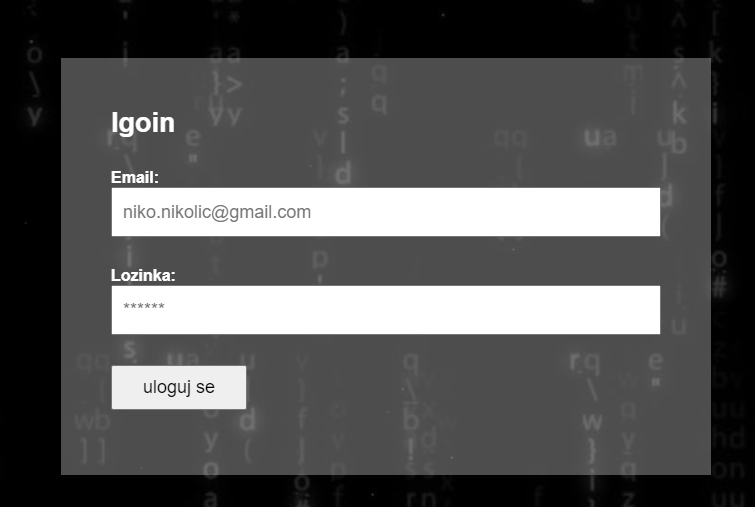
**LOGIN DUGME**

Dakle sada se zelimo pozabaviti sa loginom da se ne moramo stalno registrovati sa novim emailom,passwordom i korisnickim imenom.Dakle ozivljavamo ovaj dio aplikacije:



Dakle ovde zelimo da u polje email unesemo email koji postoji u mockAPI-ju, tj. email nekog naloga kojeg smo ranije kreirali.Takodje u polju lozinka zelimo da potvrdimo da se radi o ispravnoj lozinki koja je takodje pohranjena u mockAPI-ju.

Ono sto trebamo sada je da napravimo loginView.js fajl u kojem cemo osposobiti radnju ulogovanja u aplikaciju.

Ovako izgleda kreirana loginView.js fajl:

loginView.js fajl:

Sta smo ovde uradili:

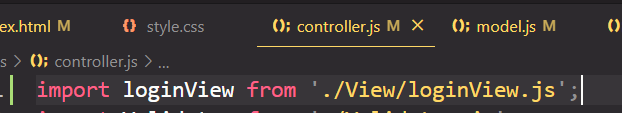
Dakle uvezli smo session= new Session();

Kreiramo LogginView clasu i u njoj smo definisali \_btn, \_parrentElement i \_loginForm varijable, te smo takodje jos definisal login() metodu.  
  
U toj login() metodi smo uradili sledece:

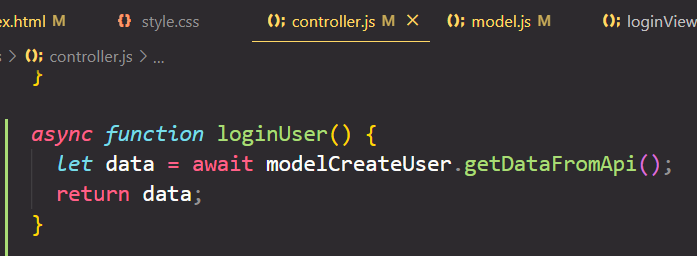
login nam ima argument (dataA) koji nam prestavlja funkciju u controller.js fajlu:

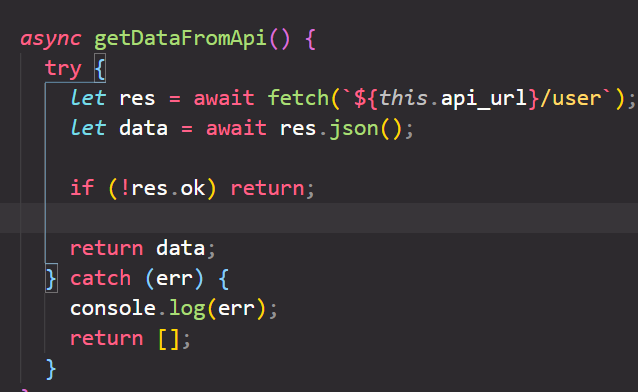


Ovde u ovom controller.js fajlu vidimo da smo u init( ) funkciji pozvali nas loginView.login( loginUser) => ovo loginUser je nasa login( dataA ) metoda iz loginView.js fajla.  
Prije nego sto smo pozvali loginView.login( loginUser ) morali smo prvo da importujemo ovu clasu u controller.js fajl:



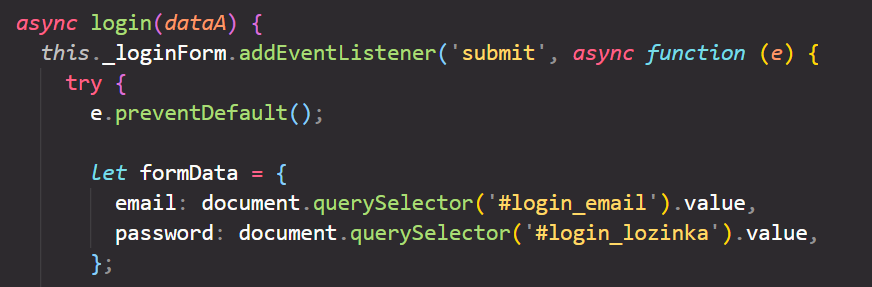
sada se i dalje nalazimo u controller.js fajlu gdje cemo da objasnimo sta smo uradili sa loginUser() funkcijom:

dakle ovde smo definisali data varijablu u koju smo sacuvali podatke iz mockAPI-ja (pozivajuci metodu getDataFromApi() koja nam poziva iz model.js fajla getDataFromApi() metodu):

