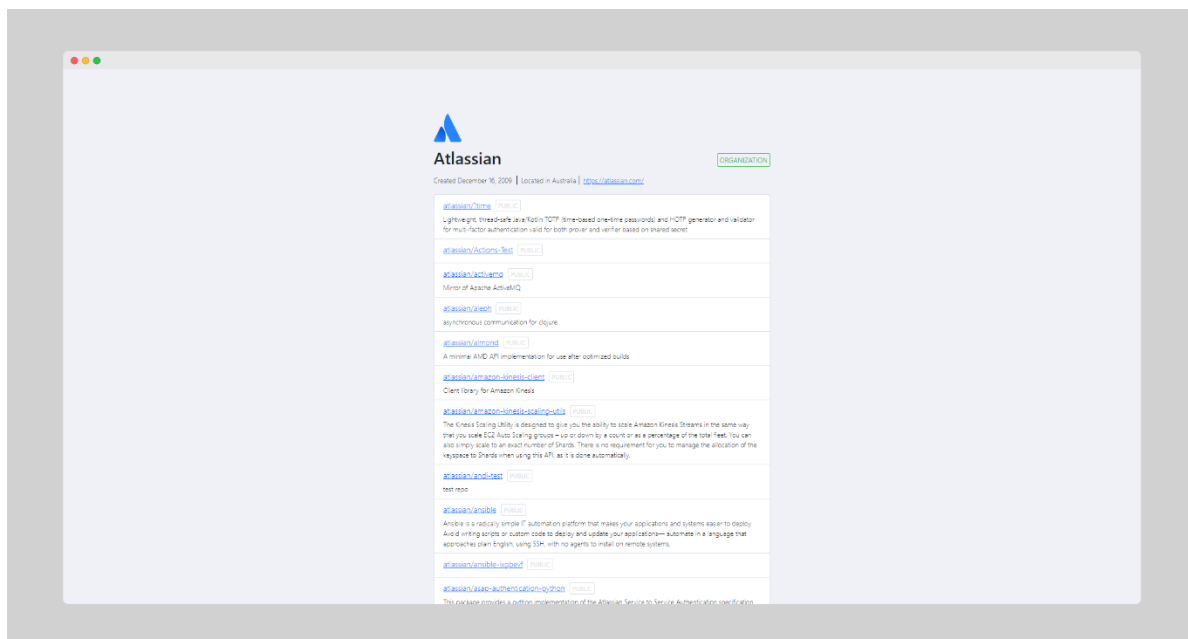


Algebra – Front-End Developer
ReactJS – Ispit



IZGLED ZAVRŠNOG RJEŠENJA

Pravila

- Dopušteno je koristiti **isključivo vanjsku literaturu** (Google, W3 Schools, MDN, itd.)
- Nije dopušteno razgovarati s drugim polaznicima
- Kad ste gotovi s zadatkom prenesite ga u zajedničku mapu [\lzg-c5-00](#), podmapa Front-End Developer – Nino Škuflić, podmapa **DROPBOX**
 - Mapa mora biti u sljedećem formatu (*vrijedi i za ZIP u slučaju online polaznika*): **IME PREZIME – CSS ISPIT** (npr. NINO ŠKUFLIĆ – CSS ISPIT)
 - **ONLINE POLAZNICI**: Pošaljite rješenje predavaču putem maila na nino.skuflic@predavaci.algebra.hr u ZIP formatu.
 - **VAŽNO**: Ako mapa / ZIP arhiva nije u navedenom formatu vaš ispit **NEĆE BITI PREGLEDAN!**

Zlatno pravilo

Iako je ovo vaš rad i dalje je ispit, stoga nije dopušteno spremanje na bilo kakve medije - ukratko, ostaje unutar Algebre, odnosno naše učionice. 😊

