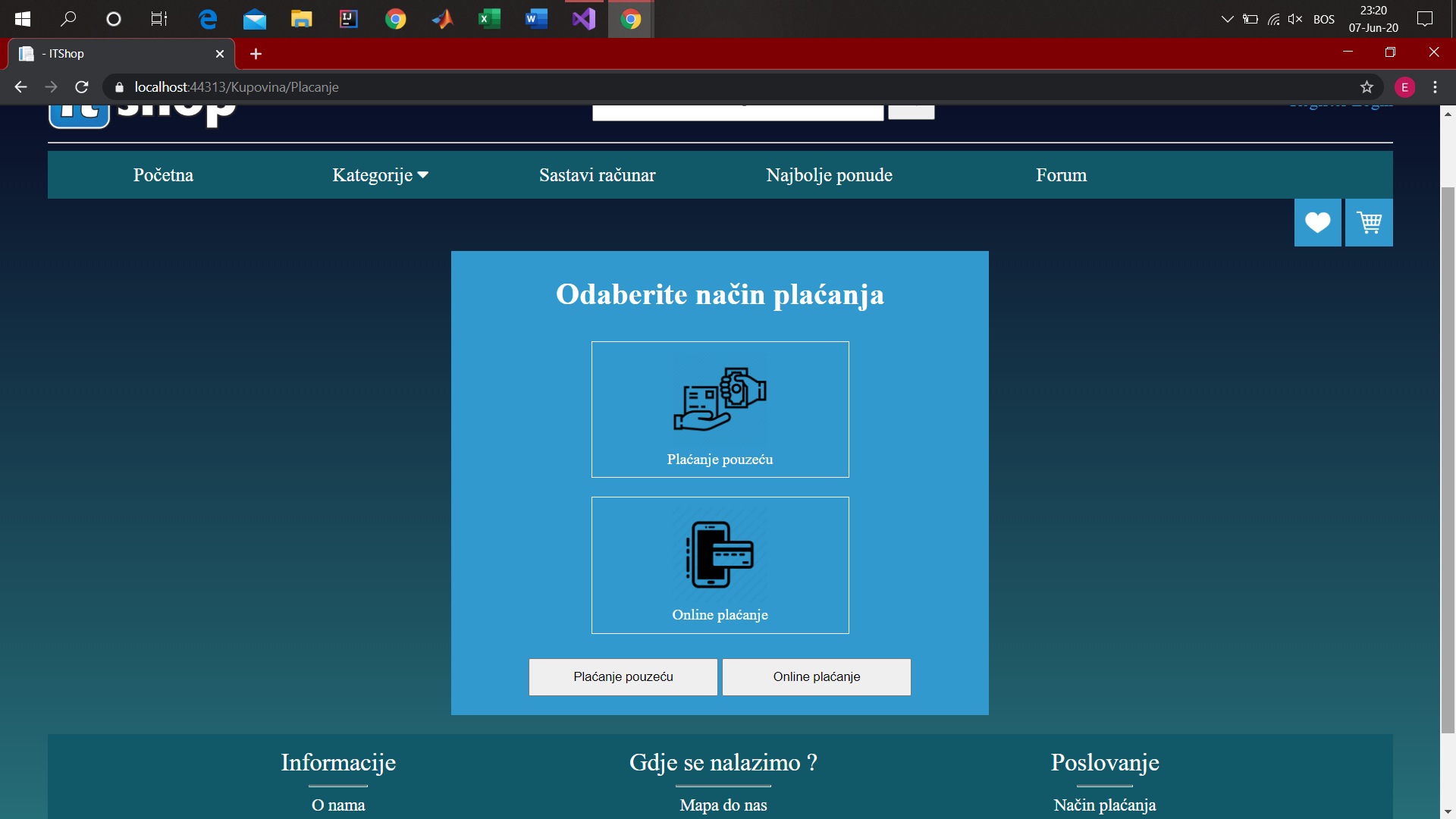
Nakon što smo kreirali i validirali model, sada možemo iskoristiti jednu od najvećih prednosti ASP.NET MVC– Scaffolding. Kao što je već navedeno, to podrazumijeva generisanje koda na osnovu modela.