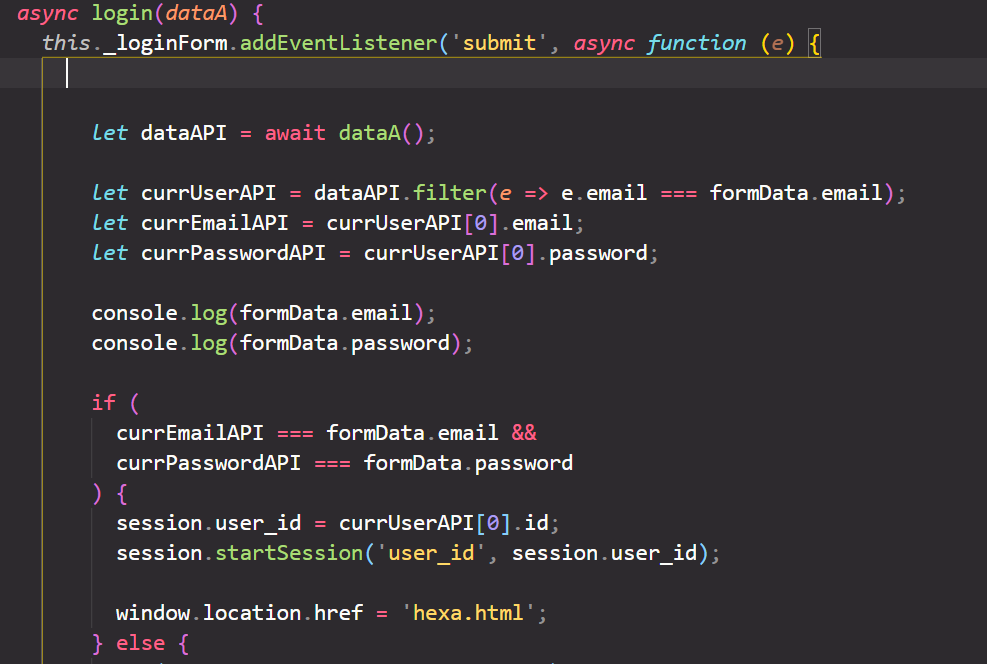


Ova getDataFromApi() metoda nam poziva /user iz mockAPI-ja (podatke user-a iz mockAPI-ja).

Te sada kada su nam se podaci sacuvali u data varijabli u loginUser() funkciji u controller.js fajlu zelimo te data podatke da return (vratimo) da bi smo ih mogli iskoristiti u login(dataA) funkciji u loginView.js fajlu.Dakle ova nasa loginUser() funkcija prestavlja argument funkcije login(dataA) iz loginView.js fajla.

Sada cemo da rasclanimo dio po dio login( dataA ) funkcije.  
Dakle kada se dogodi ‘submit’ dogadjaj od this.\_loginForm (koja prestavlja dugme => kada se forma submituje).Kada se dogodi taj ‘submit’ onda zelimo da u formData objektu sacuvamo podatke koje je korisnik unio u login polje ( email i password ).

sledece je:

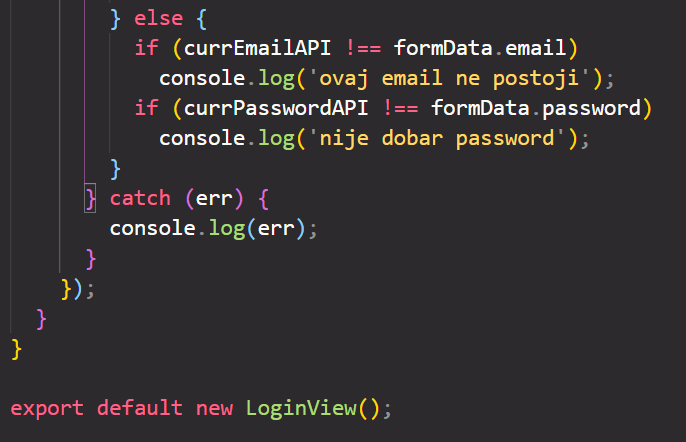
kada smo sacuvali podatke korisnika koje je unio u login polje sada cemo da sacuvamo u dataAPI varijablu vrijednost koju smo pozvali iz argumenta ove nase login metode. Program treba da saceka asinhronu funkciju da se mogu pozvati podaci iz mockAPI-ja.

Kada su podaci sacuvani u dataAPI varijablu zelimo da u currUserAPI varijabli sada sacuvamo vrijednost iz mockAPI-ja onog korisnika koji odgovara vrijednosti koje je korisnik unio u login polje.Sada kada smo to ispunili pomocu filter () array.metode sada cemo iz tog podatka (iz te nase currUserAPI varijable koja je objekat) da izvucemo vrijednost email i vrijednost password-a.

Kada imamo sada vrijednost emaila i passworda iz mockAPI-ja sada mozemo da postavimo if uslov koji kaze ako trenutni email i password iz mockAPI-ja odgovaraju unijetim vrijednostima iz login polja onda zelimo da se kreira ‘user\_id’ cookie sa vrijednosti currUserAPI[0].id => id-jom trenutnog korisnika koji dobijamo iz mockAPI-ja.

Kada smo kreirali cookie pomocu startSession() varijable kazemo da nam se sada otvori ‘hexa.html’ stranica.

sledece je ‘else’ ako se ne ispuni ovaj uslov:

Dakle ako nam korisnik nije unio email koji postoji u mockAPI-ju onda zelimo da nam izbaci neko upozorenje ili da nije dobar email ili da nije dobar password.

Sada imamo osposobljenu login formu.