

1. Ce sunt "props" și la ce folosesc?

Props = argumentele unei componente în React/React Native; practic sunt parametrii pe care îi pasăm unei funcții (functional components) sau unei clase; prin intermediul lor, avem o componentă generalizată pe care o putem folosi în diferite situații, schimbând valoarea props-urilor; de exemplu, o componentă Header poate conține titlul "HomeScreen" și string-urile "Something for home" și "Additional information" pe ecranul principal, iar pe un alt ecran să conțină doar titlul "HomeScreen" și string-ul "Something for profile"; în acest caz am reutilizat componenta cu props-uri diferite (și ca "valori" și ca număr de argumente, deoarece un prop poate fi și optional); ca și cod ar fi:

```
function HeaderComponent(p: {title: string, myString1: string, myString2?: string}) {
  return <View>
    <Text>{p.title}</Text>
    <Text>{p.myString1}</Text>
    <Text>{p.myString2}</Text>
  </View>
}
```

```
export function HomeScreen() {
  return (
    <View>
      <HeaderComponent
        title = "HomeScreen"
        myString1 = "Something for home"
        myString2 = "Additional information"
      />
    </View>
  );
}
```

```
export function ProfileScreen() {
  return(
    <View>
      <HeaderComponent
        title = "ProfileScreen"
        myString1 = "Something for profile"
      />
    </View>
  );
}
```

2. Ce este Redux în React Native?

Redux = o bibliotecă folosită pentru state management cu React, React Native, Angular etc.; se folosește pentru a ține datele într-un store și pentru a avea acces la ele și la stările lor curente, pentru a putea vedea prin ce stări trece un anumit atribut/o componentă la un moment dat, deci e mult mai ușor în debugging să afli ce nu este în regulă; starea din store se actualizează cu ajutorul reducer-ilor.

3. Care sunt componentele Redux folosite în React Native?

Componente Redux:

- Store: o aplicatie are nevoie de un singur store, adica un obiect, in care se va tine starea aplicatiei cu toate componentele ei, toate datele aplicatiei, structurata pe diferite "nivele", asemanator unui arbore
- Actions: datele aplicatiei care sunt puse la dispozitie store-ului; prin intermediul actiunilor stim ce "vrem sa facem", dar nu avem inca implementarea (care se face in reducer)
- Reducers: functii care iau ca parametri starea curenta si/sau o actiune si prin intermediul carora se returneaza o noua stare, updatata; de obicei fiecare componenta are reducer-ul ei

4. Explică noțiunea de AsyncStorage din React Native și dă câteva exemple de cazuri în care ar trebui folosită, dar și cazuri când NU ar trebui folosită.

AsyncStorage = un sistem de stocare a datelor, offline si in mod asincron; spre deosebire de localStorage din cadrul aplicatiilor web, valorile stocate in AsyncStorage trebuie sa fie "asteptate", accesul la ele nefiind instant.

```
Ex: async function getSavedData() {  
    const savedData = await AsyncStorage.getItem("savedData");  
    return savedData;  
}
```

Cazuri indicate:

- Daca se doreste ca daca utilizatorul inchide aplicatia si revine peste o anumita perioada de timp, la login se vor salva in AsyncStorage token-ul de autorizare si alte date ale user-ului necesare pentru ca acesta sa poata folosi aplicatia. Cand aplicatia va reporni va verifica daca in AsyncStorage la cheia respective exista o valoare iar daca exista va face autorizarea automat pe baza datelor din storage. La logout datele din AsyncStorage se sterg.
- In cazul in care aplicatia contine un carusel care este afisat tuturor userilor care descarca, instaleaza si intra in aplicatia pentru prima data, acestia neavand inca un cont, faptul ca au intrat de minim o data in aplicatie se va salva in AsyncStorage pentru a nu le afisa caruselul aplicatiei de fiecare data. Acestia neavand inca un cont la momentul respectiv, aceasta informatie nu poate fi salvata pe cont.
- Alte situatii in care e necesara salvarea anumitor date strict cand userul foloseste device-ul respectiv

Cazuri neindicate:

- Daca in cadrul aplicatiei exista anumite actiuni limitate la un numar specific(ex: versiunea free permite utilizatorului adaugarea de maximum 2 itemi intr-o lista oarecare, in timp ce versiunea

payed permite utilizatorului adaugarea de itemi nelimitati). In acest caz aceasta informatie trebuie sa se fie salvata in baza de date in cadrul user-ului respectiv si nu in AsyncStorage pentru a preveni cazul in care un utilizator adauga cate 2 itemi, dupa care isi sterge din setarile telefonului "app's data & cache" iar dupa aceea reintra in aplicatie si poate adauga alti 2 itemi in lista pentru ca datele din AsyncStorage au fost sterse.

- Alte situatii in care datele pot fi salvate intr-o baza de date si nu depind de device-ul respectiv.