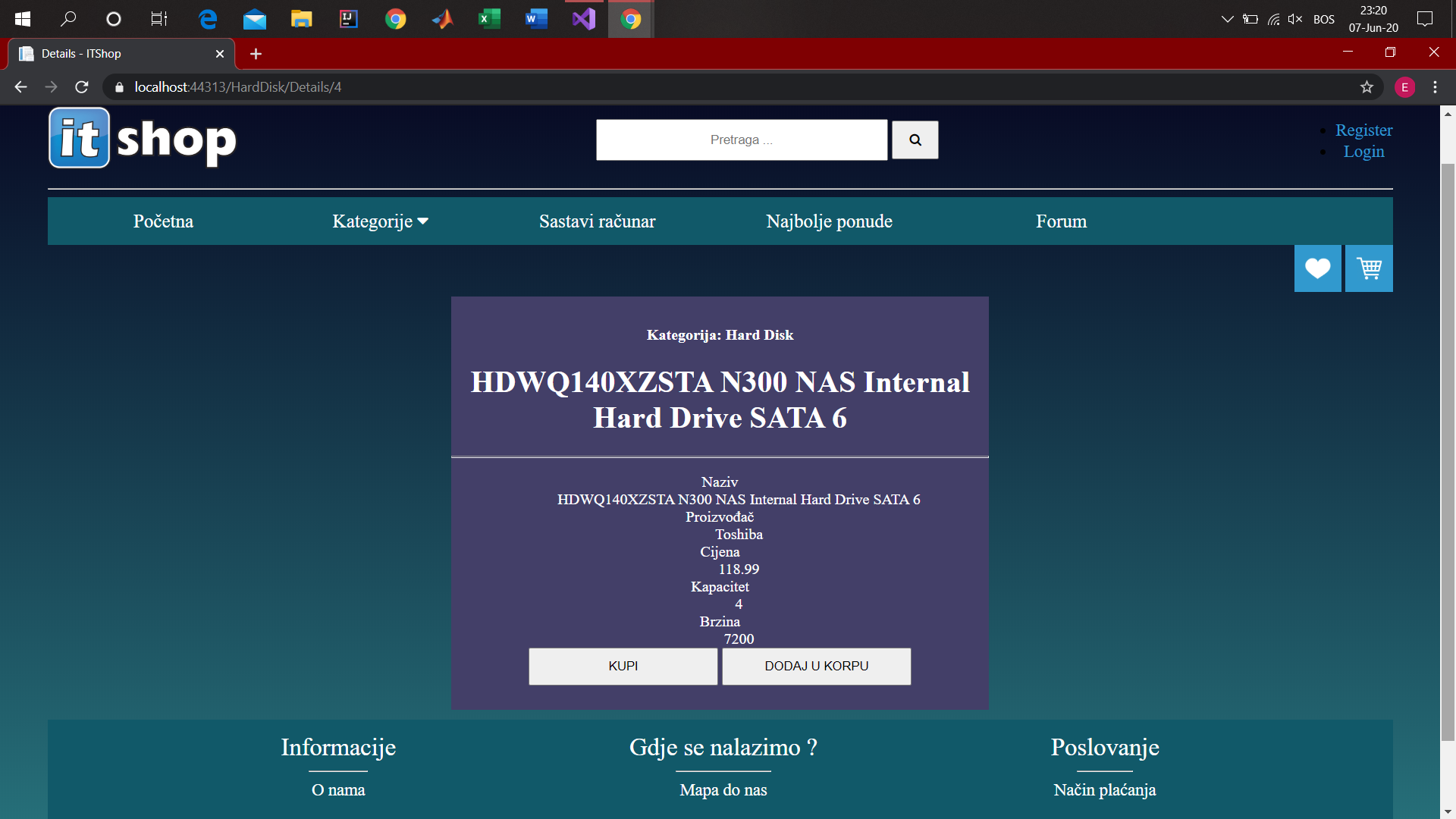


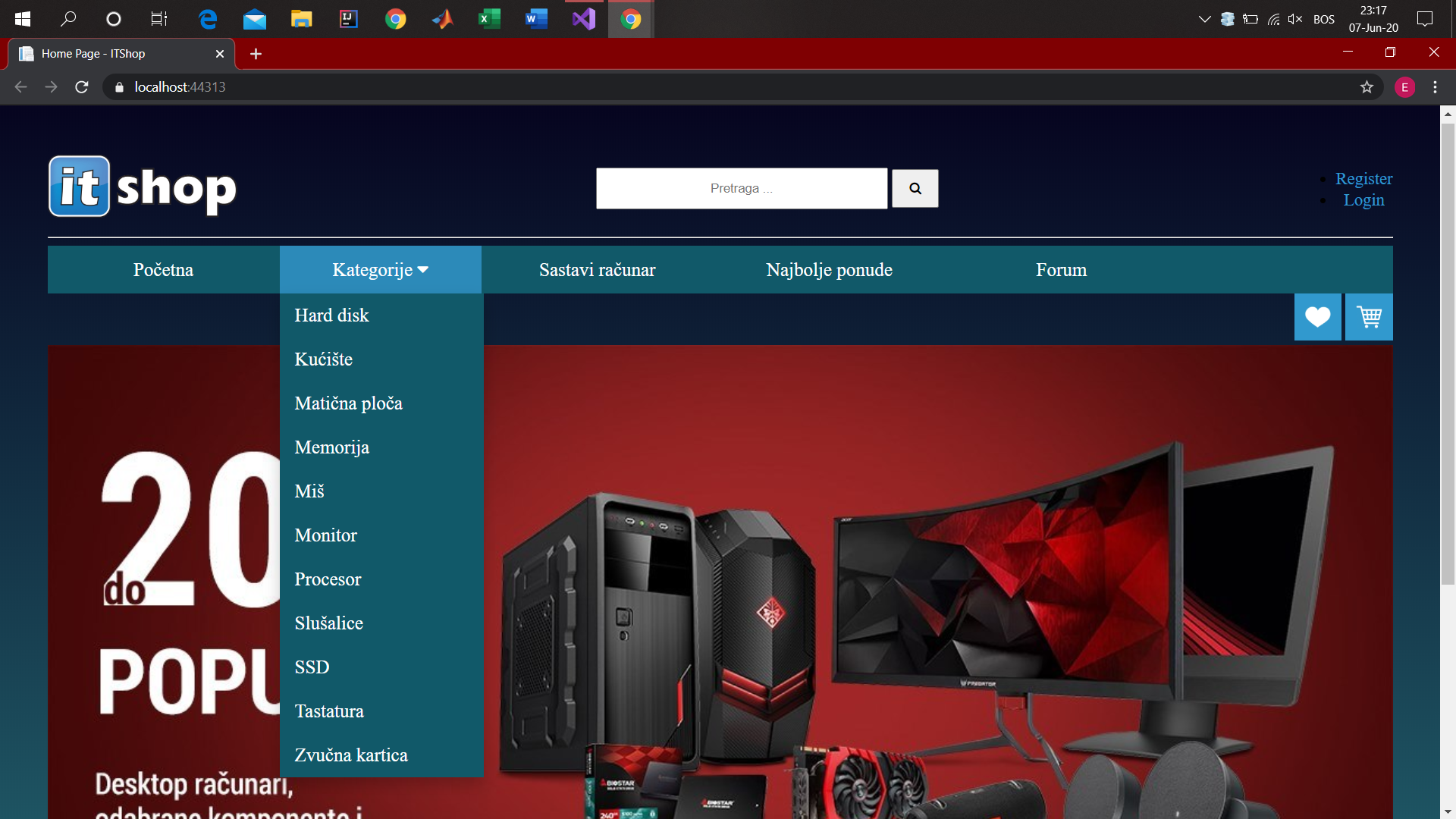
*Kontroleri i pogledi*

Prilikom kreiranja kontrolera na osnovu modela automatski se generišu pogledi, ali i pripadajuće metode u kontroleru.

