



CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF TECHNOLOGY AND ENGINEERING
CHANDUBHAI S. PATEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING



Subject : Mobile Application Development

Subject Code: AIML308

NAME : CHOKSI OM CHIRAGBHAI

Semester: 5

Academic Year :2025-26(ODD)

ID: 23AIML010

Practical 6

Problem Definition

You are building a mobile application where users navigate through multiple screens like login, dashboard, and profile. Create a Notes App with persistent storage using Shared Preferences.

Supplementary Problems -	Dark mode toggle, Remember me feature
--------------------------	---------------------------------------

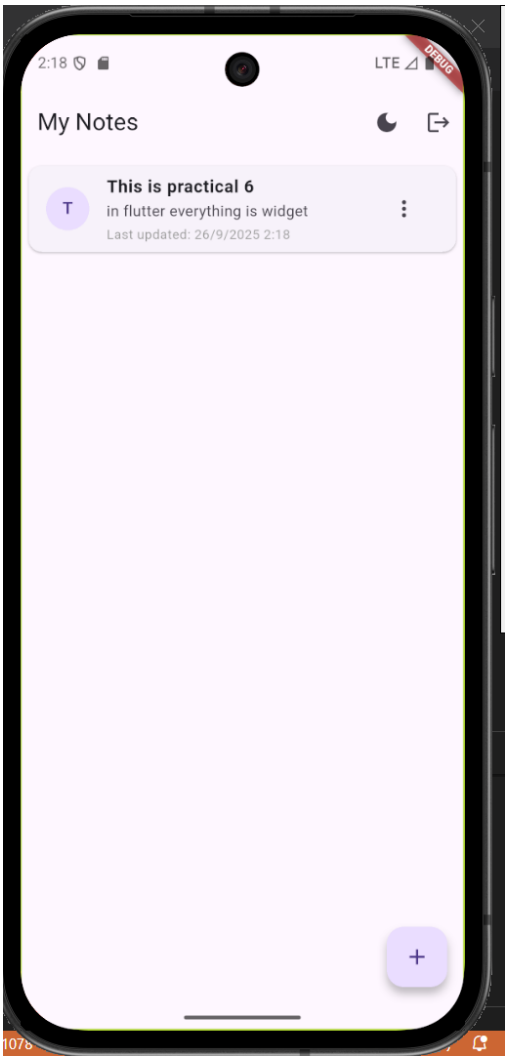
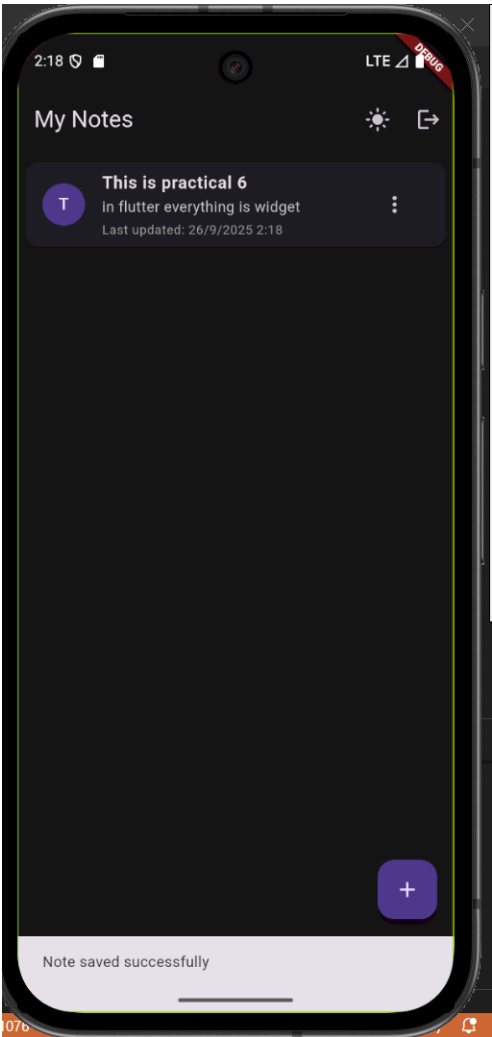
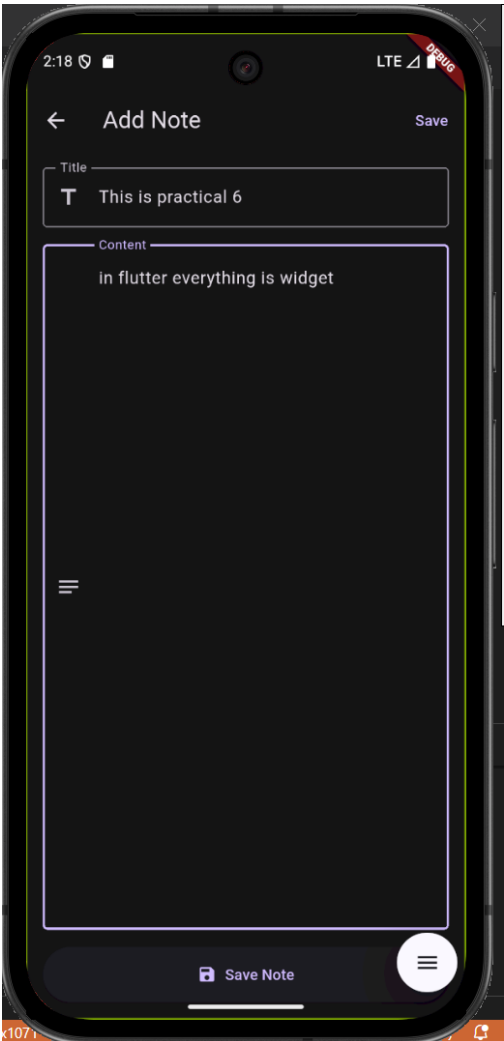
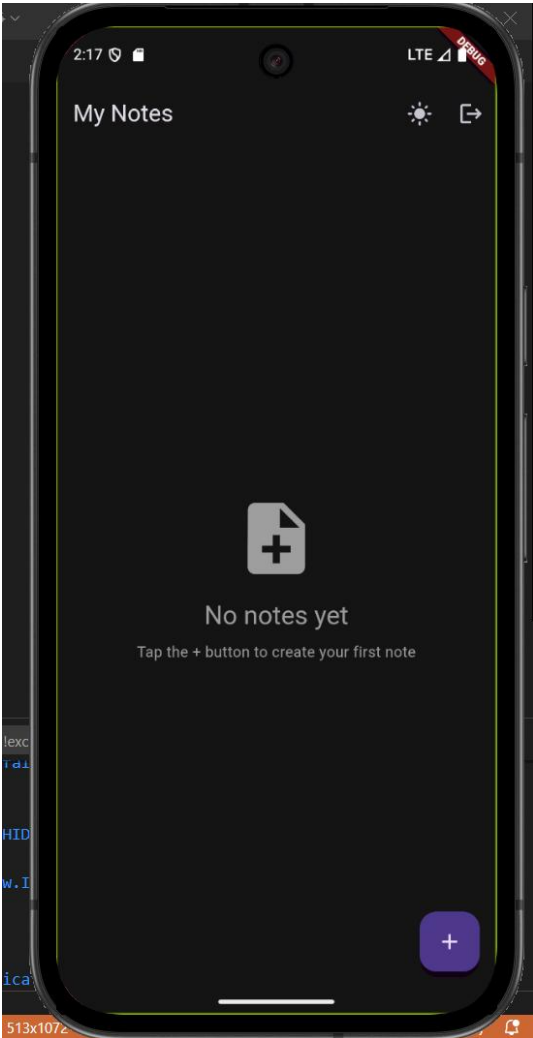
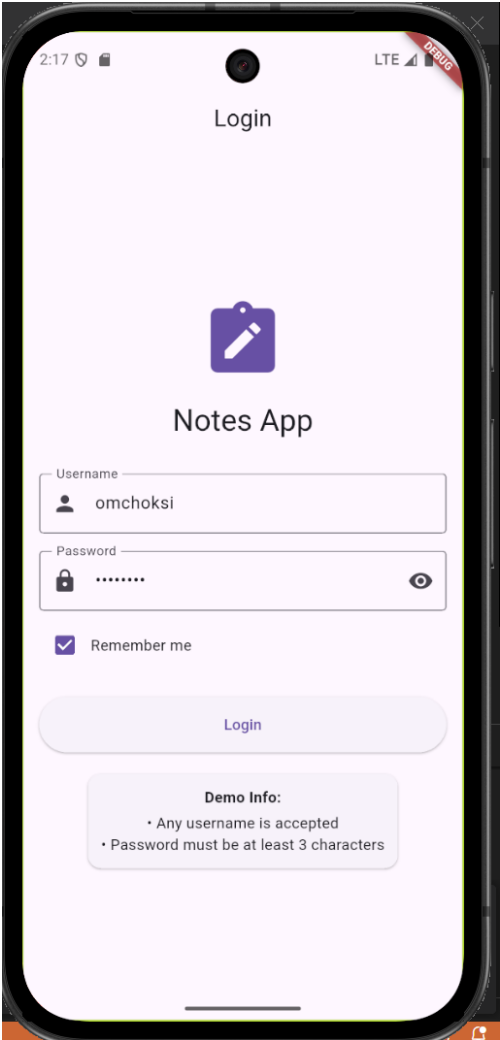
Technical Approach:

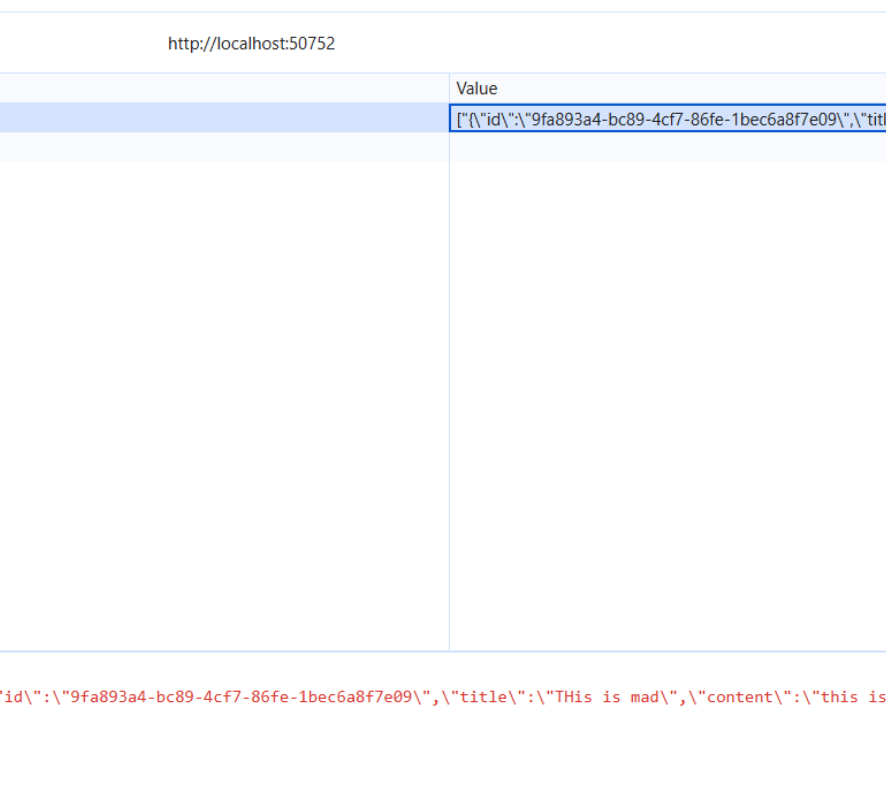
This Flutter application implements a notes management system using SharedPreferences for local data persistence across app sessions. The core logic revolves around StatefulWidget for managing UI state changes with setState, SharedPreferences for storing user preferences and notes data, and JSON encoding/decoding for complex data serialization. Key widgets include ListView.builder for displaying the notes list, Form with TextFormField for input validation, and Material Design components for consistent theming and navigation between screens.

File Structure:

```
lib/  
├── main.dart           # Application entry point with theme management  
├── models/  
│   └── note.dart      # Note data model with JSON serialization  
├── screens/  
│   ├── login_screen.dart  # Login interface with remember me functionality  
│   ├── dashboard_screen.dart  # Notes list with dark mode toggle  
│   └── add_edit_note_screen.dart # Create/edit notes interface  
└── services/  
    └── preferences_service.dart # SharedPreferences service layer
```

Screenshots:





http://localhost:50752

Origin	http://localhost:50752
Key	Value
flutter.notes	<pre>{\id\:"9fa893a4-bc89-4cf7-86fe-1bec6a8f7e09\","title\:"This is mad\","content\:"this is mad\","created\:"1640000000\"}</pre>

▼ [,...]

```
0: "{\id\:"9fa893a4-bc89-4cf7-86fe-1bec6a8f7e09\","title\:"This is mad\","content\:"this is mad\","created\:"1640000000\"}"
```

```

  ▾ OUTLINE
    ▾ main () → void
      MyApp
    ▾ MyApp
      MyApp ({super.key})
      createState () → State<MyApp>
    ▾ _MyAppState
      _isDarkMode → bool
      _isLoggedIn → bool
      _isLoading → bool
      initState () → void
      _loadAppState () → Future<void>
      _toggleTheme (bool isDark) → void
      _onLoginStateChanged (bool isLoggedIn) → void
    ▾ build (BuildContext context) → Widget
      ▾ MaterialApp
        ▾ Scaffold
          ▾ Center
            CircularProgressIndicator
      ▾ MaterialApp
        DashboardScreen
        LoginScreen

```

[illegible]

Key Questions:**1. When to use shared preferences?.**

Ans: Use SharedPreferences for storing lightweight data that needs to persist across app sessions, such as user preferences (theme, language), session tokens, simple configuration flags, and small amounts of structured data under 1MB, as implemented in this notes app for theme settings, login state, and user credentials

2. What data types can be stored?

Ans: SharedPreferences supports primitive data types including booleans, strings, integers, doubles, and string lists, as demonstrated in the app where boolean values store theme preferences and login state, strings store usernames/passwords, and string lists store JSON-encoded notes data.

3. How to persist small data?

Ans: Small data is persisted by obtaining a SharedPreferences instance, using appropriate setter methods (setBool, setString, setStringList) for different data types, and retrieving values with corresponding getter methods, ensuring data survives app restarts as shown in the PreferencesService class for theme, login, and notes storage.

Key Skills to be Understand:

Local storage, Preference handling

Implement local DB solutions

Store app settings or session