Summary and Highlights: Advanced Flutter

Congratulations! You have completed this module. At this point in the course, you know:

- The steps to use plugins in Flutter include finding the plugins in pub.dev, adding suitable plugins to your project, importing the plugin into your Dart code, and using the imported plugin.
- To manage plugin compatibility, consider managing version constraints, resolving dependency conflicts, and testing on multiple platforms.
- Native mobile features are functionalities provided by a device's operating system.
- Flutter apps use plugins to access device cameras, GPS, and Bluetooth, while native mobile plugins use platform-specific APIs.
- State refers to data utilized by Flutter to create the user interface. The two main types of state in Flutter are ephemeral state and app state.
- Implementing state management techniques is primarily intended to separate the business logic from the UI and optimize performance to its fullest potential.
- You can create your first API by sending a request to the server, parsing the JSON data, and updating the UI.
- Use error-handling techniques such as try-catch blocks, timeouts, and user messages to identify and manage errors.
- Use data persistence to ensure that data remains accessible long after the app is shut down or the device is powered off.
- Set up your storage method, perform CRUD operations, encode and decode data in JSON format, and integrate with state management solutions for efficient local storage.

Mark as completed