Zadatak

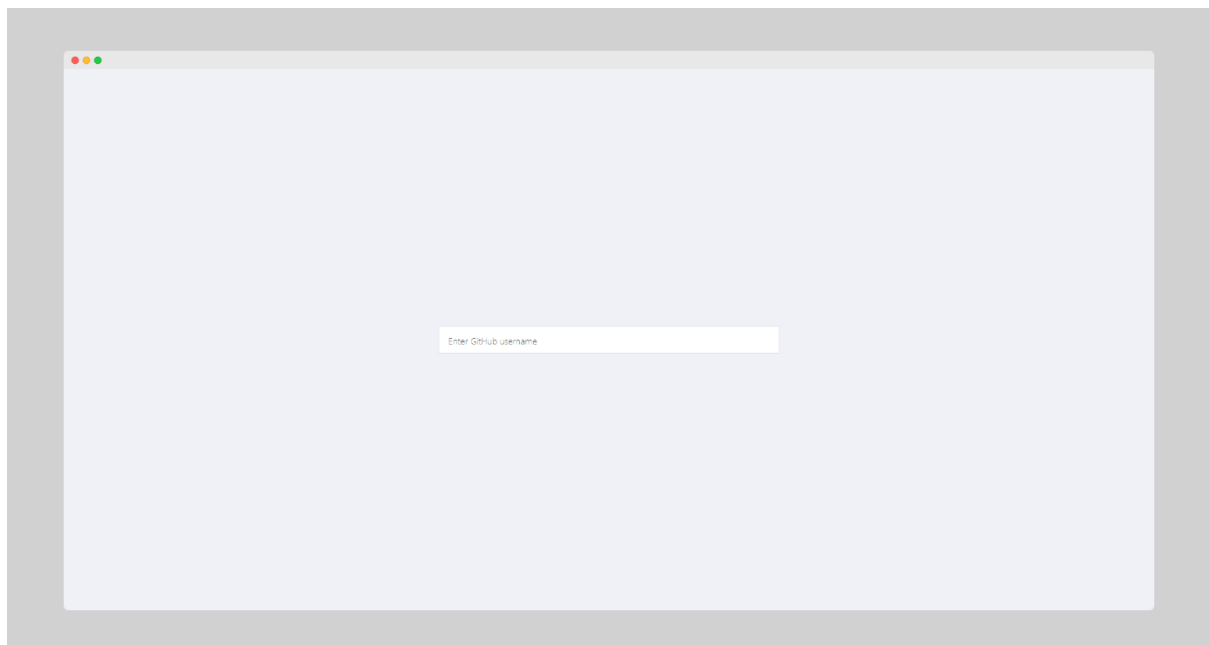
1. U sklopu parcijalnog ispita, potrebno je napraviti jednostavnu React aplikaciju koja se sastoji od dva dijela:
 - Forma (obrazac) za unos korisničkog imena
 - Prikaz detalja unešenog korisnika
2. Kao bazu korisnika, koristit ćemo GitHub (<https://github.com>). GitHub je servis na kojem se nalazi velik broj programskih repozitorija. Na primjer:
 - React: <https://github.com/facebook/react>
 - Atlassian: <https://github.com/atlassian>
3. Svaki repozitorij ima svog vlasnika. Svaki korisnik na GitHubu može imati mnogo repozitorija. Na primjer, ovo je pregled svih repozitorija koje posjeduje korisnik Facebook: <https://github.com/facebook>.

Forma za unos korisničkog imena:

- Forma se sastoji od jednog input polja i jednog buttona.
- U input polje potrebno je upisati korisničko ime bilo kojeg GitHub korisnika - npr. "atlassian".
- Korisničko ime GitHub korisnika moguće je pročitati iz bilo kojeg GitHub URL-a.
- **Primjer:**
 - <https://github.com/facebook/react> - korisničko ime = facebook
 - <https://github.com/atlassian/escalator> - korisničko ime = atlassian

Po unosu GitHub korisničkog imena, na klik buttona, potrebno je s GitHub REST API-ja dohvatiti i prikazati detalje tog korisnika i njegove repozitorije.

Primjer kako forma može izgledati:



GitHub REST API

GitHub API nalazi se na sljedećem URL-u: <https://developer.github.com/v3/repos>

Za uspješno odrađivanje ove vježbe, bitna su dva poziva:

1. Dohvaćanje GitHub usera

- a. Dohvaćanje GitHub usera moguće je napraviti GET pozivom na sljedeći URL:
 - i. <https://api.github.com/users/<username>>

Za dohvaćanje detalja korisnika s korisničkim imenom facebook, potrebno je napraviti GET poziv na URL <https://api.github.com/users/facebook>

Primjer odgovora:

```
1 // 20231220172130
2 // https://api.github.com/users/atlassian
3
4 {
5   "login": "atlassian",
6   "id": 168166,
7   "node_id": "MDY6MDYyZ2IuaxphdGlvbjE2ODDE2Hg=",
8   "avatar_url": "https://avatars.githubusercontent.com/u/168166?v=4",
9   "gravatar_id": "",
10  "url": "https://api.github.com/users/atlassian",
11  "html_url": "https://github.com/atlassian",
12  "followers_url": "https://api.github.com/users/atlassian/followers",
13  "following_url": "https://api.github.com/users/atlassian/following/{other_user}",
14  "gists_url": "https://api.github.com/users/atlassian/gists/{gist_id}",
15  "starred_url": "https://api.github.com/users/atlassian/starred/{owner}/{repo}",
16  "subscriptions_url": "https://api.github.com/users/atlassian/subscriptions",
17  "organizations_url": "https://api.github.com/users/atlassian/orgs",
18  "repos_url": "https://api.github.com/users/atlassian/repos",
19  "events_url": "https://api.github.com/users/atlassian/events/{privacy}",
20  "received_events_url": "https://api.github.com/users/atlassian/received_events",
21  "type": "Organization",
22  "site_admin": false,
23  "name": "Atlassian",
24  "company": null,
25  "blog": "https://atlassian.com/",
26  "location": "Australia",
27  "email": "github@atlassian.com",
28  "hireable": null,
29  "bio": null,
30  "twitter_username": null,
31  "public_repos": 376,
32  "public_gists": 0,
33  "followers": 688,
34  "following": 0,
35  "created_at": "2009-12-16T00:16:42Z",
36  "updated_at": "2023-10-05T04:47:19Z"
37 }
```