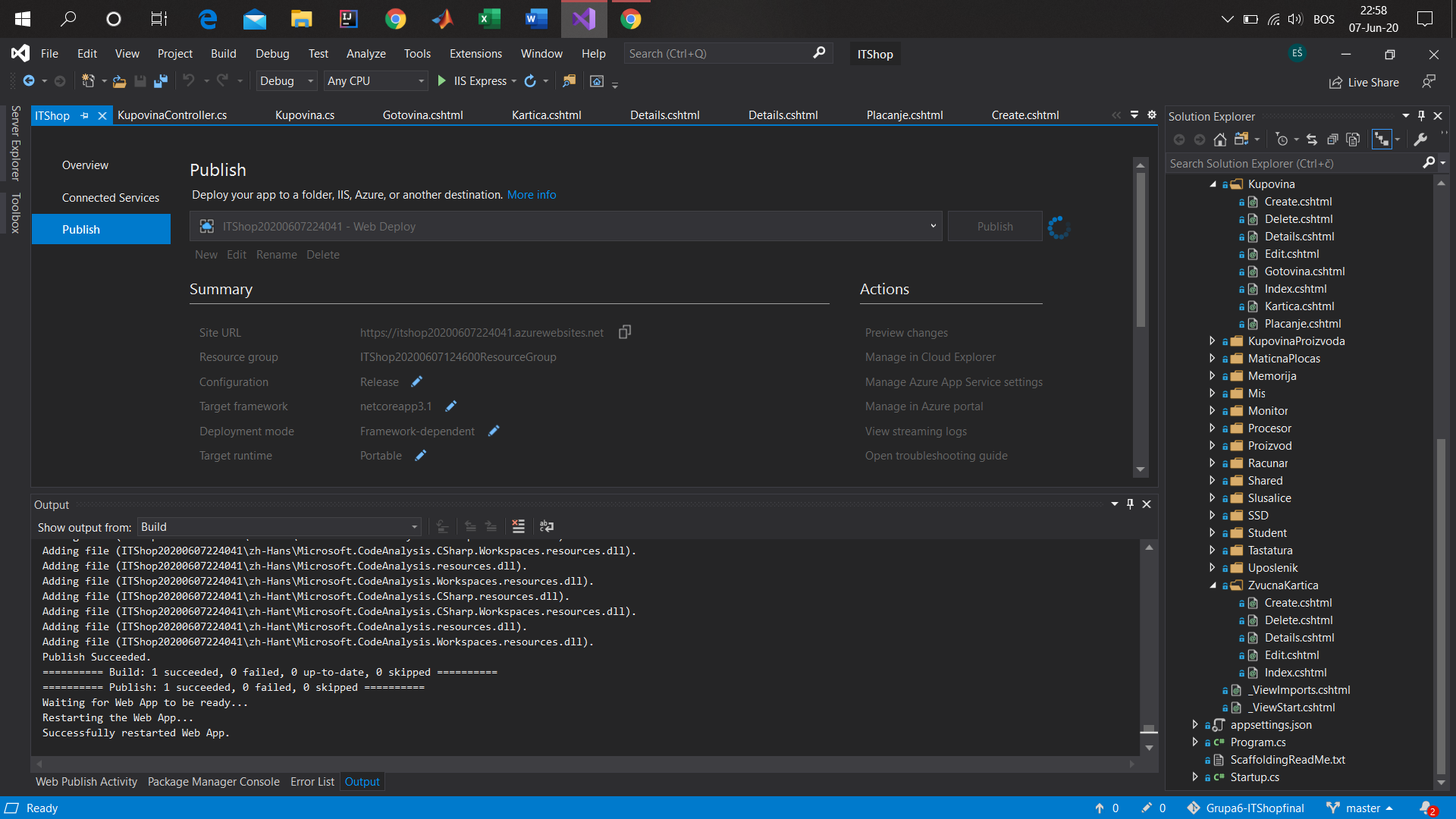


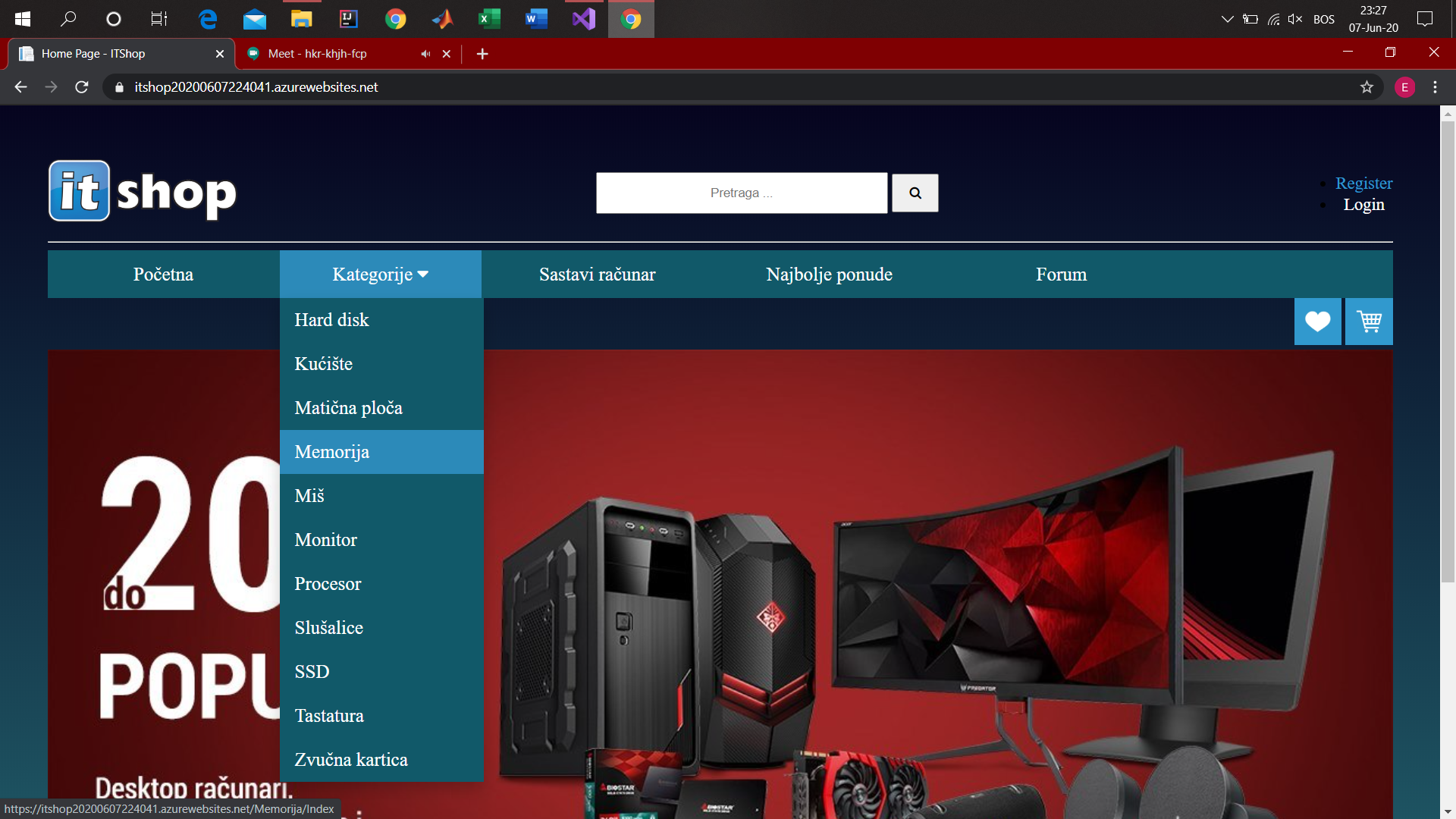






*Deployment*





U videu na našem repozitoriju možemo vidjeti funkcionalnosti koje smo implementirali.