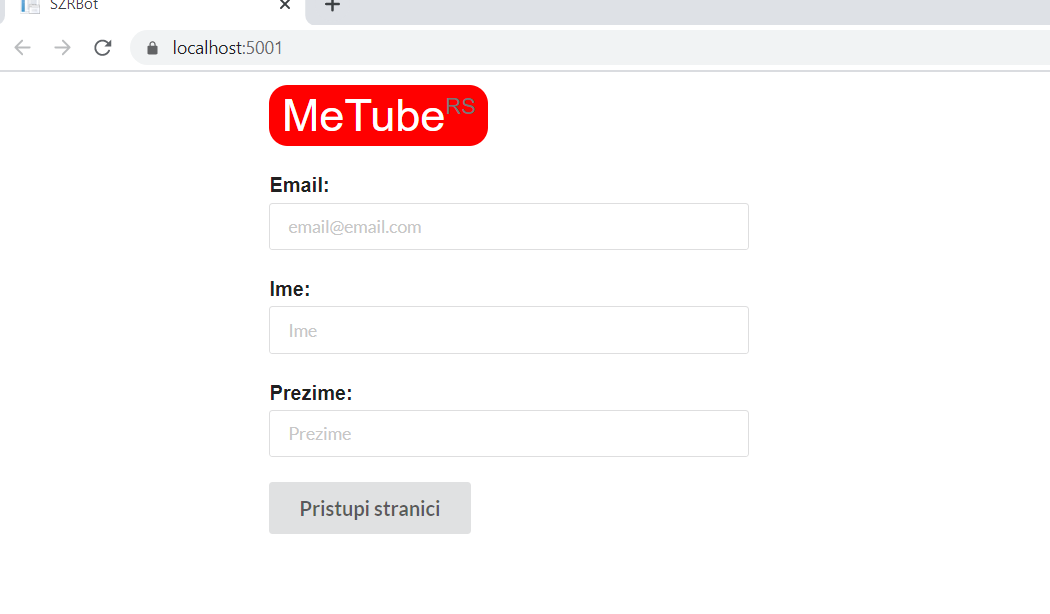
**SEMINARSKI RAD - PROGRAMIRANJE ZA INTERNET**

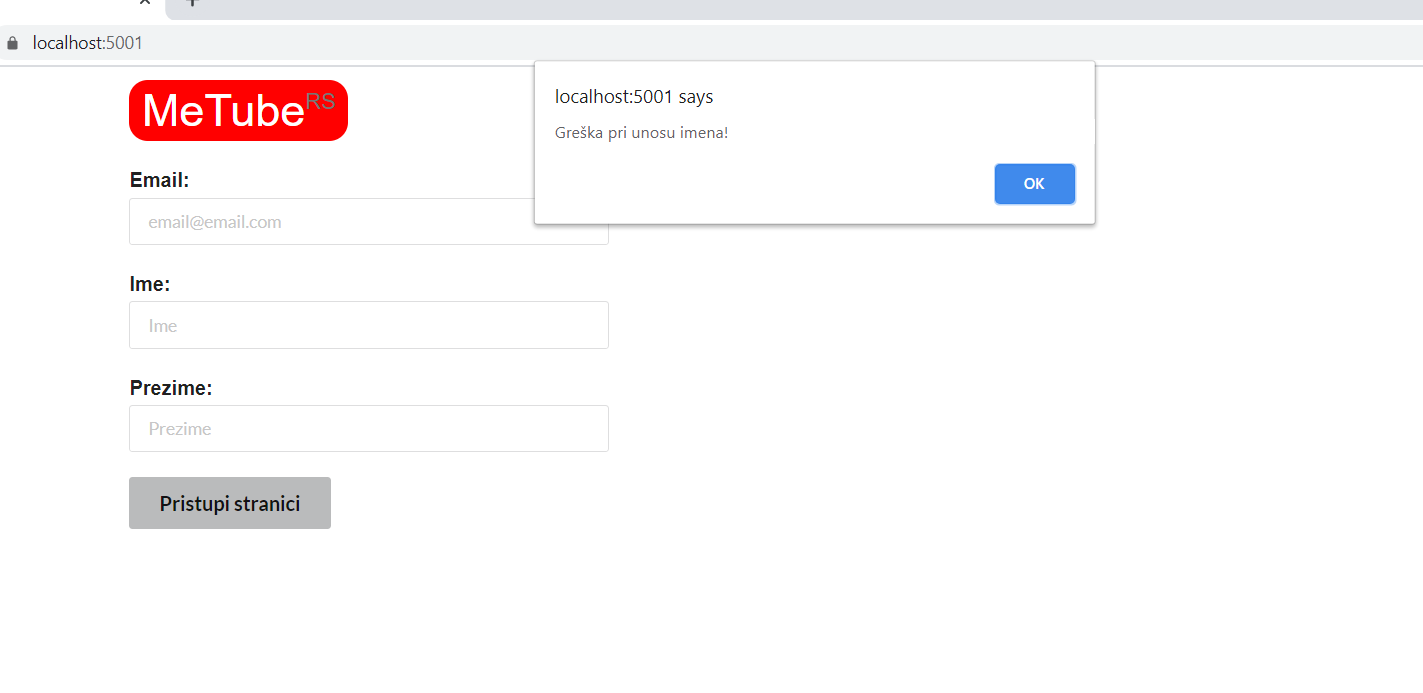
**Opis aplikacije**

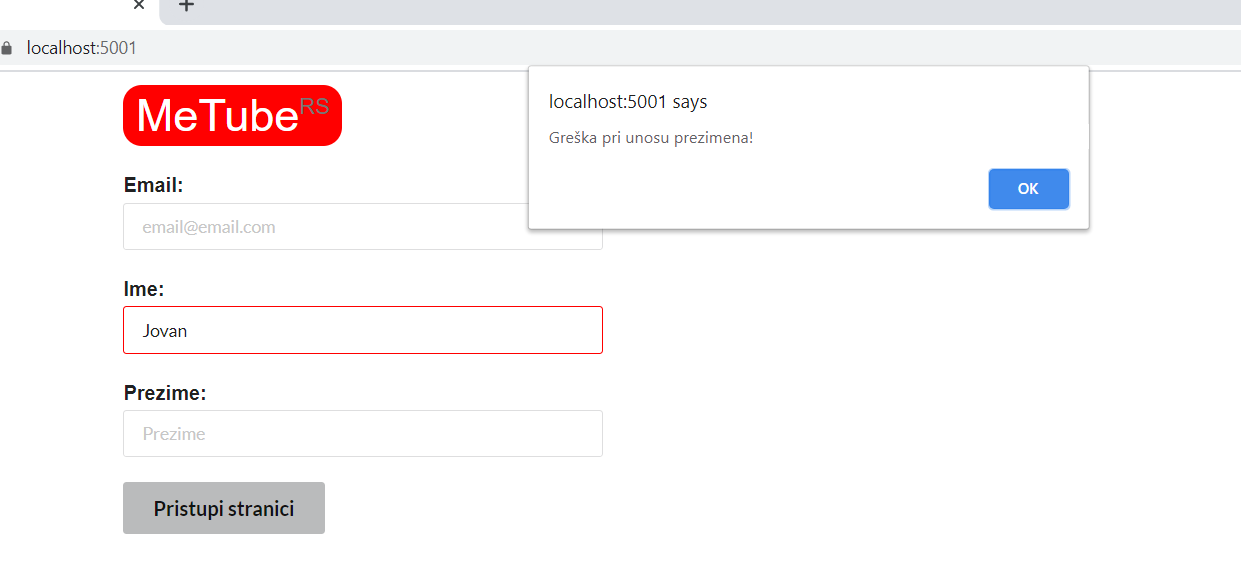
Ovaj seminarski rad treba da predstavlja sajt za pregled video snimak sa YouTube socijalne mreže. Pored toga, u pozadini ova aplikacija će čekati na okidač - pretraženi tekst koji u sebi ima reč koja ima veze sa fiktivnim likom iz serije "Bela lađa" Srećkom Šojićem. Kada se ta akcija desi, aplikacija će aktivirati metodu za slanje mejla (pozivnica za učlanjenje u fiktivnu političku stranku SZR - Stranka Zdravog Razuma) korisniku. Pre slanja mejla će se izvršiti upis podataka o korisniku u bazu kako bi se pratilo kome je poslat već mejl da ne bi došlo do ponavljanja.

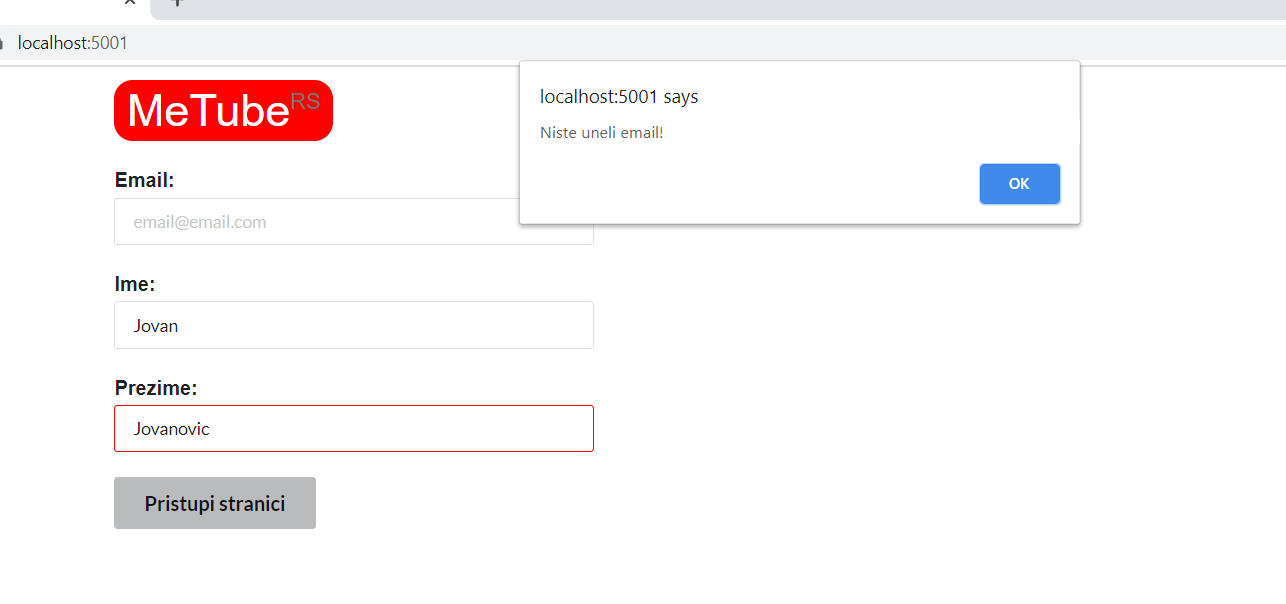
Kada se pokrene aplikacija, pojaviće se forma za unos podataka. Tu je potrebno uneti email (ukoliko korisnik sa tom adresom već postoji, ostala polja će se sama popuniti), ime i prezime:

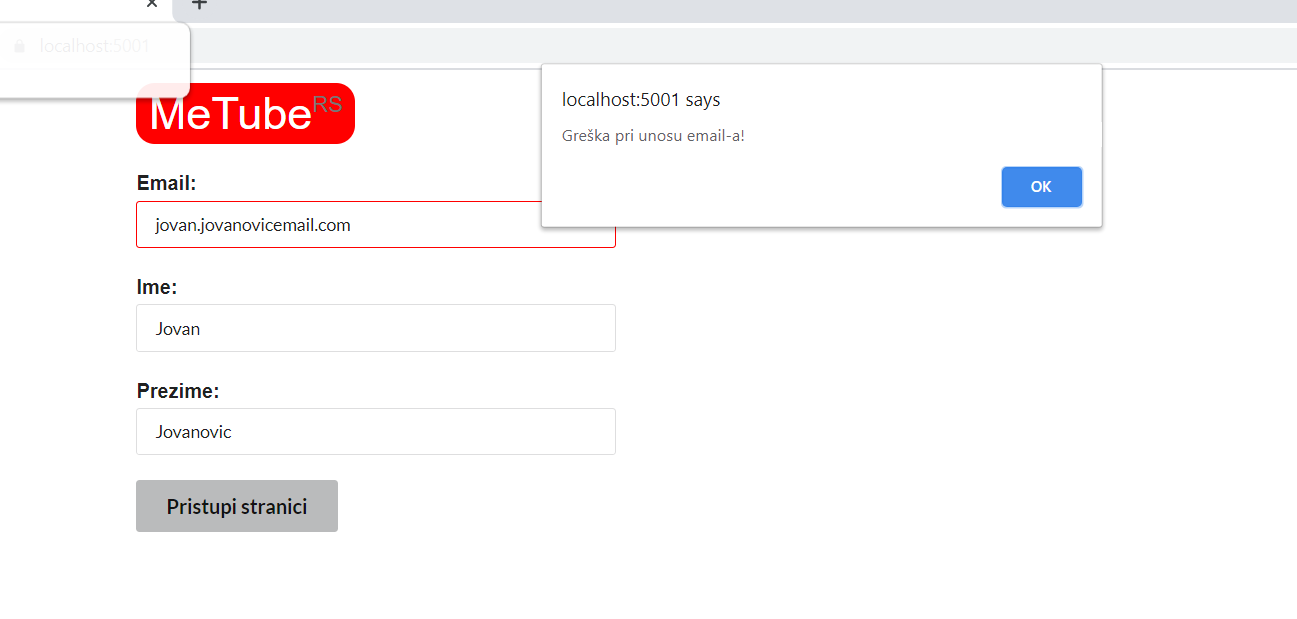


Ukoliko ne unesemo ime, pojaviće se prozor koji nas o tome obaveštava, a ista je situacija i za prezime i email, kao i email koji nije ispravan:

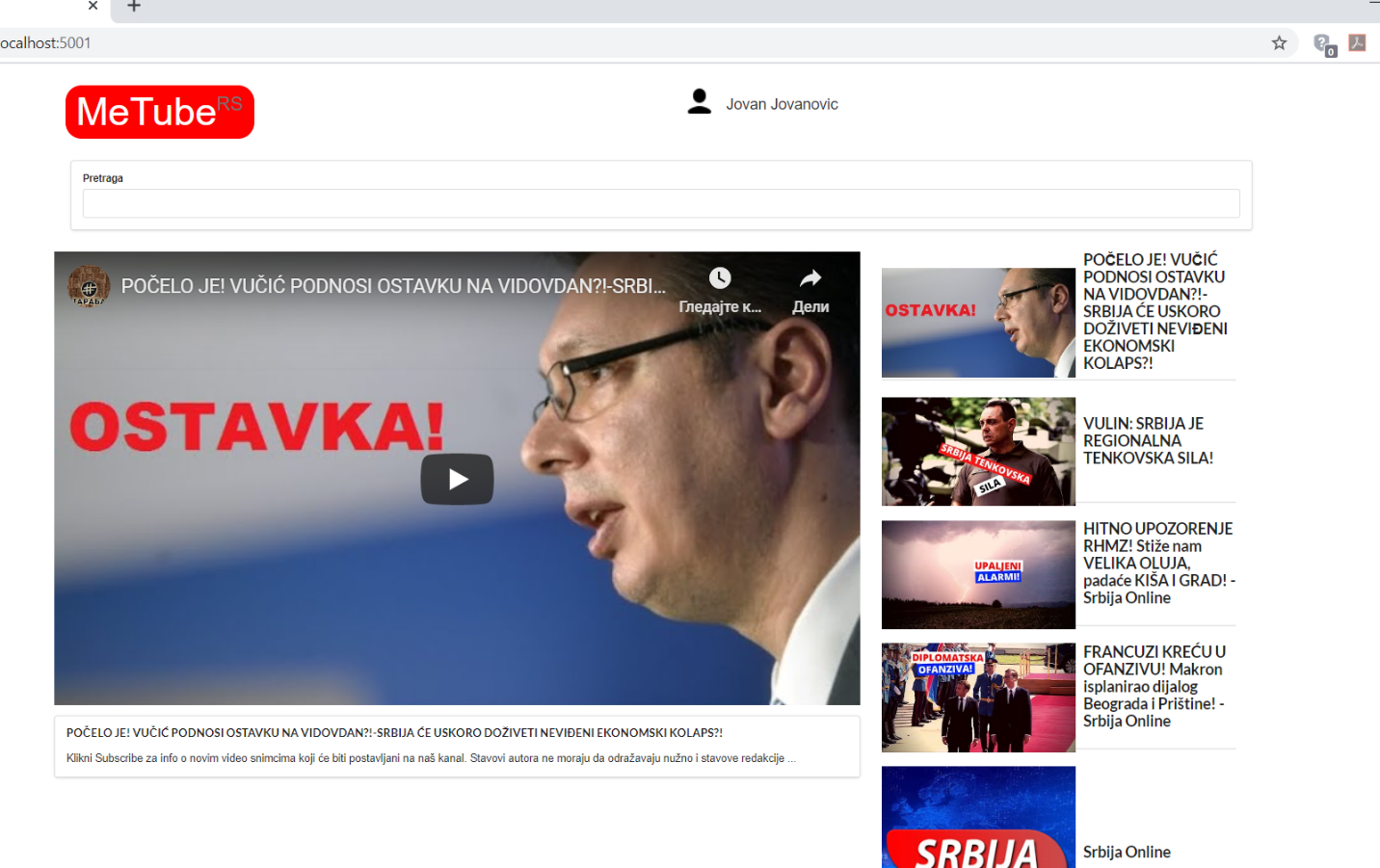








Nakon unošenja podataka, pojaviće se stranica sa video snimcima, slična YouTube stranici:



Aplikacija sve video snimke povlači direktno sa YT sajta preko API-a koji omogućava Google. Početni "search" termin je "Srbija" i zbog toga možemo na početku videti ponuđene video snimke vezane sa Srbiju.

