

Priprema za prvu proveru znanja

1. Kreirati **direktorijum** sa proizvoljnim nazivom.
2. Inicijalizovati **git repozitorijum** u kreiranom direktorijumu.
3. Napraviti **index.html** datoteku sa osnovnom strukturom HTML dokumenta.
4. Napraviti **commit** sa porukom *'Prvi commit'* koji sadrži dosadašnje promene.
5. Napraviti udaljeni repozitorijum na svom GitHub-u i sinhronizovati ga sa ovim lokalnim.
6. Napraviti **novu granu** *'forma-za-unos-izlaska-na-ispit'* i **prebaciti se** na tu granu.
7. U index.html fajlu **napraviti formu** za unos izlaska na ispit sa poljima kao sa date slike:

The screenshot shows a web form with the following fields and controls:

- Ocena:** A text input field containing the value '6'.
- Datum izlaska na ispit:** A date picker showing '03/29/2024'.
- Broj indeksa:** A text input field containing '2022/0043'.
- Rok:** Radio buttons for 'Januar', 'April', 'Jun', and 'Septembar'. 'Septembar' is selected.
- Redni broj izlaska na ispit:** A dropdown menu showing '2'.
- Položio/la:** A checked checkbox.
- Pošalji:** A blue button at the bottom right.

8. Napraviti **commit** sa porukom *'Struktura forme za unos izlaska'* sa dosadašnjim promenama.
9. Vratiti se na master granu i u nju **spojiti** granu *'forma-za-unos-izlaska-na-ispit'*. Sinhronizovati promene sa udaljenim repozitorijumom.
10. Napraviti **novu granu** *'serijalizacija-podataka-forme'* i **prebaciti se** na tu granu.
11. Napraviti **index.js** datoteku i referencirati je iz index.html dokumenta.
12. U index.js datoteci napraviti funkciju **podaciForme()** koja čita sve unete podatke iz forme i vrši validaciju unetih podataka na sledeći način:
 - a) Ocena mora biti u opsegu [5 – 10].
 - b) Broj indeksa mora biti u formatu YYYY/XXXX (godina/broj).
 - c) Godina mora biti veća od 2000, dok broj mora biti u opsegu [1 – 1000].
 - d) Ukoliko je ocena u opsegu [6-10] onda checkbox „Položio/la“ mora biti označen.

Ukoliko neki podatak nije validan neophodno je ispod odgovarajućeg polja prikazati razlog greške crvenim slovima.

Ukoliko su svi podaci validni funkcija vraća JSON objekat u vidu string-a sa sledećom strukturom:

```
{
  "ocena": 6,
  "datumIzlaska": "03/29/2024",
  "brojIndeksa": "2022/0043",
  "rok": "Septembar",
  "redniBrojIzlaska": 2,
  "polozen": true
}
```

13. U index.html datoteci dodati ispod forme **textarea element** sa labelom 'Podaci forme'.
14. U index.js datoteci **registrovati handler funkciju** na dugme 'Pošalji' koja pozove funkciju podaciForme() i njen rezultat (JSON) postavi kao vrednost textarea elementa ispod forme. Nakon uspešnog unosa u textarea element, vratiti sva polja na podrazumevane vrednosti i skloniti greške ispod polja ukoliko postoje.

15. Napraviti **commit** sa porukom *'form submit handler'* sa dosadašnjim promenama. Sinhronizovati promene sa udaljenim repozitorijumom.
16. Vratiti se na master granu i u nju **spojiti** granu *'serijalizacija-podataka-forme'*. Sinhronizovati promene sa udaljenim repozitorijumom.
17. Napraviti **novu granu** *'deserijalizacija-podataka-forme'* i **prebaciti se** na tu granu.
18. Napraviti **deserialize.js** datoteku i referencirati je iz index.html dokumenta.
19. U deserialize.js datoteci napraviti funkciju **popuniFormu()** koja na osnovu unetog *JSON string-a* u okviru textarea elementa, popunjava polja forme sa odgovarajućim podacima.
20. Dodati dugme *Ucitaj* ispod textarea elementa i registrovati handler funkciju koja poziva prethodno definisanu **popuniFormu()** funkciju.
21. Ukoliko podaci u JSON formatu nisu validni ili nedostaje neki od podataka ispisati razlog greske (crvenim slovima) ispod textarea elementa.
22. Napraviti **commit** sa porukom *'ucitavanje podataka forme iz JSON-a'* sa dosadašnjim promenama. Sinhronizovati promene sa udaljenim repozitorijumom.
23. Vratiti se na master granu i u nju **spojiti** granu *'deserijalizacija-podataka-forme'*. Sinhronizovati promene sa udaljenim repozitorijumom.