Iz ovog odgovora potrebno je izdvojiti i prikazati sljedeće informacije o korisniku:

- avatar_url – URL link na profilnu sliku korisnika
- blog – web stranica
- name – ime korisnika
- location - lokacija korisnika
- bio - detalji korisnika

- created_at – datum kreiranja GitHub računa
- type – tip korisnika

2. Dohvaćanje repozitorija GitHub usera

- Dohvaćanje repozitorija GitHub usera moguće je napraviti GET pozivom na sljedeći URL:
 - <https://api.github.com/users/repos>

Za dohvaćanje repozitorija korisnika s korisničkim imenom facebook, potrebno je napraviti GET poziv na URL <https://api.github.com/users/facebook/repos>.

Odgovor na ovaj API poziv je JSON lista repozitorija navedenog korisnika (kako biste vidjeli puni odgovor s GitHub API-ja, otvorite link u internet pregledniku).

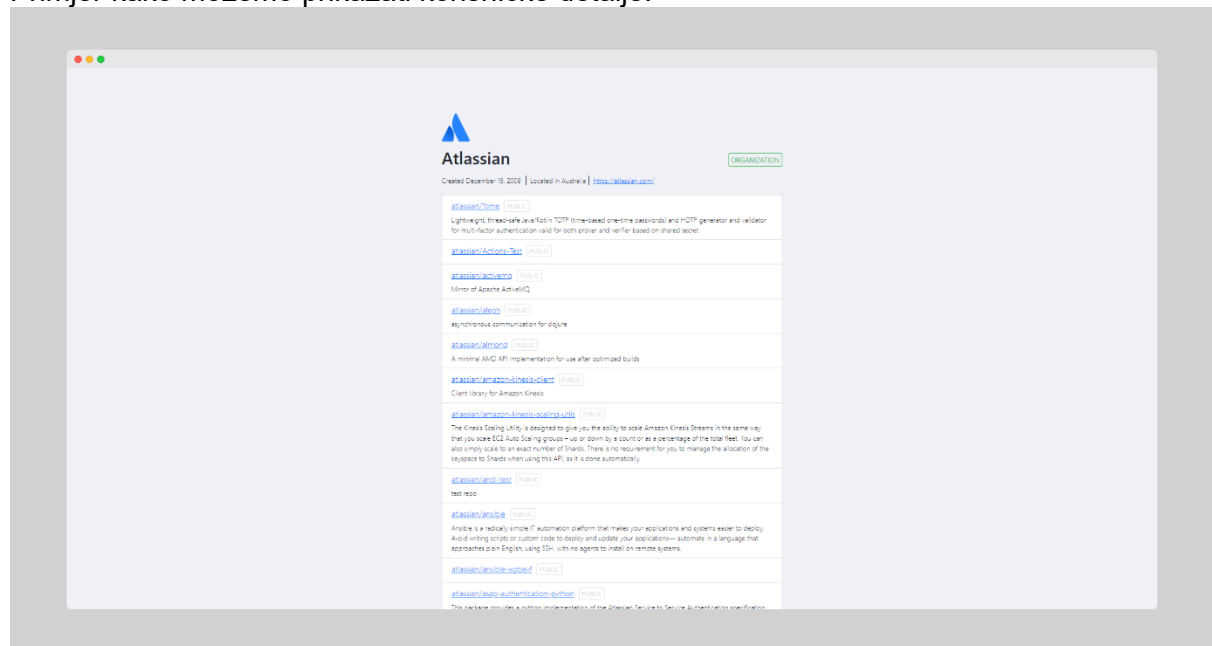
Svaki element liste ima sljedeće bitne attribute:

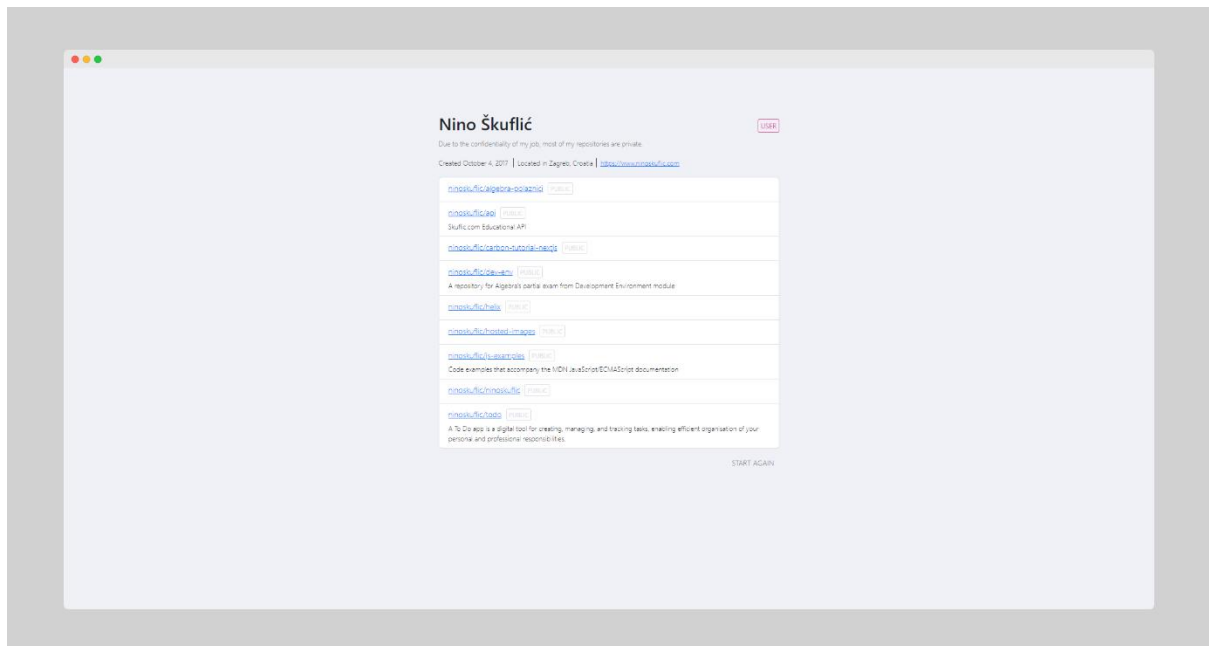
- id – jedinstveni broj koji predstavlja identifikator GitHub repozitorija
- full_name – ime repozitorija
- description – opis repozitorija
- visibility – vidljivost repozitorija (javni ili privatni)

Prikaz detalja unešenog korisnika:

Po unosu korisničkog imena, klikom na gumb, dohvaćaju se s REST API-ja i prikazuju detalji GitHub korisnika.

Primjer kako možemo prikazati korisničke detalje:





Ako je tip repozitorija USER (privatni) nemojte pokazati sliku profila, a ako je tip repozitorija ORGANIZATION (kompanija) pokažite sliku profila.

Kao što je prikazano na prethodnoj slici, potrebno je prikazati sljedeće informacije o GitHub korisniku:

- Profilna slika korisnika
- Ime korisnika
- Lokacija korisnika
- Detalji (bio) korisnika
- Datum kreiranja računa korisnika
- Web adresu korisnika
- Tip korisnika
- Lista koja sadrži imena korisnikovih repozitorija

Na detaljima korisnika, možemo na dno dodati button “START AGAIN”. Klikom na taj button, brišemo trenutno stanje aplikacije i ponovno prikazujemo formu za unos korisničkog imena.

Očekivanja:

- ✓ Kreirati novu aplikaciju pomoću create-react-app naredbe
- ✓ Napraviti više od jedne React komponente
- ✓ Koristiti props za komunikaciju među komponentama
 - definirati propTypes za sve komponente koje primaju propse
- ✓ Složiti komponente u parent-child odnosu
 - ispravno exportati child komponentu
 - ispravno importati child komponentu unutar parent komponente
 -
- ✓ Koristiti state za čuvanje stanja aplikacije
 - Čuvati stanje aplikacije u jednoj Statefull komponenti
 - Ova komponenta je “glavna” komponenta, u njoj čuvamo unešeno korisničko ime, korisnika i repozitorije dohvaćene s API-ja.
 - Ova komponenta ima nekoliko child komponenti kojima kroz propse proslijeđuje vlastito stanje (GitHub korisnika i repozitorije)

- ✓ Koristiti listu i renderirati listu unutar JSX-a (lista repozitorija)
- ✓ Koristiti formu za unos GitHub korisničkog imena
 - Koristiti događaje (evente) - npr. onChange prilikom unošenja GitHub korisničkog imena - npr. onClick za reakciju na klik buttona u formi za unos korisničkog imena

Završeni zadatak vidljiv je na zaslonu projektora / Adobe Connectu.