**Izrada aplikacije kroz korake**

Za izradu aplikacije koristićemo okruženje Visual Studio 2019. Ova aplikacija je tipa ASP.NET Core i pisaćemo je u C# programskom jeziku. Koristićemo i framework-ove i biblioteke poput: React.js (za dizajn), Redux (za kreiranje akcija koje aktiviraju metode kontrolera), Entity Framework (za kreiranje i upotrebu baze podataka), Axios (za asinhrone zahteve Youtube-u) i Semantic-UI (za stilove).

Prvih 5 koraka su vezani za pripremu za rad.

1. Pokrenuti VS2019 i kreirati novi projekat. Izabrati ASP.NET Core Web Application. Zatim izabrati React.js and Redux kao template i označiti ASP.NET Core 2.1 verziju.

2. Kada je projekat kreiran, potrebno je instalirati dodatne pakete. Pomoću NuGet PM ćemo instalirati:

1) Microsoft.EntityFrameworkCore (2.1.1);

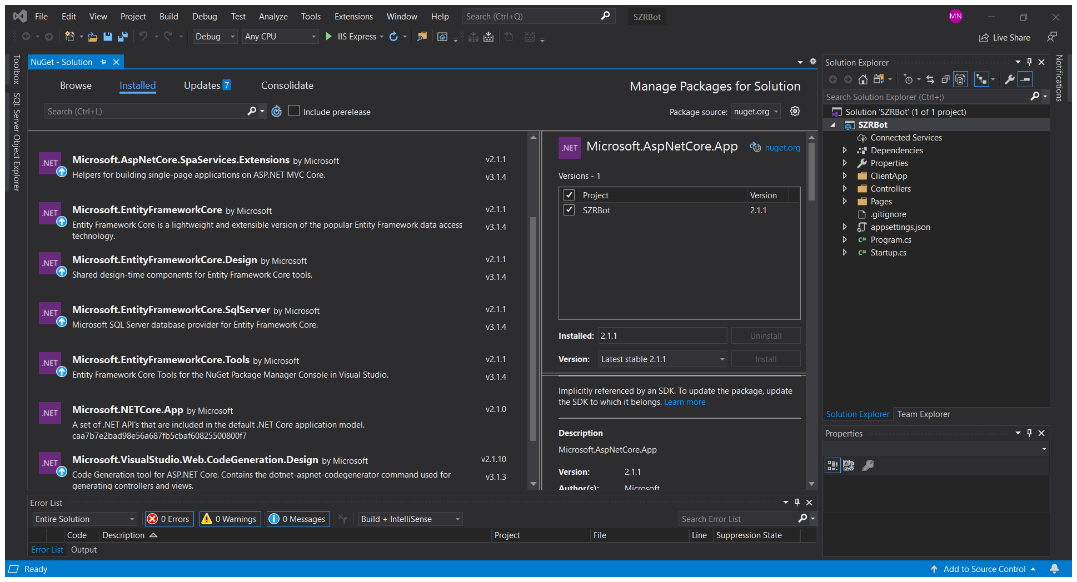
2) Microsoft.EntityFrameworkCore.SqlServer (2.1.1);

3) Microsoft.EntityFrameworkCore.Tools (2.1.1);

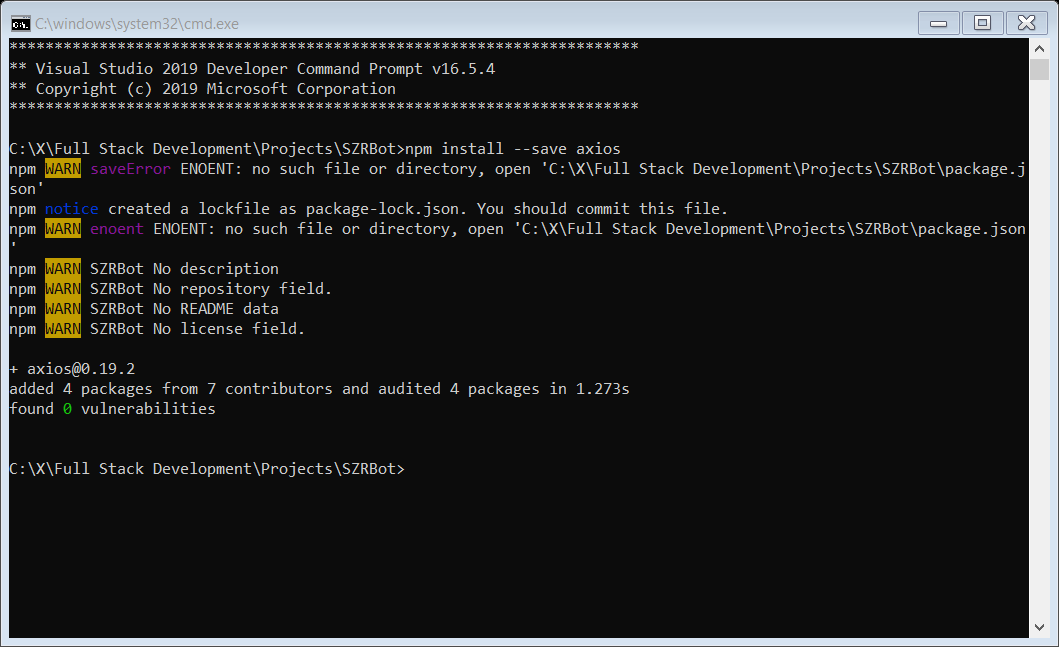
4) Microsoft.EntityFrameworkCore.Design (2.1.1);

5) Microsoft.VisualStudio.Web.CodeGeneration.Design (2.1.10).

Nakon toga naši paketi će izgledati ovako:



3. Nakon dodavanja paketa pomoću NuGet-a, potrebno je otvoriti Command Line -> Developer Command Prompt, i uneti komandu "npm install --save axios" i pritisnuti enter. Na ovaj način smo instalirali axios biblioteku u projekat i trebalo bi da se pojavi ovakav prozor:

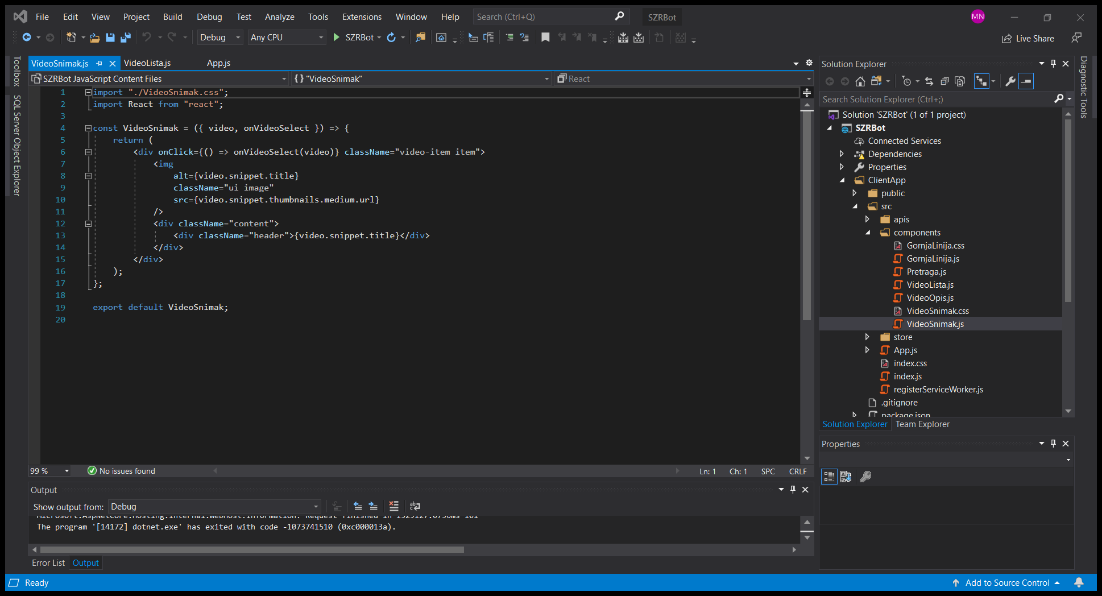


4. U Solution Explorer prozoru otvoriti ClientApp -> src -> components i izbrisati sve fajlove unutar njega jer su to nepotrebni fajlovi kreirani po default-u. Takođe, u folderu ClientApp -> src -> store izbrisati fajlove Counter.js i WeatherForecast.js. Unutar foldera Controllers potrebno je izbrisati fajl koji je tu: Sample.

5. Još jedna pripremna stavka je povezivanje aplikacije sa semantic-ui sajtom kako bi bilo omogućeno korišćenje stilskih klasa. Potrebno je dodati <link> tag u index.html fajl: <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/semantic.min.css" />.

Nakon ovih 5 koraka, sledi 7 (6-12) koji će biti posvećeni komponentama i dizajnu aplikacije kao i učitavanju video snimaka sa YT platforme.

6. Nakon završene pripreme, prvi korak će biti dodavanja u folder components, fajlova vezanih za svaki video snimak pojedinačno, VideoSnimak.js (komponenta) i VideoSnimak.css (stil vezan za VideoSnimak komponentu):

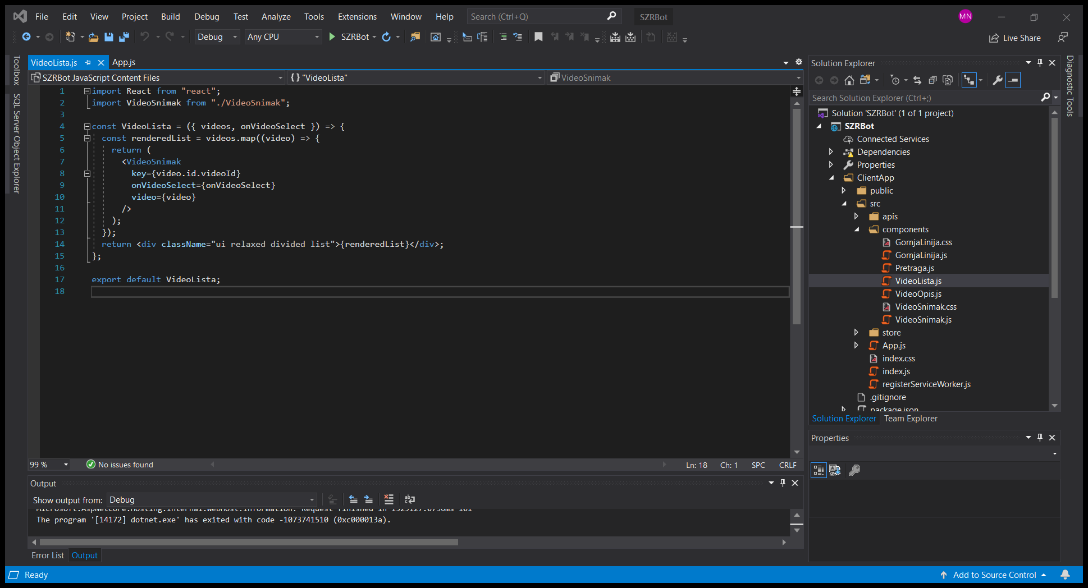


Što se tiče VideoSnimak.css, kod je vrlo kratak:

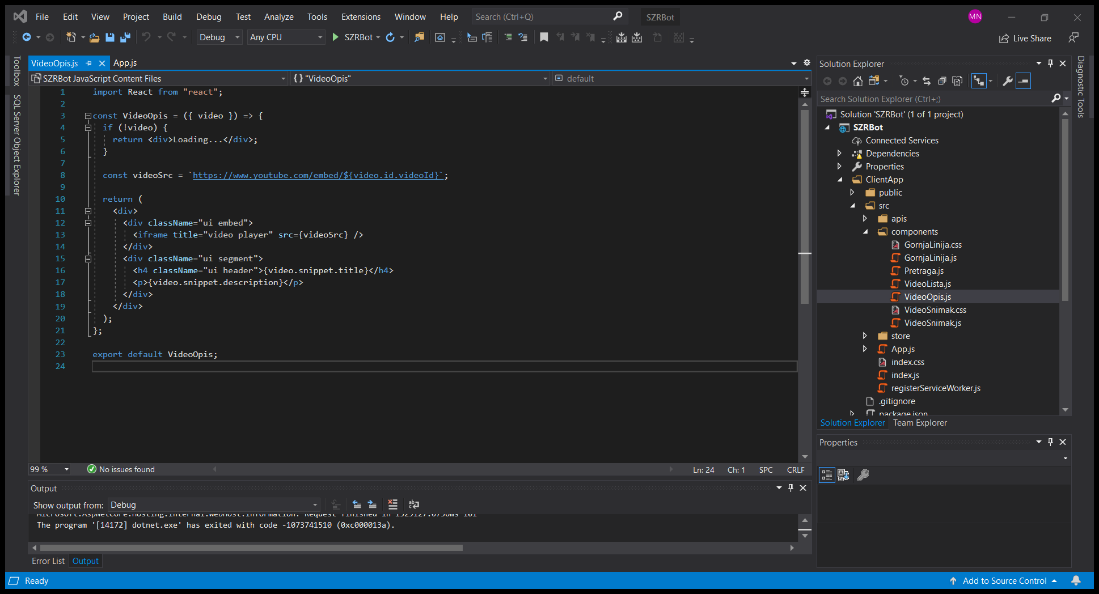
.video-item { display: flex !important; align-items: center !important; cursor: pointer; }

