Amazon Clone Project – Flutter

Custom Features Implemented

Summer of Code 2025

Goru Gokul

Flutter features applied using dart

As part of the Summer of Code 2025 Amazon Clone project using Flutter, I have implemented three additional features that go beyond the original tutorial playlist. These features enhance the UI, interactivity, and persistence of the app.

1. Feature 1: Theme Toggle (Light/Dark Mode)

Description

This feature allows users to dynamically switch between light and dark themes from the AppBar. The implementation is based on a boolean state that controls the theme of the entire MaterialApp. This adds a modern UI touch and improves accessibility, especially in varied lighting environments.

Purpose

The purpose of the theme toggle is to enhance the visual experience for users by allowing them to choose between light and dark modes based on their environment and personal preference. It adds to the professional look and accessibility of the application, making it more user-centric, especially for those using the app in low-light conditions or with vision sensitivity.

Code Snippet

```
bool _darkMode = false;

void toggleTheme() {
  setState(() {
    _darkMode = !_darkMode;
  });
}

theme: _darkMode ? ThemeData.dark() : ThemeData.light(),
```

—

2. Feature 2: Add New Quote Functionality

Description

This feature provides a user-friendly input form that lets users contribute their own quotes. It contains two text fields—for the quote content and the author's name—and an action button. When submitted, the quote is added to the UI and stored for future access, increasing user engagement.

Purpose

This feature empowers users to personalize the content by submitting their own quotes, enhancing engagement and interactivity. It demonstrates how to build dynamic, real-time updates to the interface using text fields and input validation, forming a critical foundation for more advanced form-based features in Flutter applications.

Code Snippet

```
TextEditingController _textController = TextEditingController();
TextEditingController _authorController = TextEditingController();

ElevatedButton.icon(
  onPressed: () {
    final newQuote = Quote(
      author: _authorController.text.trim(),
      text: _textController.text.trim(),
    );
    addQuote(newQuote);
  },
  icon: Icon(Icons.add),
  label: Text(’Add Quote’),
)
```

3. Feature 3:Animated Addition and Deletion of Quote Cards

Description

The app uses Flutter's `AnimatedList` and `SizeTransition` widgets to smoothly animate the insertion and removal of quote cards. When a new quote is added, it slides into view. Similarly, deletions animate as the quote smoothly collapses and fades out.

Purpose

This feature significantly improves user experience by adding polished, responsive animations to interactions that affect the UI. It communicates feedback clearly and makes transitions between states feel smooth and natural, contributing to a more professional and engaging app interface.

Code Snippet

```
_listKey.currentState?.insertItem(0);

_listKey.currentState?.removeItem(
  index,
  (context, animation) => SizeTransition(
    sizeFactor: animation,
    child: QuoteCard(quote: removedQuote, delete: () {}),
  ),
);
```