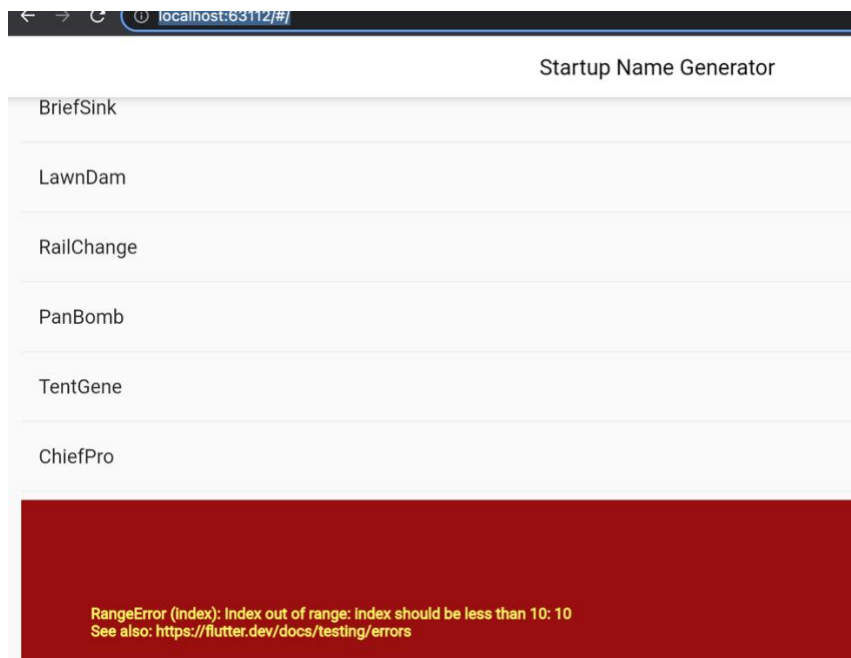# Project In Android Development 236271- Assignment1

Question 1

The lines that enable "infinite-scrolling" are:

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

If these lines were removed, scrolling past the 10th word on the list would trigger an "index out of range" error as follows:

Question 2

An additional option for constructing a list is a default construct which receives and explicit list<Widget> of children. This method eagerly constructs the entire list, not only the visible items like the option used in our app.

If the list is finite and contains only 100 items, we'd prefer this method to the original one as it would enable us to remove the runtime overhead of creating additional items in exchange for an increase in compilation time.

Question 3

As explained in the workshop, our app is utilizing the reactive programming paradigm. When the user taps the widget, the state of the widget changes. More specifically, when tapping on a word pair in the **RandomWords** list, the user expects the words to be added/removed from the favorite list and changes the favorite icon in the list to change. As such calling **setState()** communicates that the **RandomWords** widget's state has changed which causes the framework to call _ **RandomWordsState's build()** method to update the widget.

Exercise 2

Question 1

The MaterialApp widget wraps several widgets that are commonly used for material design applications. We will expand on some of them:

1. MaterialApp configures the Navigator which is used to traverse the different routes of the app
2. MaterialApp defines a home for the app which is the default route and normally the first screen displayed when the app starts up (unless InitalRoute is defined)
3. MaterialApp can also set a shared theme across the entire app

Question 2

When a dismissible widget is swiped some action needs to be performed on it, typically removal from a list. As such we need to remove or update its node in the app's widget tree. In order to find this node, we need a key to find the widget in the tree. Hence the required key property for a dismissible widget.