.video-item.item img { max-width: 180px; }

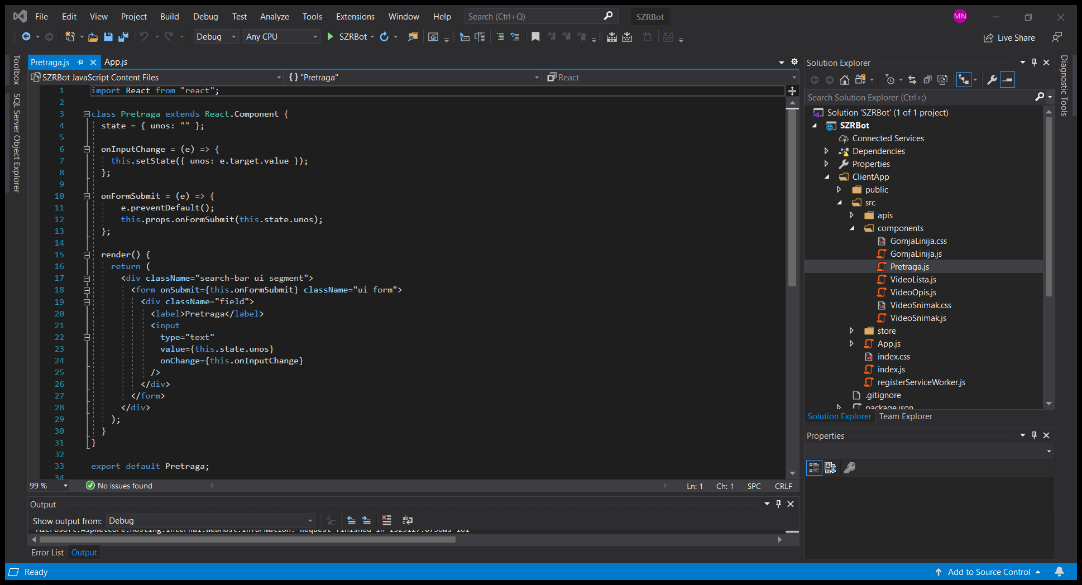
7. Sledeća komponenta je vezana za listu video snimaka (VideoSnimak komponenti). Kreirati fajl u components folderu pod imenom VideoLista.js i uneti kod kao sa slike:



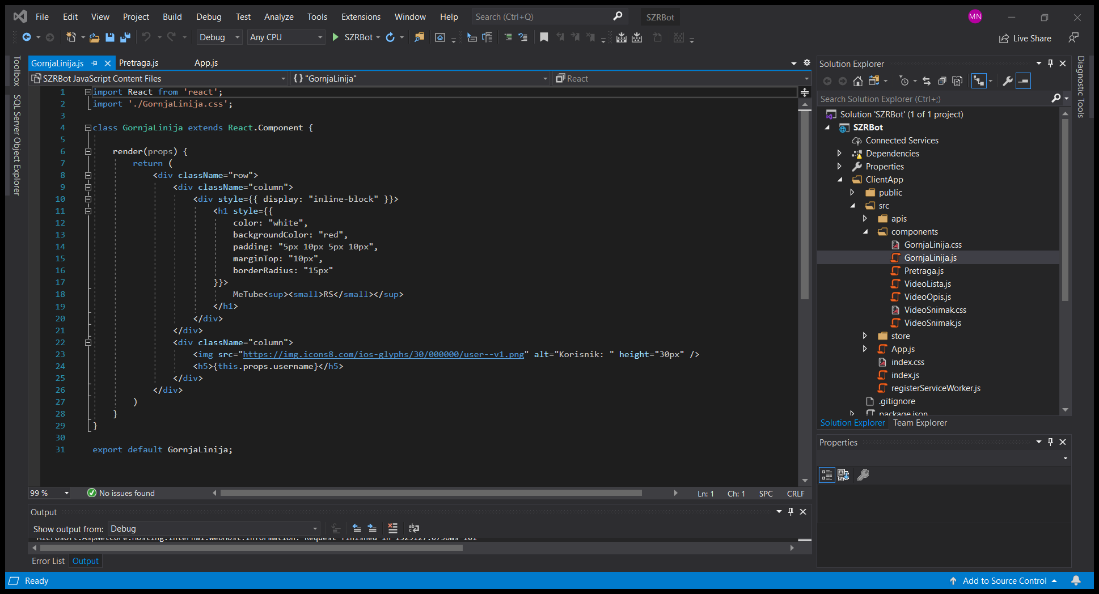
8. Sledeća na redu je komponenta VideoOpis koja izgleda ovako:

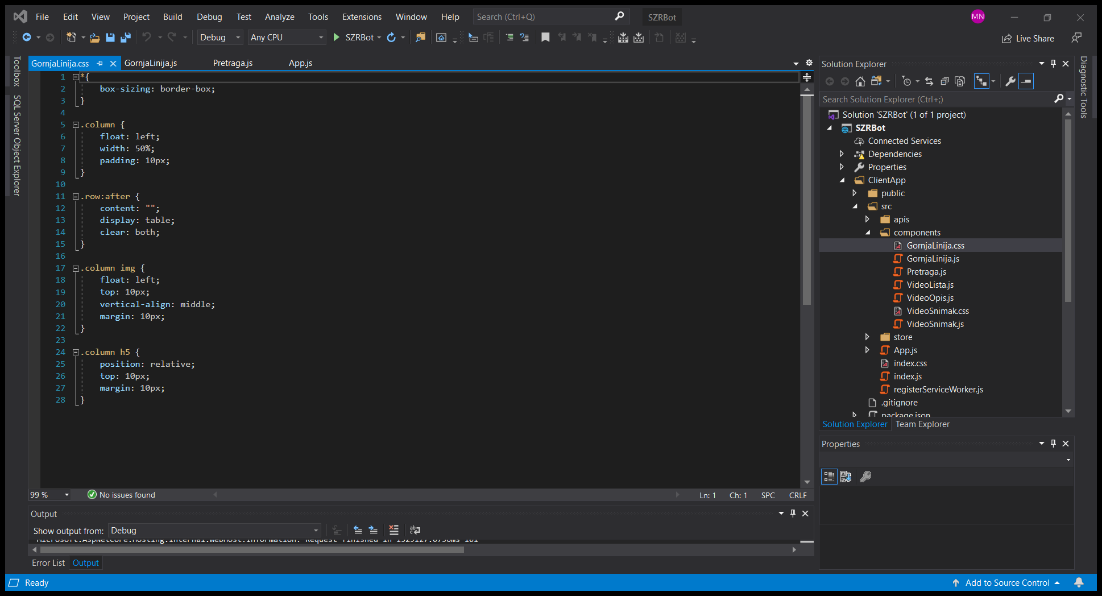


9. Komponenta koju sledeću treba kreirati vezana je za liniju za pretragu odnosno searchbar i zove se Pretraga:

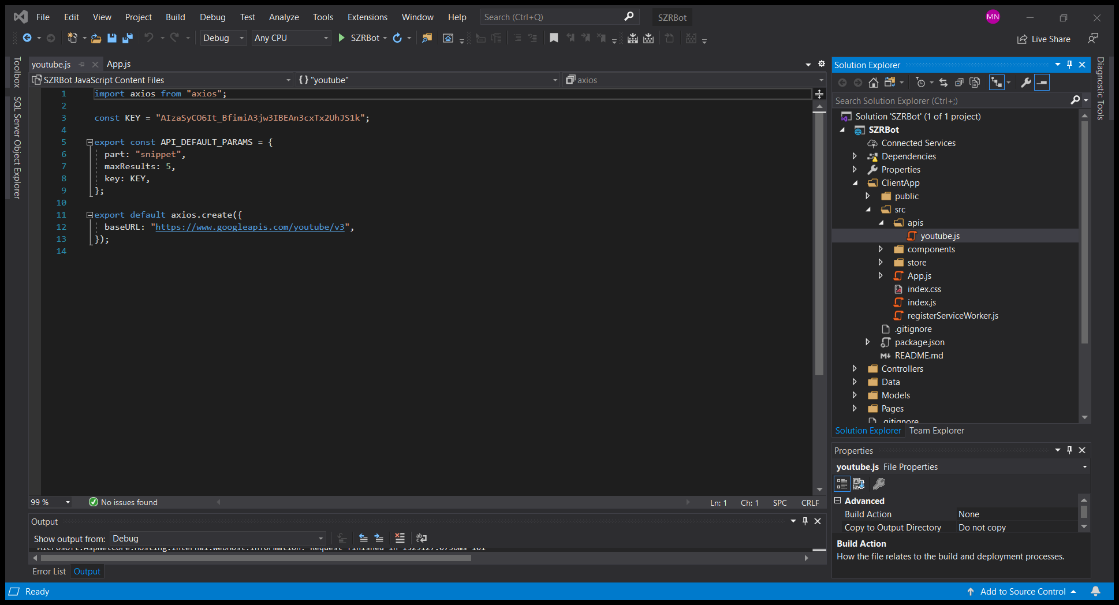


10. Još je potrebno kreirati komponentu koja se odnosi na celu gornju traku. To će biti TopLine komponenta koja će imati svoj poseban stilski fajl TopLine.css i oba će biti smeštena u components folderu:

