

Android Development - Dry Exercise 1

Tal Dahan

Part I

1. The 2 lines that make the list infinitely scrolling are:

```
if (index >= _suggestions.length) {  
  _suggestions.addAll(generateWordPairs().take(10)); }
```

When the index reaches the length of the `_suggestion` list, we generate 10 more words and add them to the `_suggestion` list, that way we have 10 more words to display.

The item builder keeps building new items as long as we don't supply an `ItemCount` parameter. So when we remove those lines, we get an Out Of Range error:

```
RangeError (index): Invalid  
value: Not in inclusive  
range 0..29: 30  
See also: https://  
flutter.dev/docs/testing/  
errors
```

2. Another way of creating a list with separators is by using `ListView.separated`, which creates a `ListView` with 2 indexed builders: one of the items of the list, and one for the separators. According to the documentation, this constructor is appropriate when the list is of finite length. In my opinion, if the list is finite using the `ListView.separated` constructor is a better way of adding a divider between each item, since we don't have to manipulate the indexes and use odd indexes for separators and even indexes for items- this constructor can do it for us. this makes the code more readable and organized.

3. The `ListView.Builder`'s `ItemBuilder` contains a `setState()` call in the `onTap` handler since we are required to change the UI that is displayed.

When tapped, we expect the icon and its color to change. `setState()` calls the `build` method of the stateful widget again, which "draws" all the items of the list again- with the updated values. In this case, the `"alreadySaved"` variable will change its value, therefore the color and icon will now be different, and since the `build` method was called again, the UI will update.

Part II

1. The `MaterialApp` widget's purpose is to wrap a number of widgets which are usually required when using a Material UI design.

Some of its properties are:

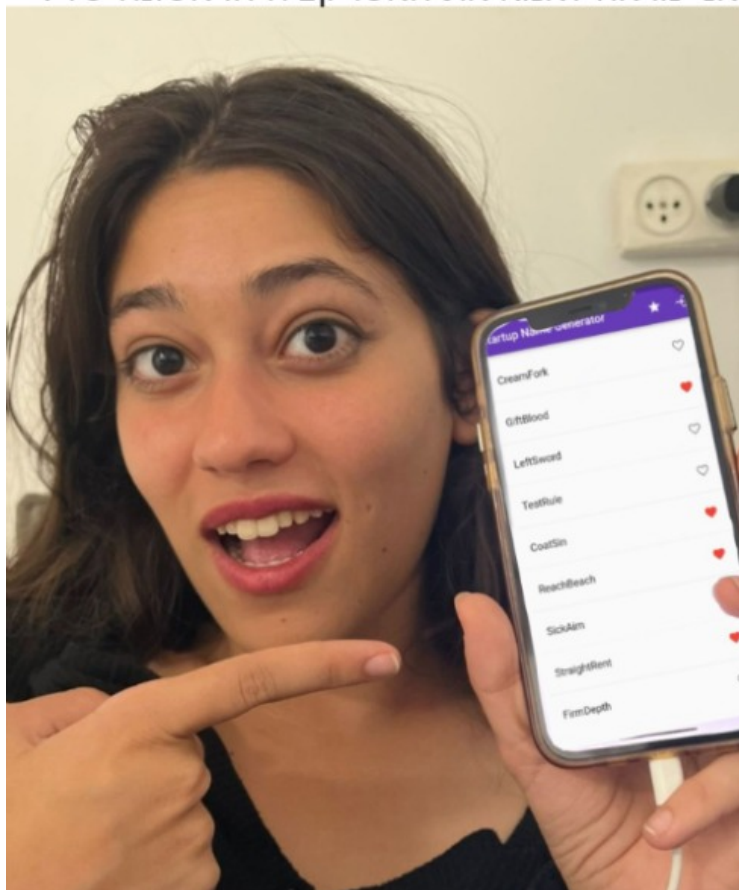
- home: defines the default route of the app. This route will be the first widget to show on screen when the app is opened.
- theme: defines default visual properties such as the app's colors.
- title: defines a one-lined description of the app which is used by the device to identify the app.

2. The `key` property in the `Dismissible` widget helps flutter control how one widget replaces another widget in the tree. Since the list contains many dismissible widgets, When dismissed, flutter needs a way to identify which of those is the one that has to be removed from the list.

כשכתוב שאפשר להוסיף אקסטרה UI Features מעבר לנדרש



אני מראה לאמא את האפליקציה הראשונה שלי:





CTRL+ALT+L

הקוד שלי:

