1. To capture Future's return value, we have the following options:

a. Using then()
b. Using pushNamed()
c. Using async/await
d .Using push()

2. What are some approaches to state management?

There are several approaches to state management. The simplest is using setState() within a StatefulWidget to manage local state changes within a single widget. For more complex apps, Provider package offers a lightweight solution for managing app-wide state. Another popular option is the BLoC (Business Logic Component) pattern, where UI components interact with streams of data. Riverpod is also a modern state management library offering a simple yet powerful alternative to Provider, with support for both Flutter and non-Flutter applications.

3. It is strongly suggested to open SQLite database as many times as needed (multiple instances).
- False

4. Select all the supported data types in SharedPreferences

a.List<double>
b.List<int>
c.String
d.double
e.List<String>
f.bool
g.int

5. Future must not be created during State.build or StatelessWidget.build method call when constructing the FutureBuilder.
- True

6. Timer is a class that represents a count-down timer and it can be fired
a.once

b.repeatedly
c.once or repeatedly

7.In Flutter, state management can be categorized into how many types ?
a.2
b.3
c.4

8.The [app] state can be distributed across multiple areas of your app and the same is maintained with user sessions.

9.And keyboardType property set it to TextinputType.multiline, so that user will get a button using which he/she can move the cursor to next line.

10.User authentication and app authentication are the same thing.
- False