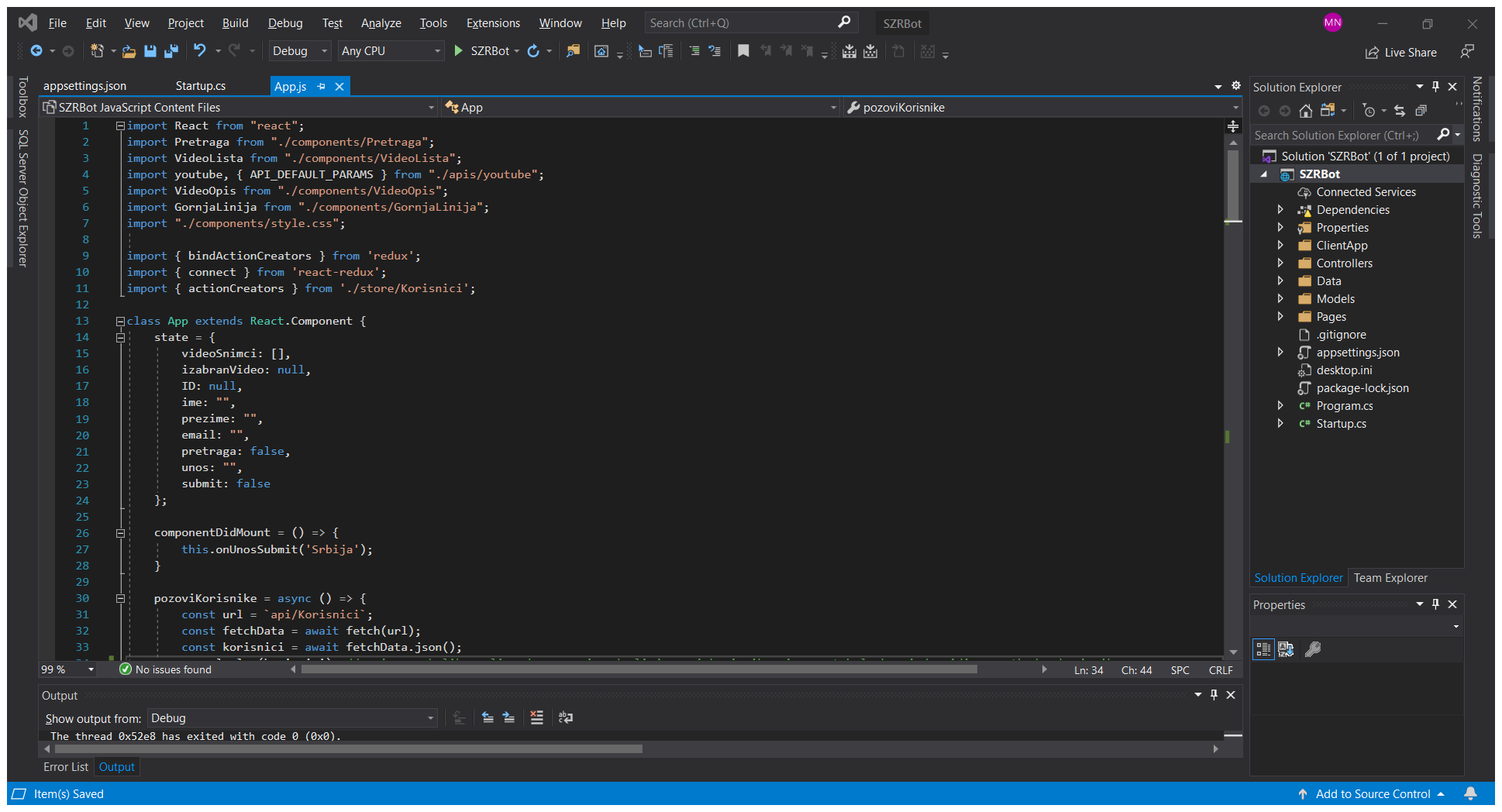


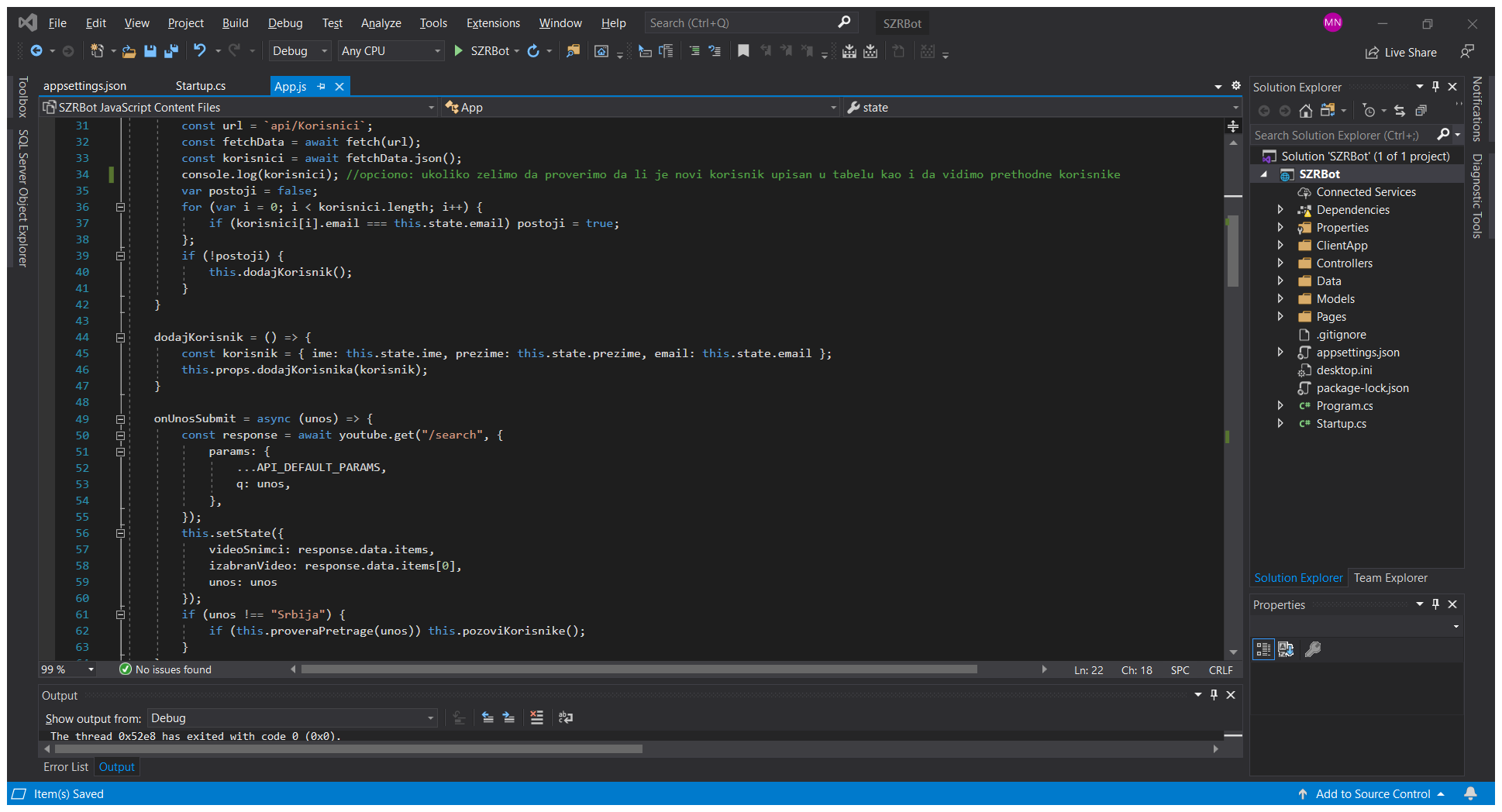


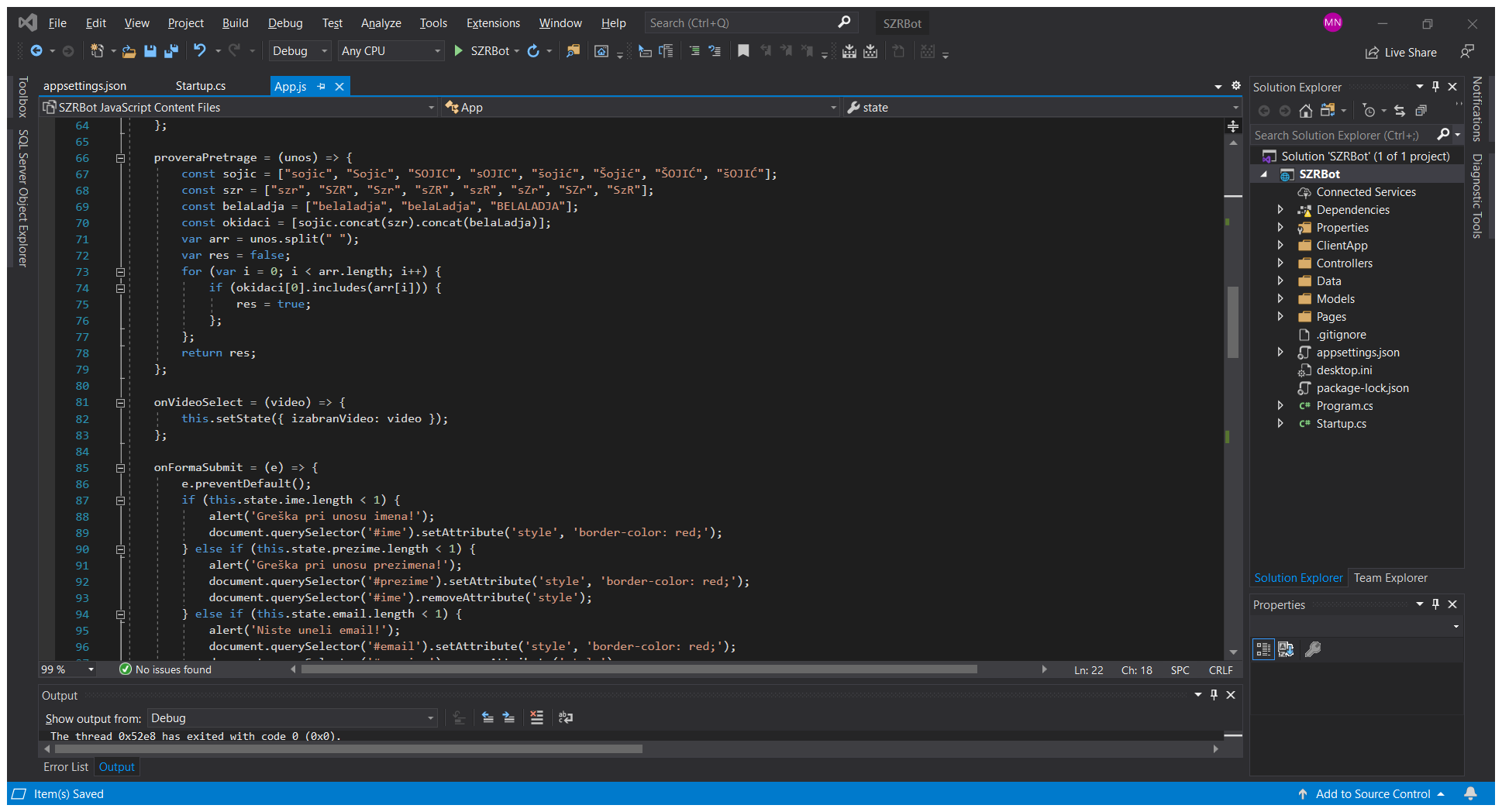
11. Pre nego što uredimo App.js fajl, potrebno je u folderu src kreirati novi folder pod imenom apis u koji ćemo smestiti API za YouTube. Naime, potrebno je imati Google nalog i u pretraživaču pronaći sajt console.developers.google.com (Google API console sajt). Tu je potrebno kreirati novi projekat i nakon toga pritisnuti ENABLE APIS AND SERVICES i pronaći API pod imenom YouTube Data API v3. Nakon toga potrebno je kreirati kredencijale u sekciji Credentials (odgovoriti na pitanja i setapovati API). Ključ koji se izgeneriše potrebno je u fajlu youtube.js (u folderu apis) smestiti u posebnu konstantu pod imenom KEY. Taj svojevrsni ključ će biti jedina razlika u odnosu na dati string sa sledeće slike. Sve ostalo što je na slici će ostati isto (ključ je podešen na public tako da se može koristiti sa različitih računara ukoliko je neophodno):

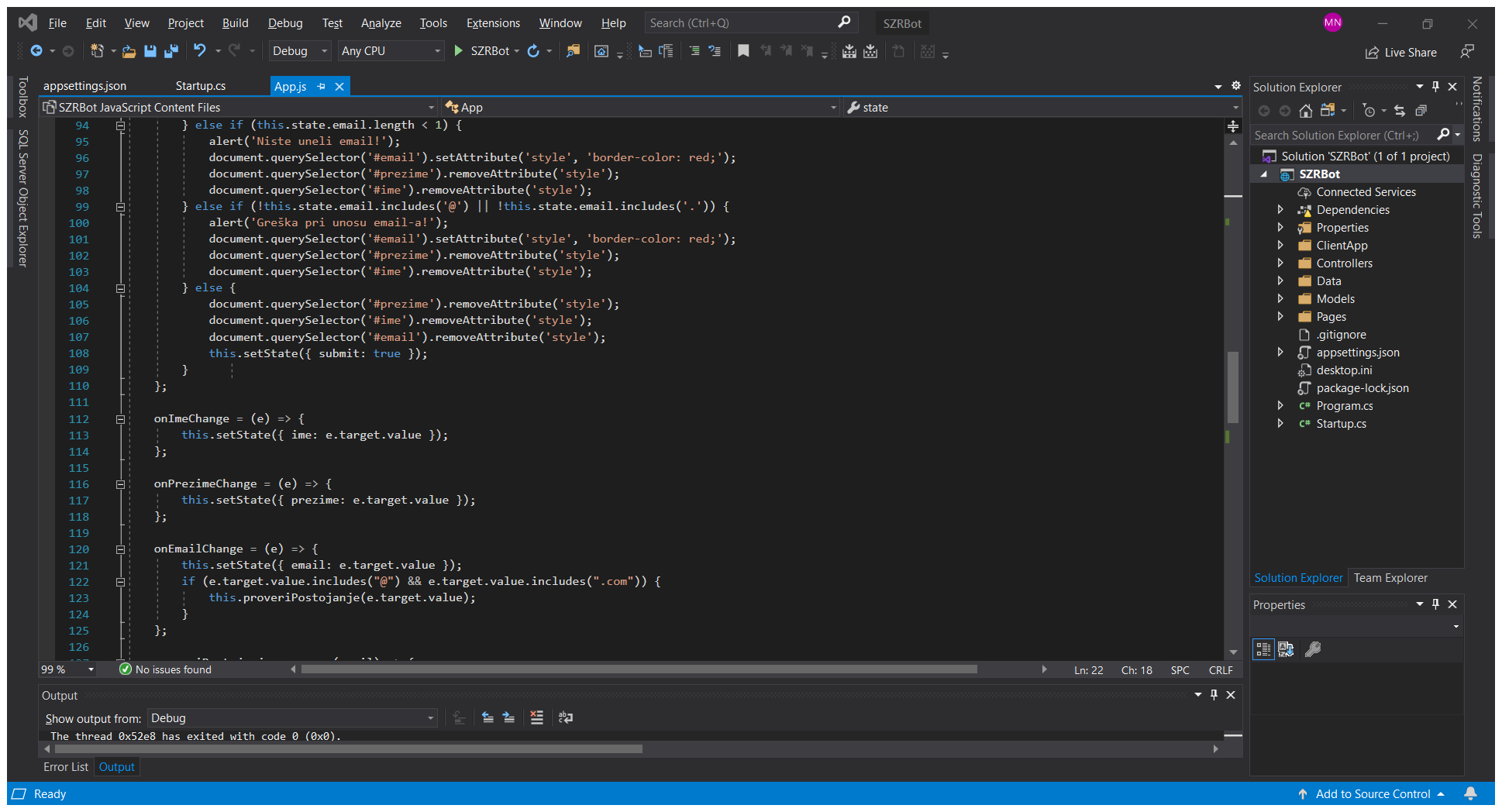


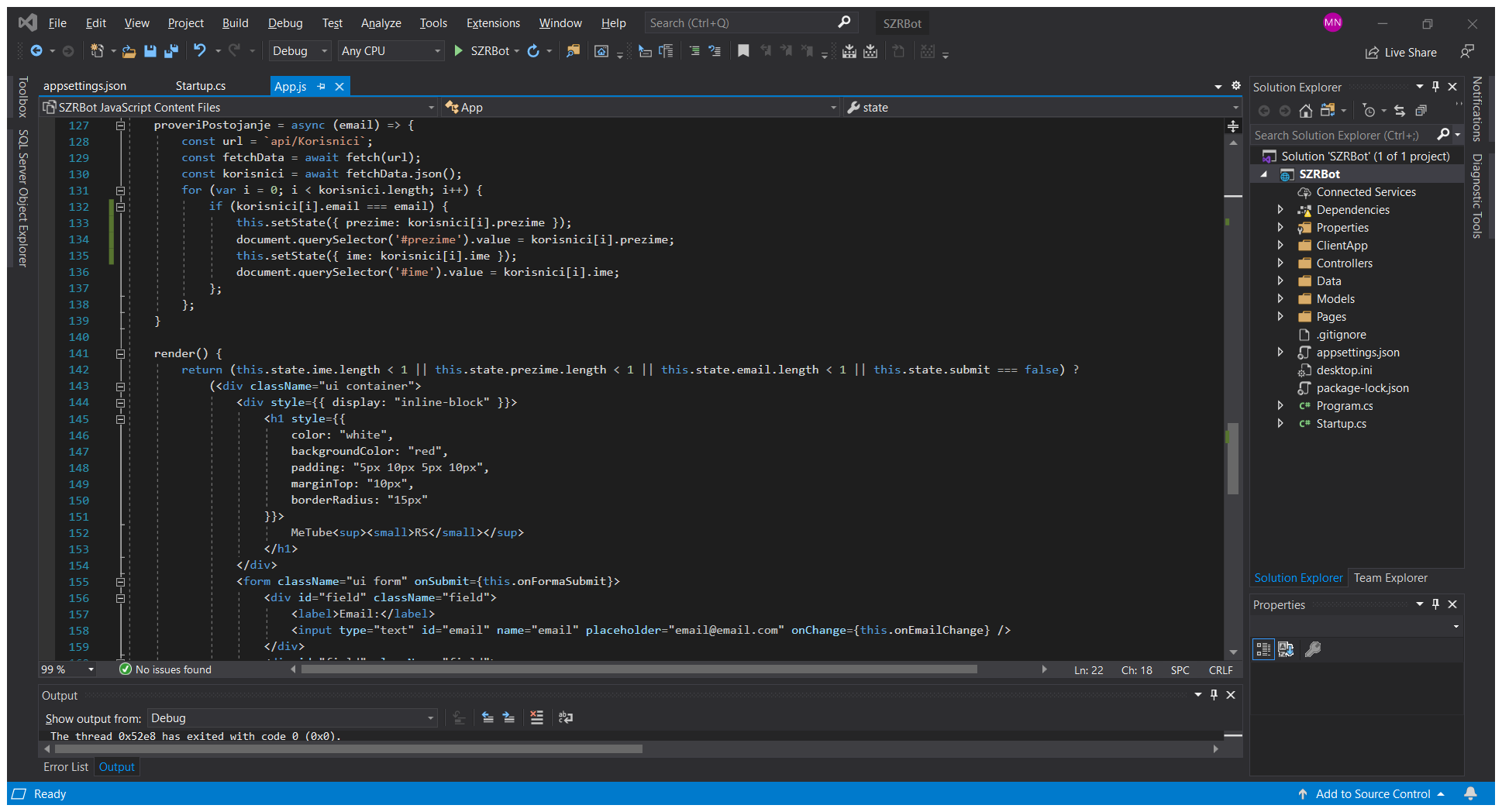
12. Kao finalnu komponentu koja će obuhvatiti sve prethodne komponente i dati im smisao, potrebno je izmeniti App.js fajl tako da izgleda kao na slikama:

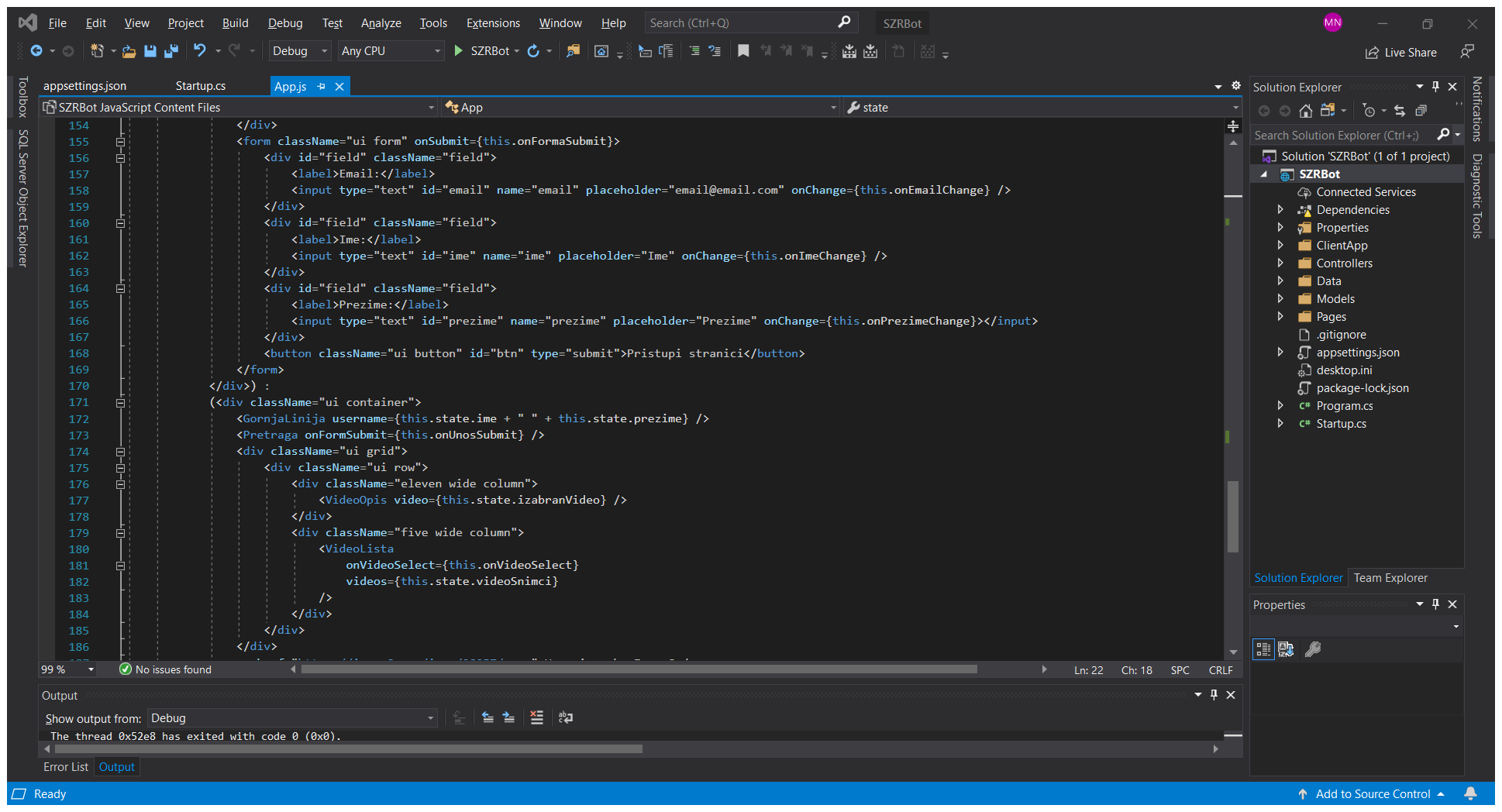


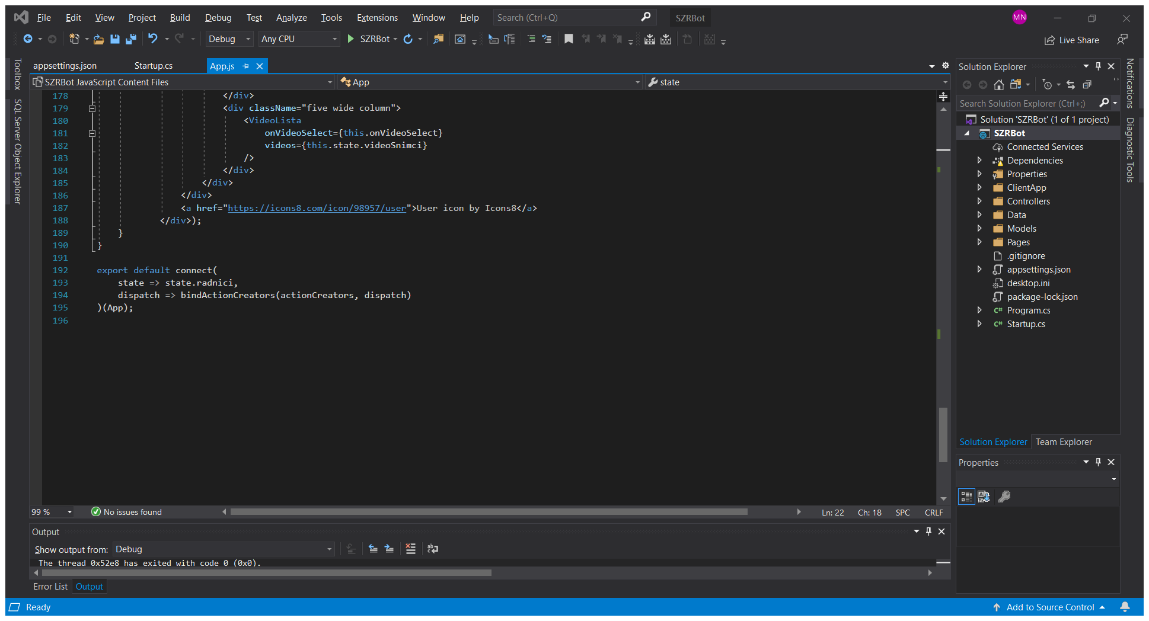




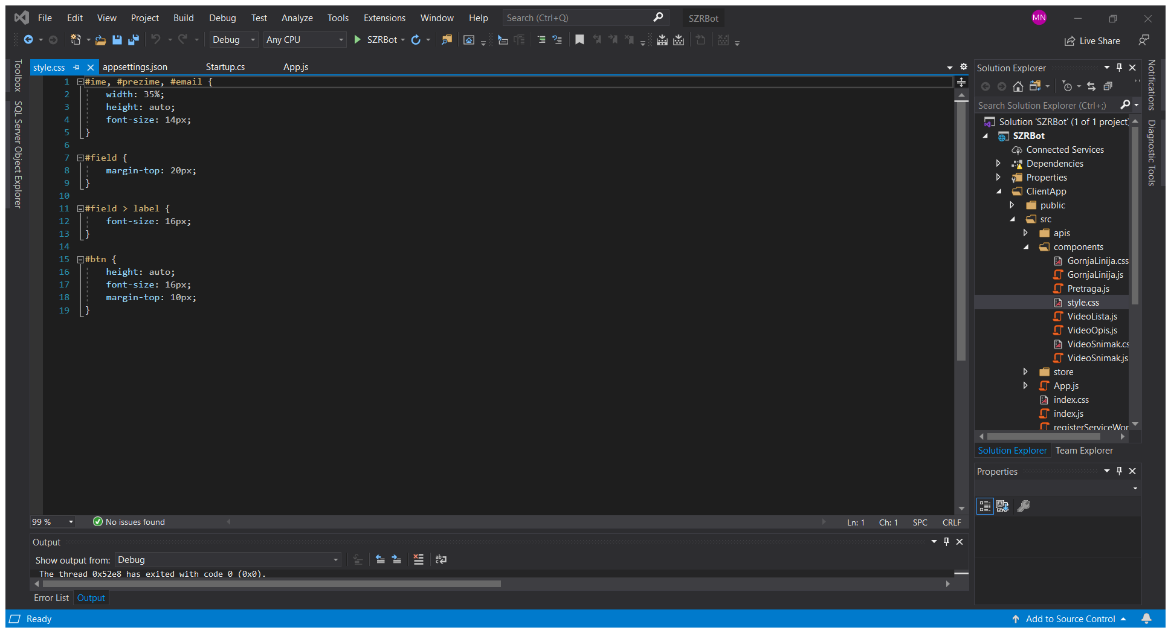






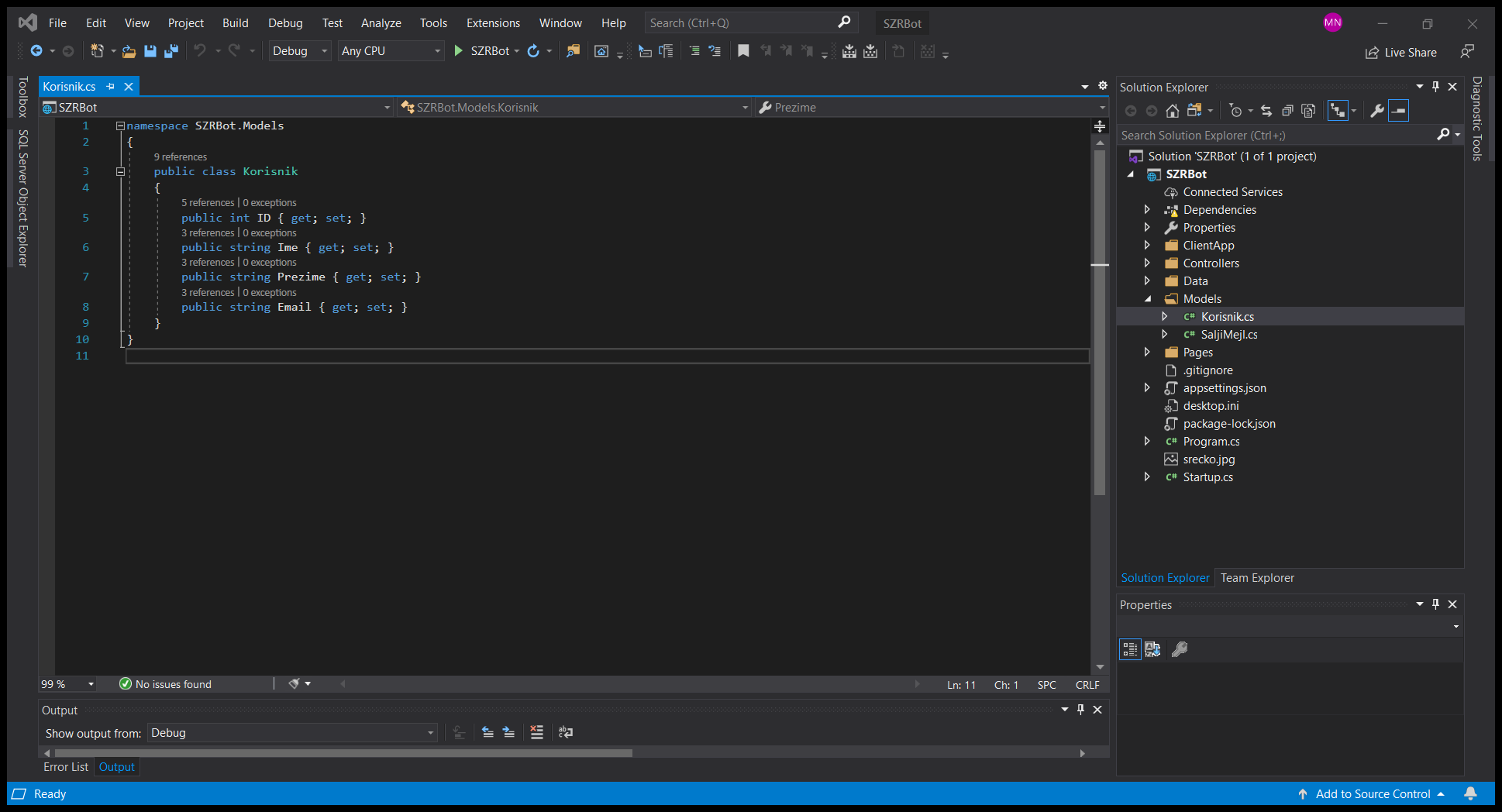


Kod ove komponente još je samo potrebno dodati kratak stilski fajl style.css (za stilizovanje forme):



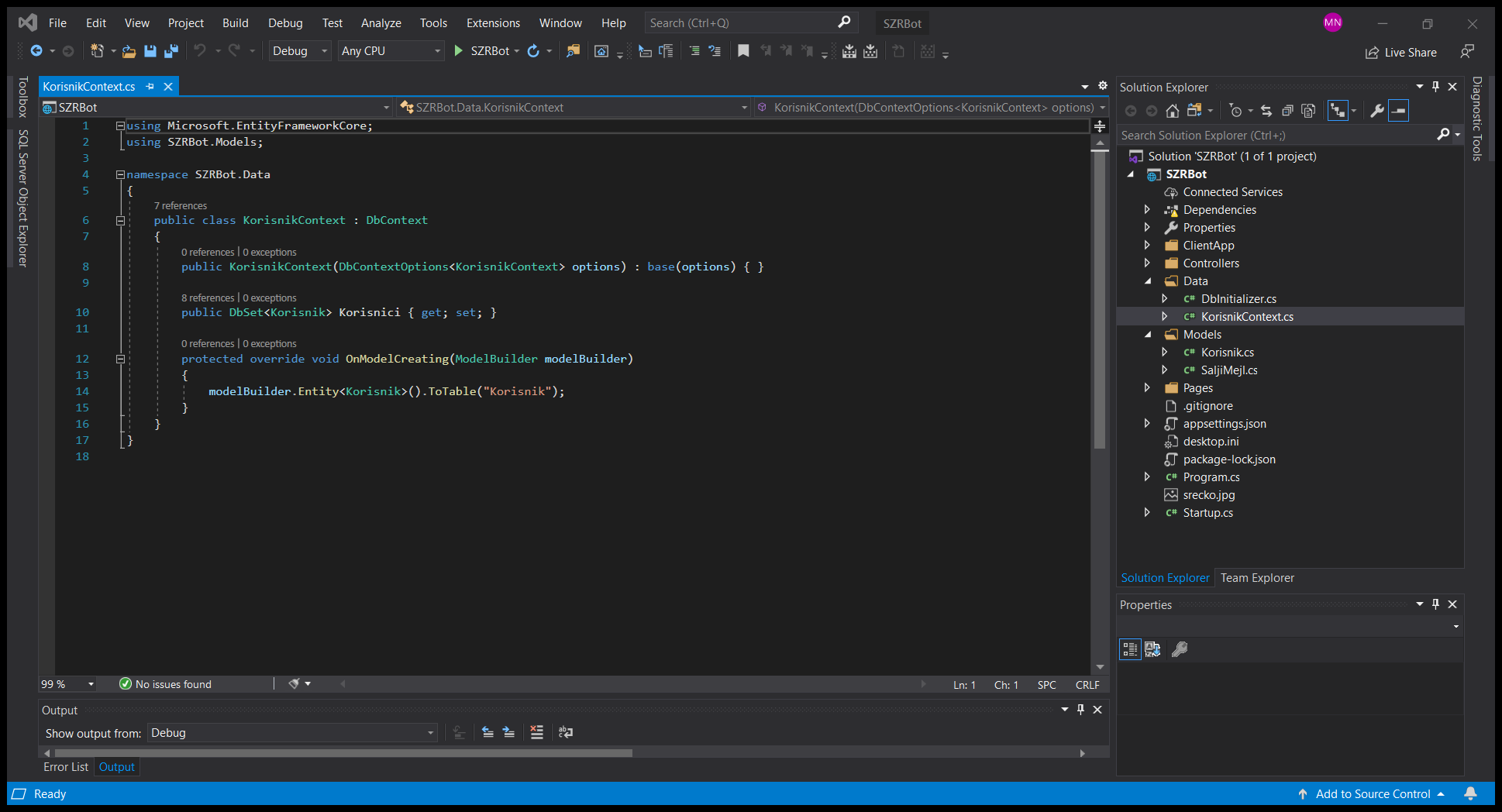
Kada je dizajn gotov, potrebno je okrenuti se bazi podataka. Za to nam služi Entity Framework i narednih 6 koraka (13-18) biće posvećeno kreiranju baze podataka sa tabelom za upis korisnika kojima je već poslat mejl.

13. Počećemo sa principom code-first. To znači da će Entity kreirane klase kao i kontekst klasu pretvoriti u bazu podataka. Prvi korak je kreiranje foldera pod imenom Models unutar projekta. Nakon toga u njemu kreirati klasu pod Imenom Korisnik.cs i preurediti je kao na slici:

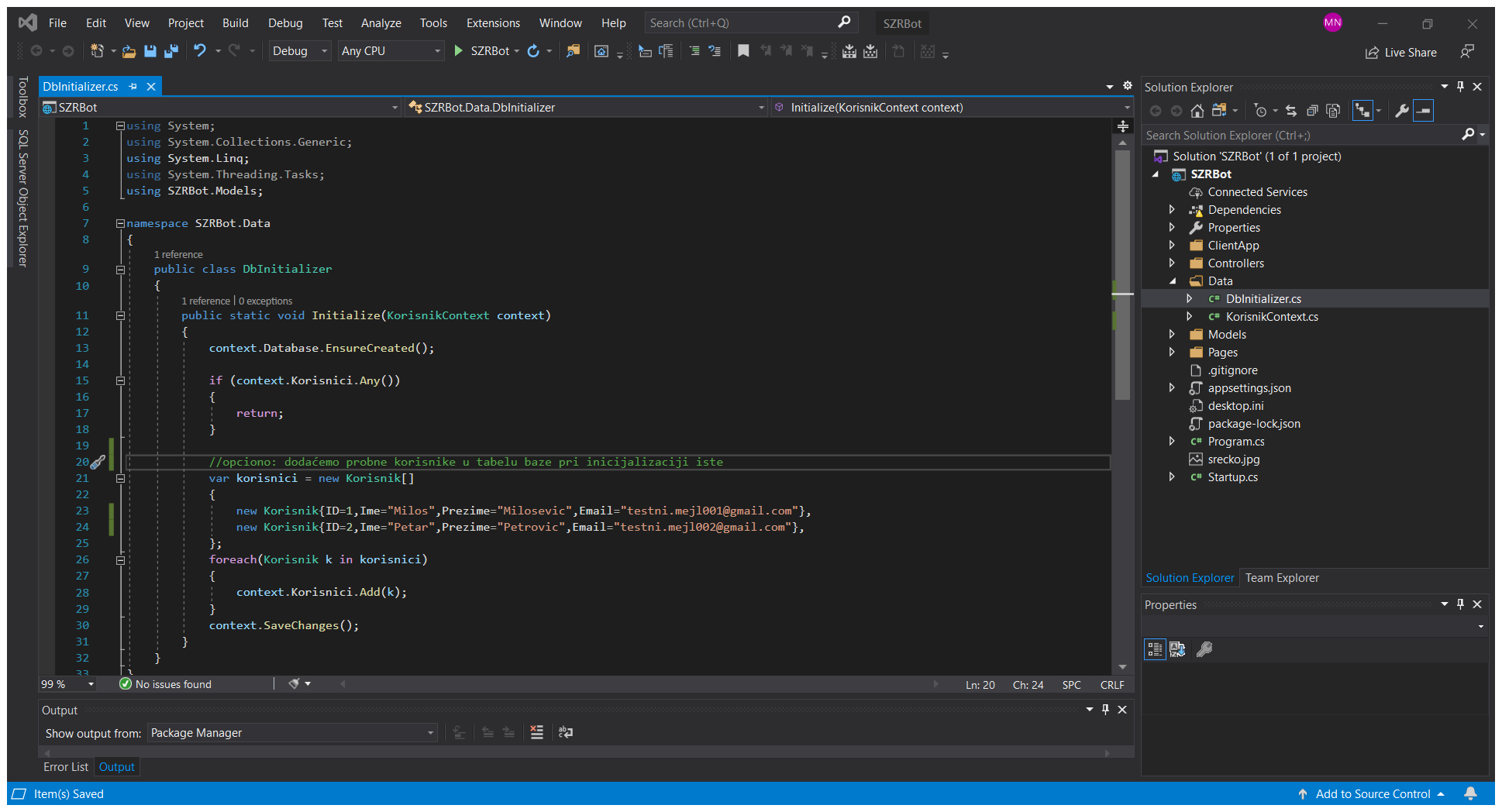


Ovako smo kreirali (kreiraće se pri prvom pokretanju aplikacije) ne samo klasu sa atributima već i kolone ID, Ime, Prezime i Email u tabeli Korisnik.

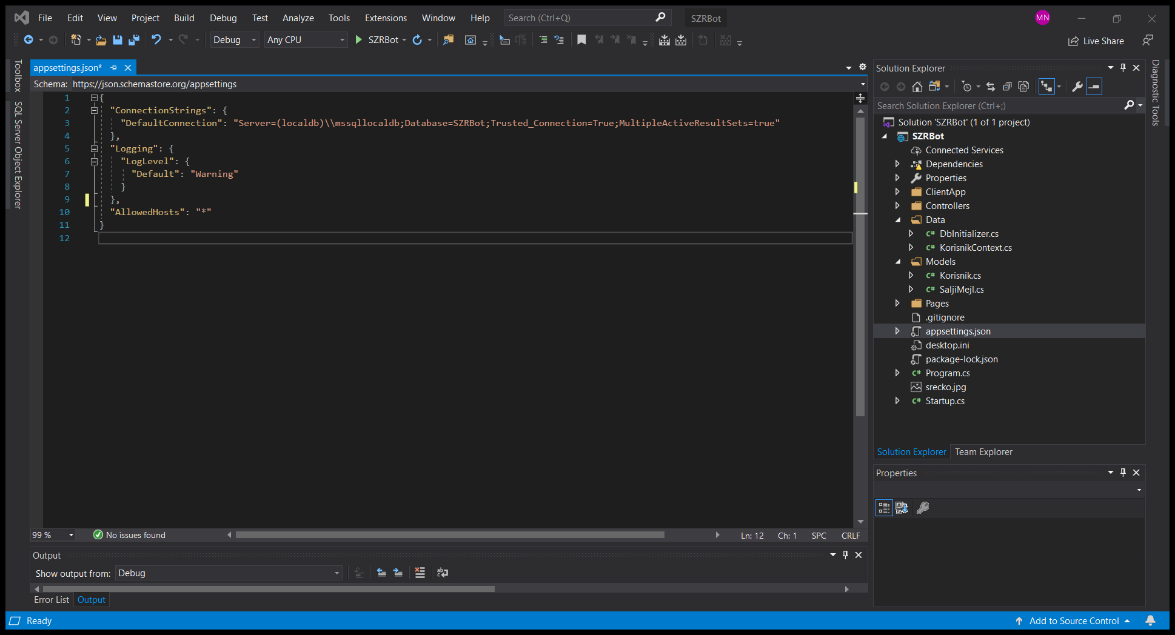
14. Nakon kreirane klase potrebno je napraviti novi folder u projektu pod imenom Data. U njega smestiti kontekst klasu pod imenom KorisnikContext koja će nasleđivati klasu DbContext:



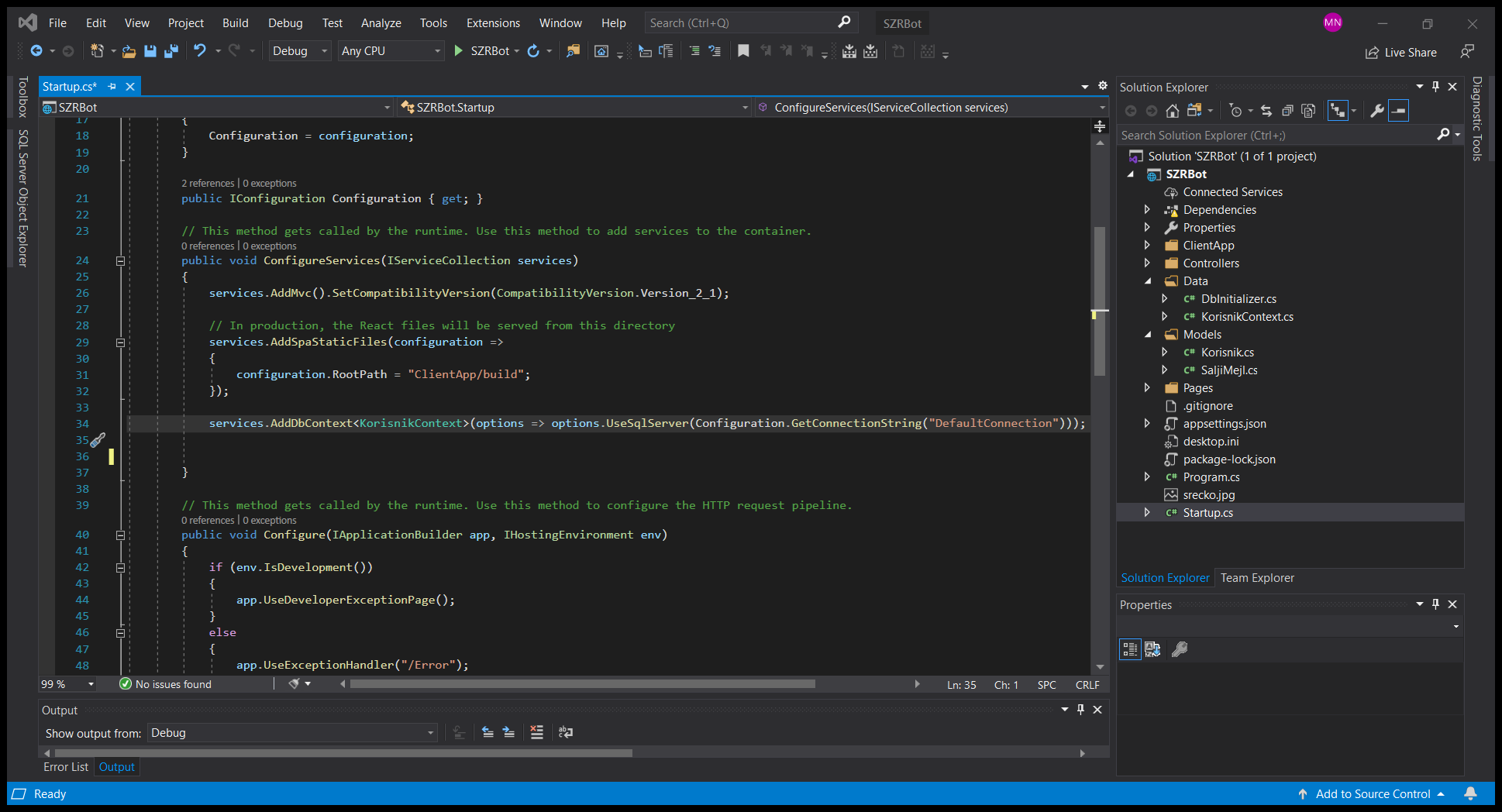
15. Sada je potrebno u istom folder Data napraviti klasu DbInitializer i ispisati kod:



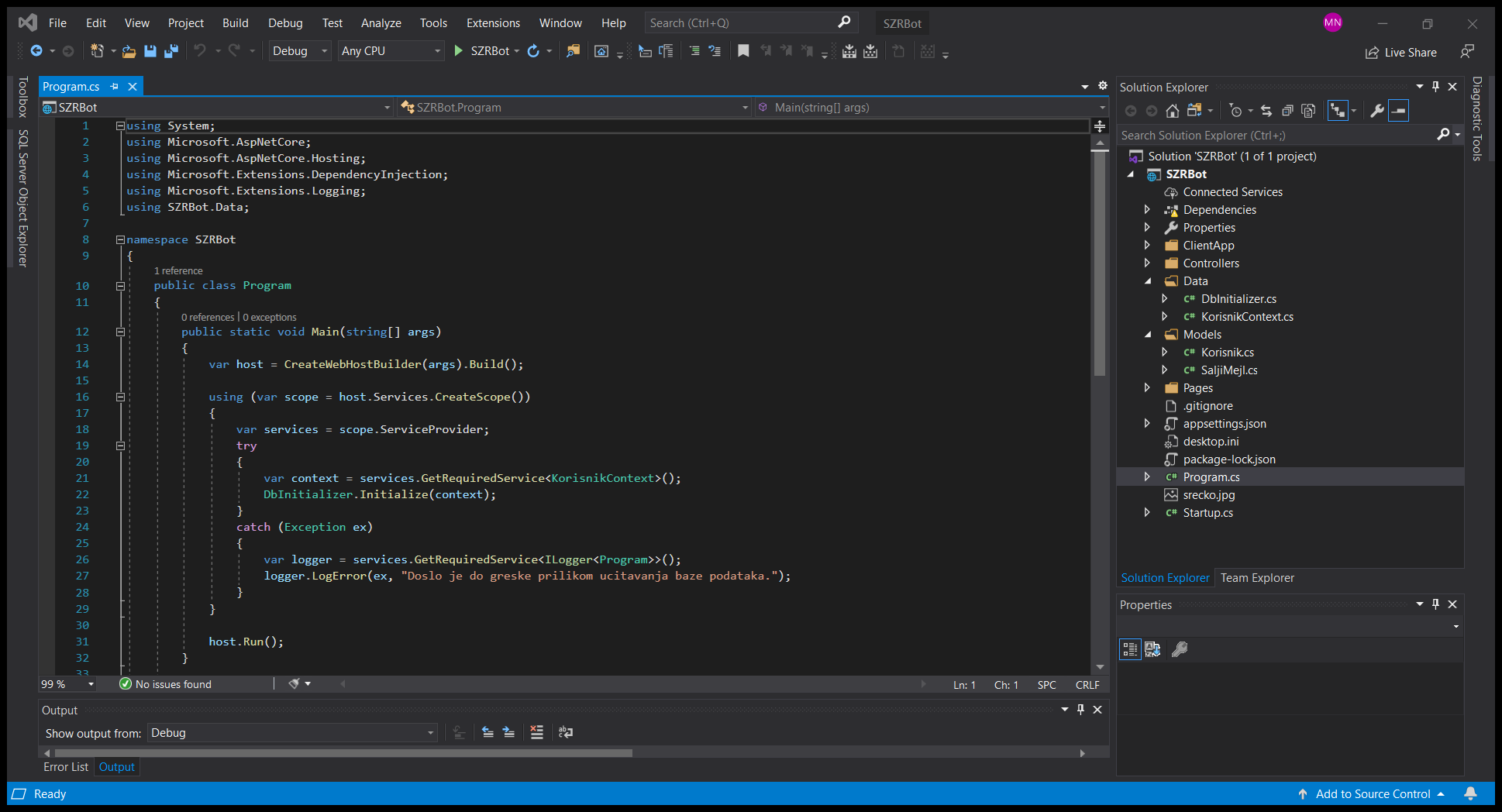
16. Kada smo ovo sredili, moramo dodati konekcioni string u fajl appsetings.json. Ovaj konekcioni string će spajati aplikaciju sa bazom kreiranom na lokalnom serveru u folderu C:/Users. Srediti ovaj JSON fajl tako da izgleda ovako:



17. U fajlu Startup.cs potrebno je u funkciji ConfigureServices() dodati highlight-ovanu liniju koda kao na slici:

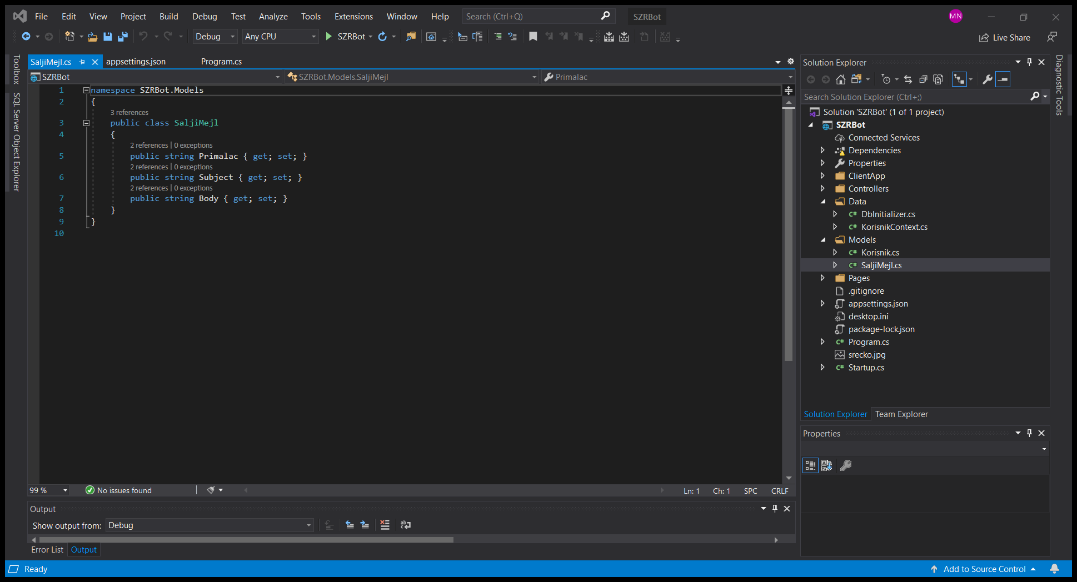


18. Nakon toga, u fajlu Program.cs potrebno je modifikovati Main metodu tako da izgleda kao na sledećoj slici:

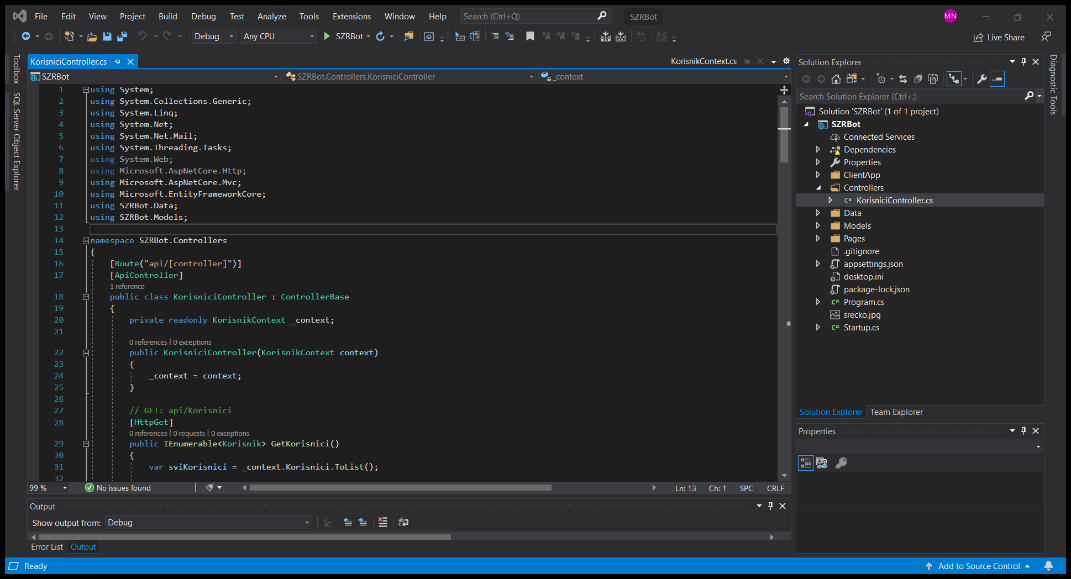


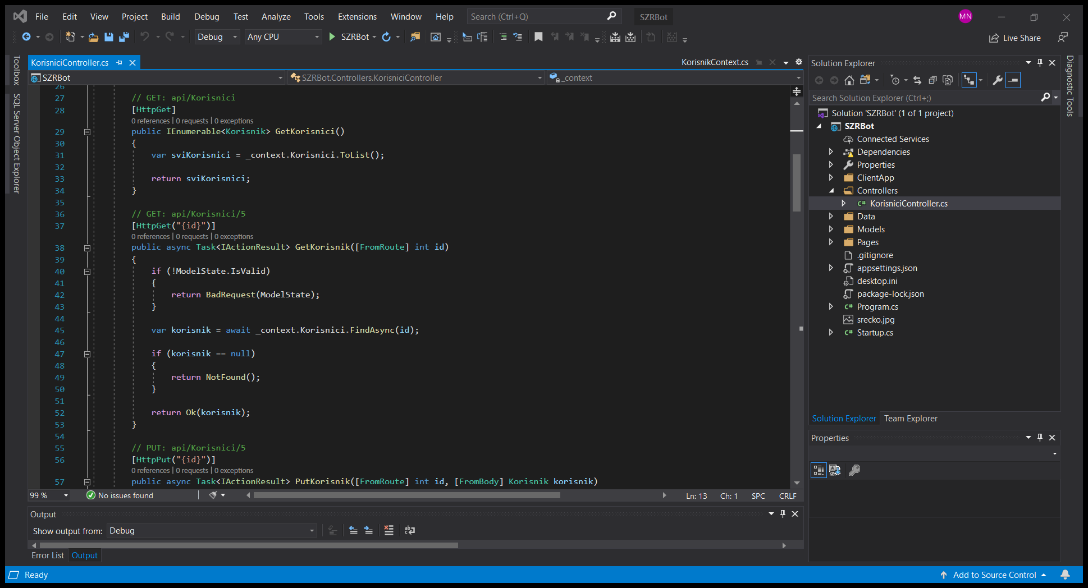
Sada smo gotovi sa kreiranjem baze. Sledeća stvar koja nam je potrebna je kontroler sa metodama za upravljanje Http request-ovima (nama će biti bitni Get i Post).

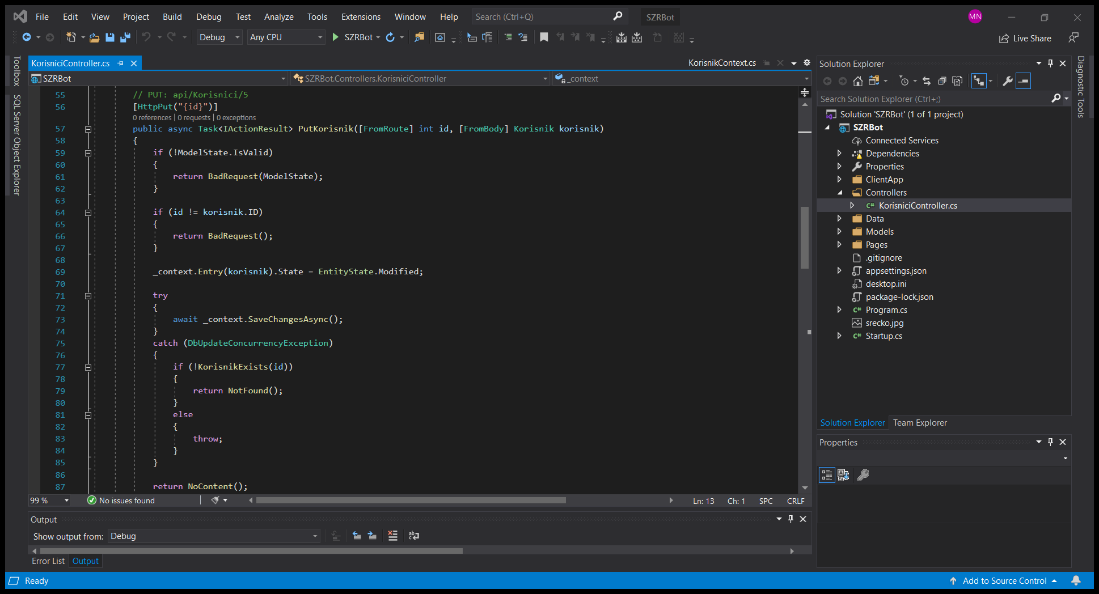
19. Pre toga potrebno je napraviti klasu koju ćemo koristiti prilikom slanja mejla, tačnije za instanciranje objekta koji se odnosi na poruku (ima sve karakteristike specifične za mejl). Ta klasa se zove SaljiMejl i nalazi se u folderu Models:

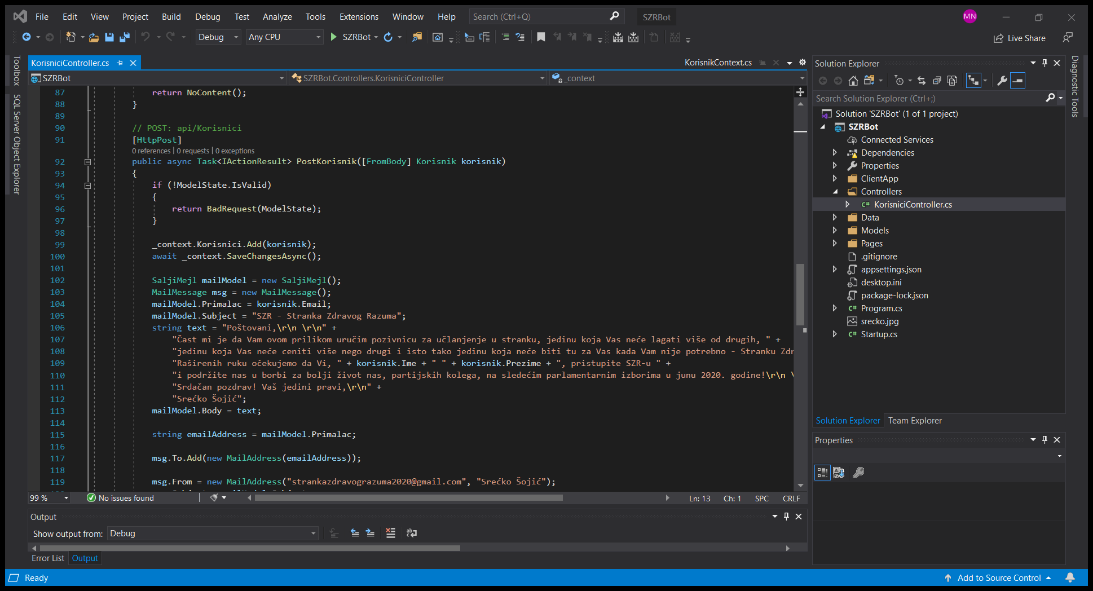


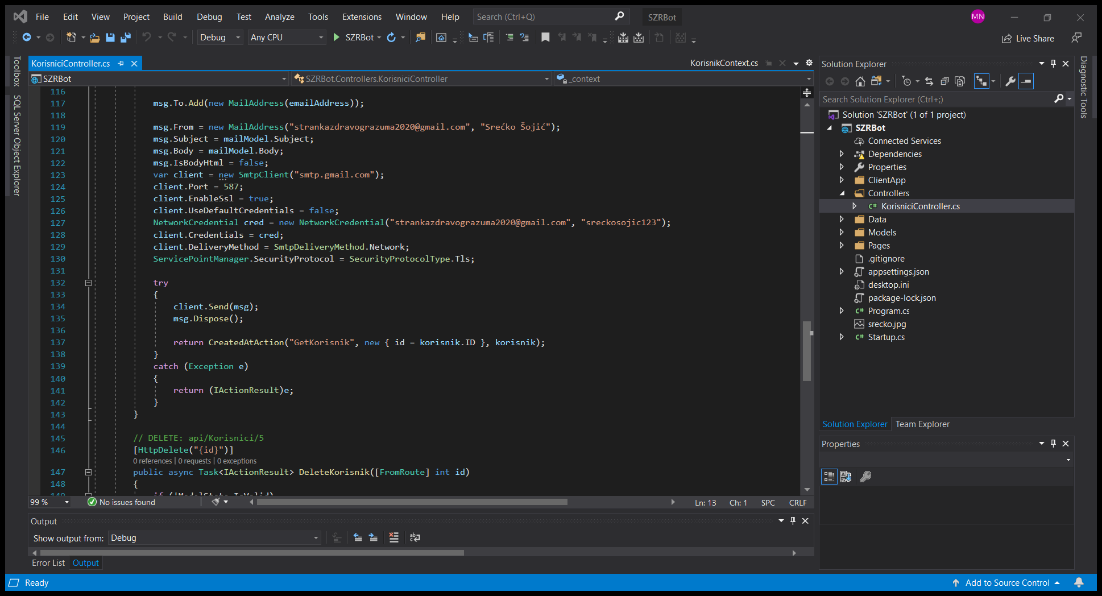
20. Potreban nam je kontroler KorisniciController koji ćemo smestiti u folder Controllers. Kreiraćemo ga tako što ćemo desnim klikom na taj folder izabrati Add -> New Scaffolded Item... i izabrati Korisnik i KorisnikContext. U ovom kontroleru će se obavljati 3 bitna zadatka: čitanje postojećih korisnika iz tabele, dodavanje korisnika u tabelu i slanje mejla korisnicima koji su novi. Generisan kod ne treba brisati već samo izmeniti odgovarajuće metode: Ovako će ovaj fajl izgledati u kodu:

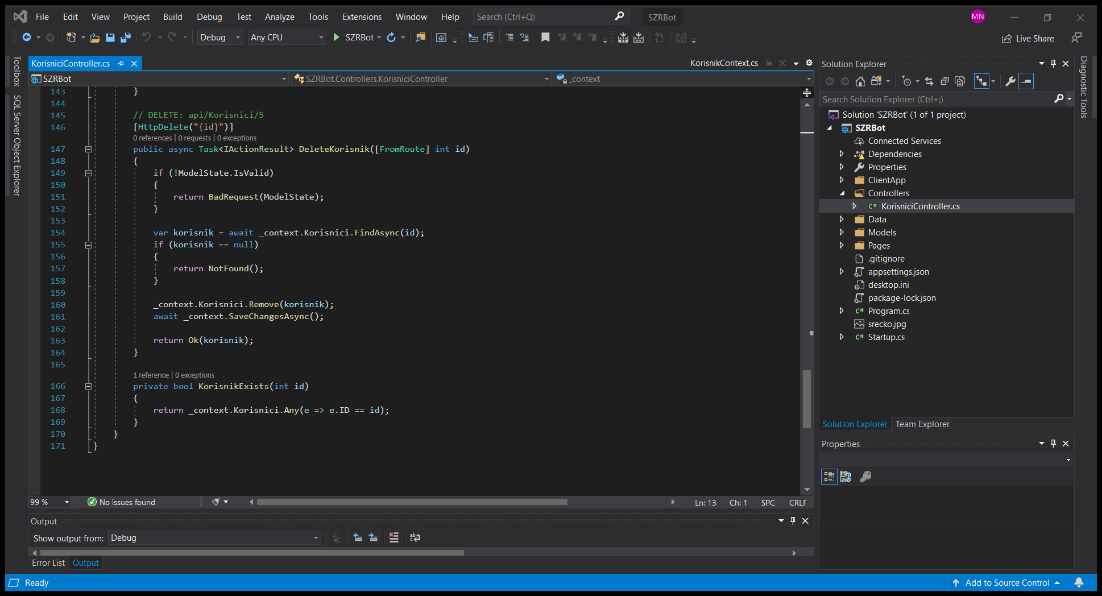




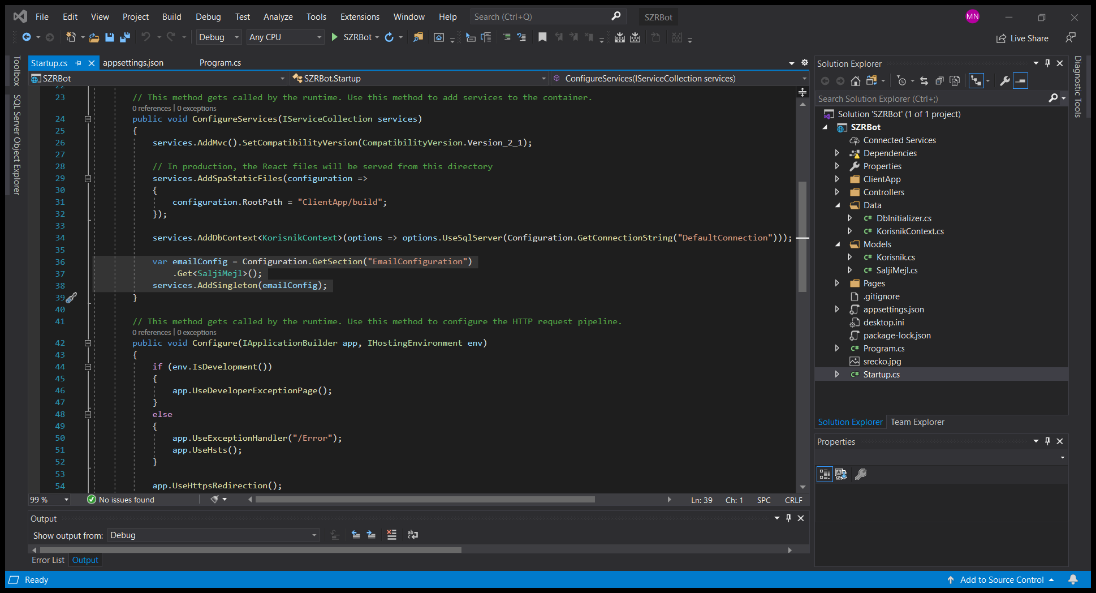


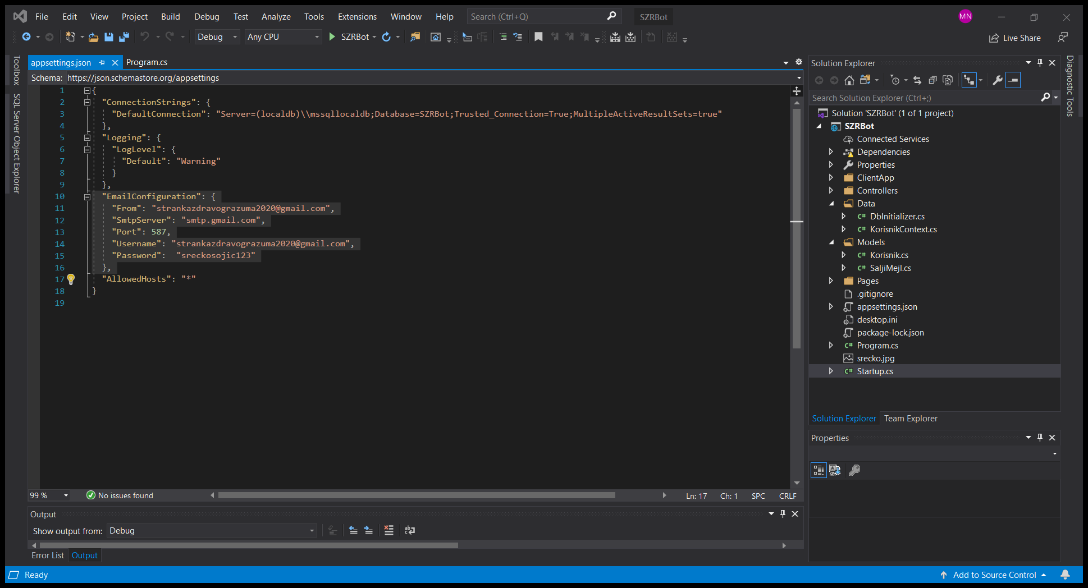






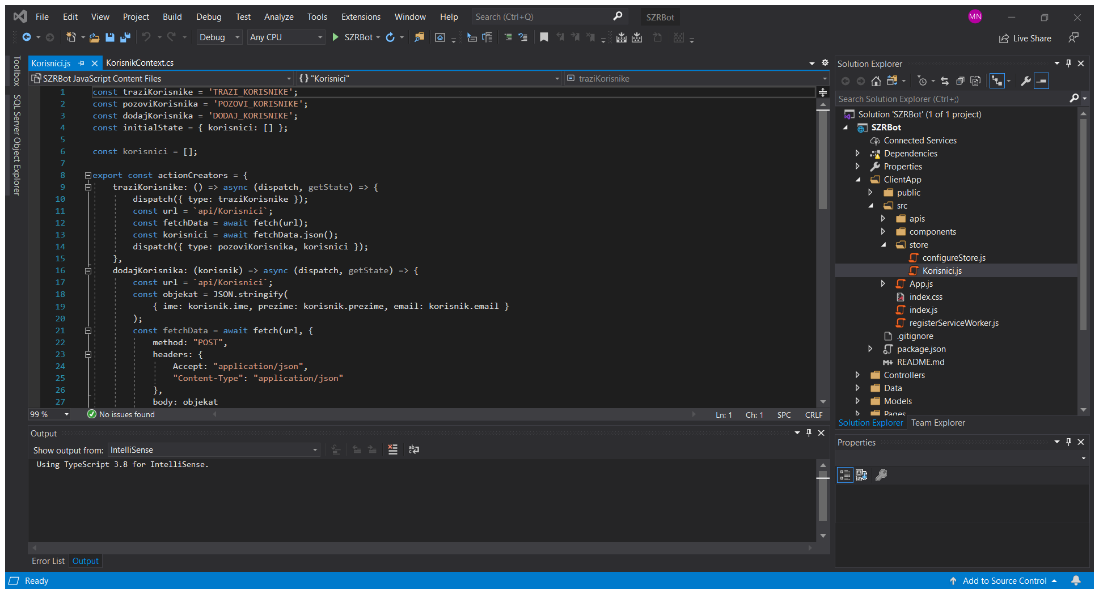
21. Da bi nam slanje mejla bilo u potpunosti implementirano, potrebno je izvršiti male prepravke već kreiranih fajlova: appsettings.json i Startup.cs. Izmene su highlight-ovane na sledećim slikama:

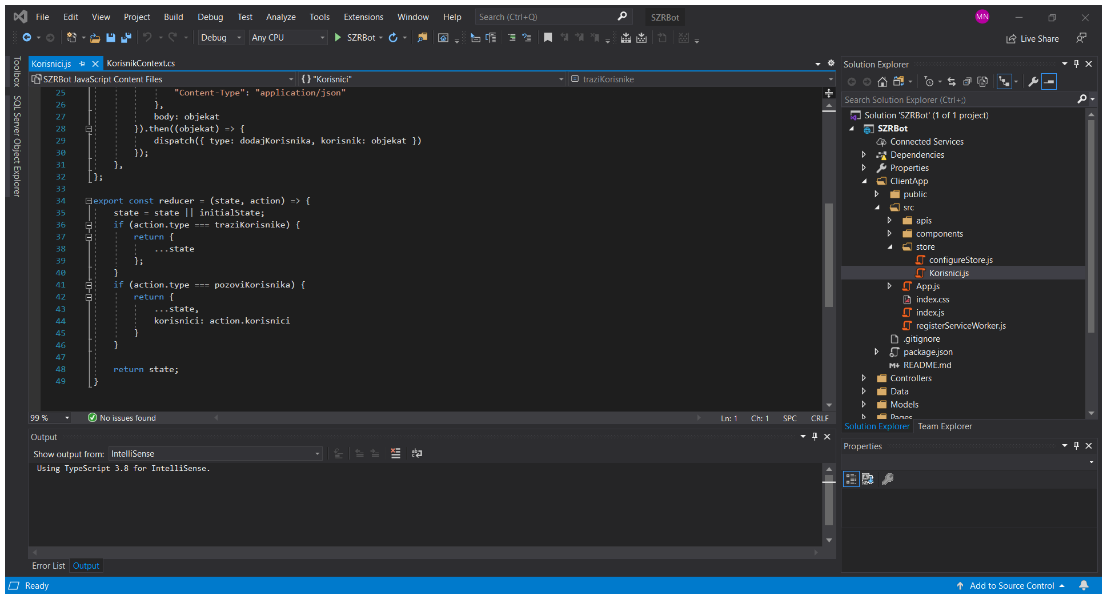




Na redu su koraci (22 i 23) koje treba odraditi kako bi se implementirao Redux store u folderu store.

22. Definisane su akcije koje šalju zahteve serveru (pozivaju RESTful funkcije unutar kontrolera):

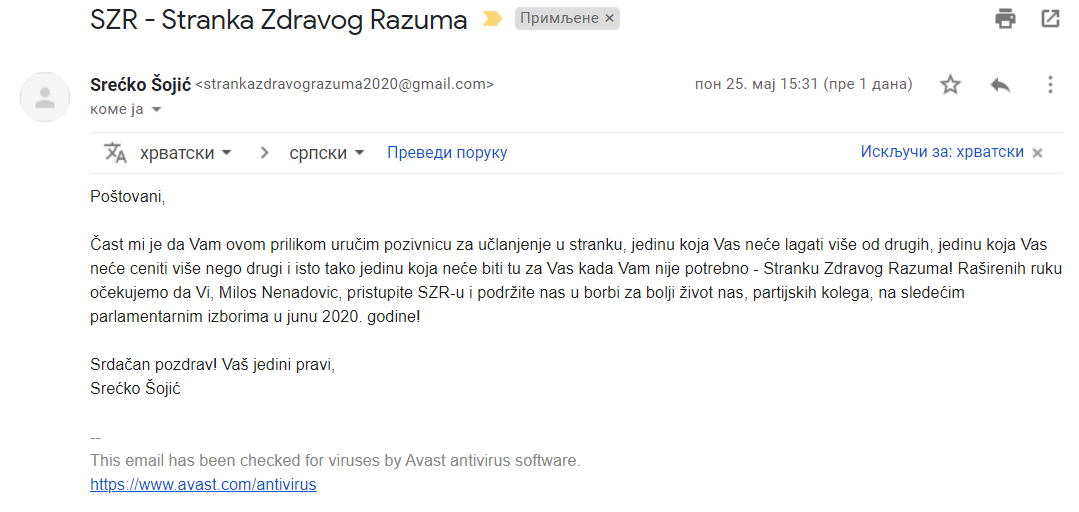




23. U fajlu configureStore.js u folderu store, potrebno je preraditi kod tako da izgleda kao sa slike:



Ovime smo završili izradu aplikacije pa ceo projekat možemo Build-ovati i pokrenuti na SZRBot serveru. Finalni rezultat u slučaju ciljanog pretraživanja jeste mejl sličan mejlu sa slike:



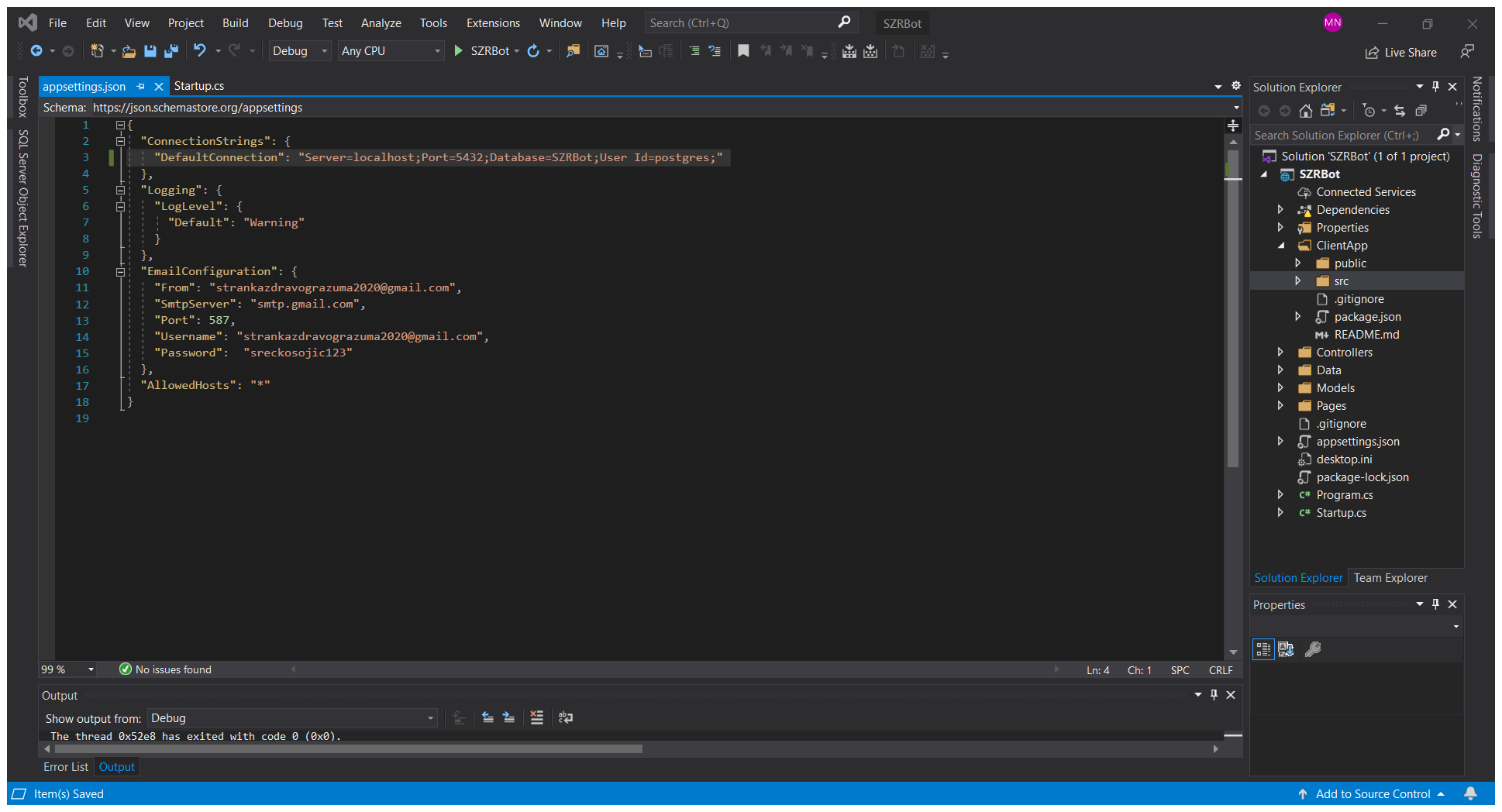
**Dodatak - izmena za open-source bazu podataka (PostgreSQL):**

Kako bismo koristili open-source bazu umesto MS SQL, potrebno je izvršiti nekoliko sitnih izmena.

Pre svega potrebno je skinuti i instalirati PostgreSQL sa sajta https://www.postgresql.org/download/windows/ (za korisnike windows operativnog sistema). Verzija koju ćemo instalirati je 10.13 (poslednja koja je dostupna za sve platforme).

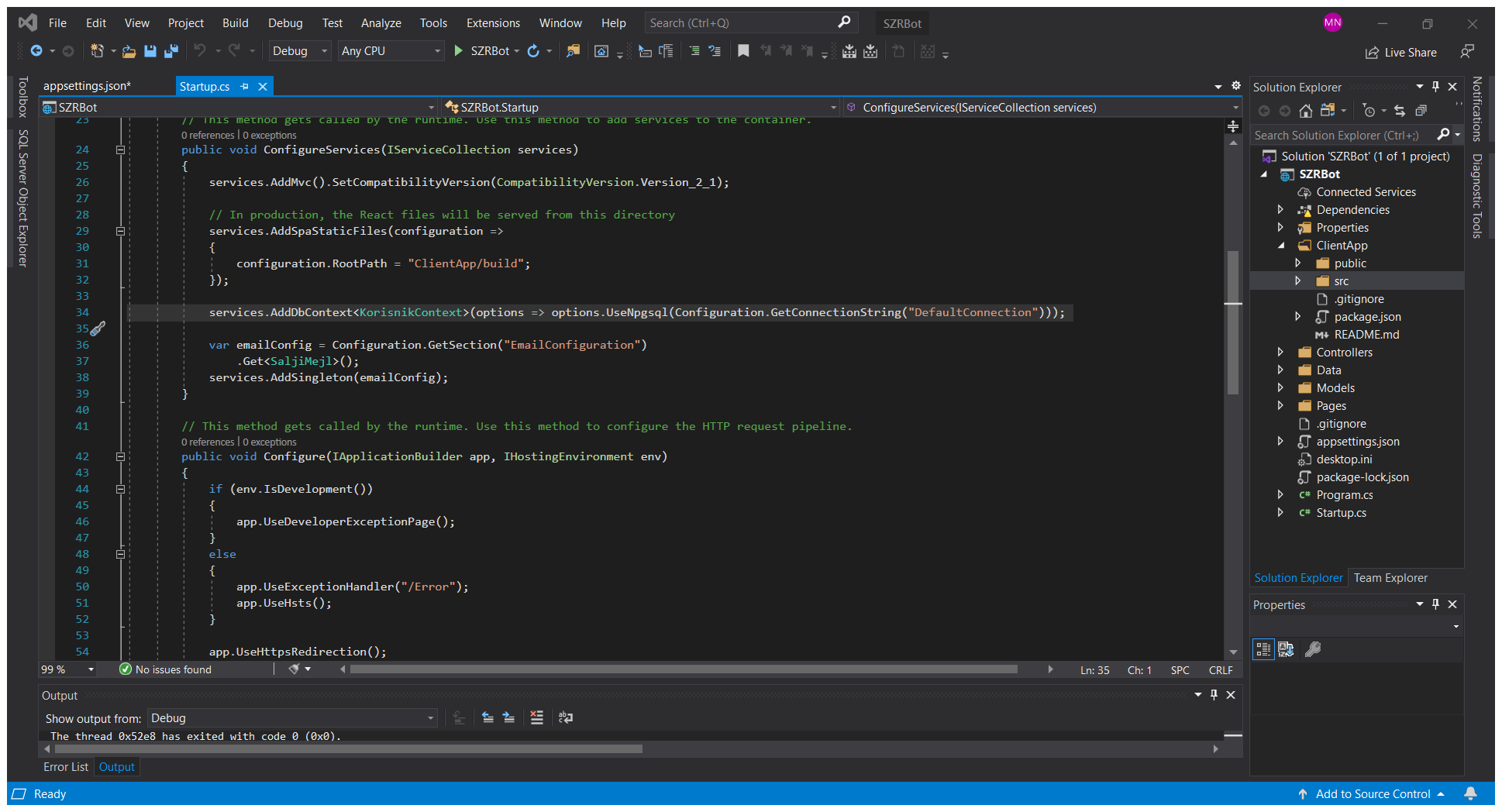
Nakon kompletirane instalacije (uključujući i restartovanje računara), potrebno je pronaći fajl pg\_hba.conf koji se nalazi unutar foldera data u instaliranom direktorijumu PostgreSQL. Taj fajl je potrebno otvoriti pomoću tekst editora (pokrenuti kao administrator). Nakon toga skrolovati do dna tog tekstualnog fajla gde se nalazi deo teksta koji podseća na tabelu sa informacijama o konekcijama kao i o metodi zaštite (md5). Taj deo teksta treba prepraviti tako da svuda gde piše md5 prepraviti u trust. Ovako se omogućuje modifikacija baze.

1. U fajlu appsetting.json potrebno je izmeniti konekcioni string ("Default Connection" property "Connection Strings" objekta) tako da izgleda kao sa slike:



Linija koju treba izmeniti je higlight-ovana. Ovaj konekcioni string izuzima upotrebu Password polja a to nam je omogućeno modifikacijom fajla pg\_hba.conf gde smo za metodu autentifikacije umesto md5 stavili trust.

2. Naredna izmena je unutar Startup.cs fajla:



Highlight-ovanu liniju koda treba izmeniti tako da se poziva funkcija UseNpgsql() funkcija umesto UseSqlServer().