Name: Sufyan Ahmed Mughal

SAP: 55766

Section: 5-2

Subject: Mobile Application Development

QUIZ – 3

```dart
import 'package:flutter/material.dart';

void main() => runApp(const FlashcardQuizApp());

class FlashcardQuizApp extends StatelessWidget {
  const FlashcardQuizApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flashcard Quiz',
      theme: ThemeData(
        primarySwatch: Colors.deepPurple,
        useMaterial3: true,
      ),
      home: const QuizScreen(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class Flashcard {
  final String id;
  final String question;
  final String answer;
  bool isLearned;

  Flashcard({
    required this.id,
    required this.question,
    required this.answer,
    this.isLearned = false,
  });
}

class QuizScreen extends StatefulWidget {
  const QuizScreen({Key? key}) : super(key: key);

  @override
  State<QuizScreen> createState() => _QuizScreenState();
}

class _QuizScreenState extends State<QuizScreen> {
```

```
  final GlobalKey<AnimatedListState> _listKey =
GlobalKey<AnimatedListState>();
  List<Flashcard> _cards = [];
  int _questionCounter = 1;

  @override
  void initState() {
    super.initState();
    _initializeCards();
  }

  void _initializeCards() {
    _cards = [
      Flashcard(
        id: '1',
        question: 'What is Flutter?',
        answer: 'An open-source UI toolkit by Google for building natively
compiled applications.',
      ),
      Flashcard(
        id: '2',
        question: 'What language does Flutter use?',
        answer: 'Dart programming language.',
      ),
      Flashcard(
        id: '3',
        question: 'What is a Widget in Flutter?',
        answer: 'The basic building block of Flutter UI, describing what the
view should look like.',
      ),
      Flashcard(
        id: '4',
        question: 'What is StatefulWidget?',
        answer: 'A widget that has mutable state and can rebuild when the
state changes.',
      ),
      Flashcard(
        id: '5',
        question: 'What is setState()?',
        answer: 'A method that notifies the framework that the internal state
has changed.',
      ),
    ];
```

```
    _questionCounter = _cards.length + 1;
  }

  int get _learnedCount => _cards.where((c) => c.isLearned).length;

  Future<void> _refreshCards() async {
    await Future.delayed(const Duration(milliseconds: 800));
    setState(() {
      _initializeCards();
    });
  }

  void _markAsLearned(int index) {
    final card = _cards[index];
    setState(() {
      card.isLearned = true;
    });
    _listKey.currentState?.removeItem(
      index,
      (context, animation) => _buildCardItem(card, animation, index),
      duration: const Duration(milliseconds: 300),
    );
    setState(() {
      _cards.removeAt(index);
    });

    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: const Text('Card marked as learned!'),
        duration: const Duration(seconds: 1),
        behavior: SnackBarBehavior.floating,
      ),
    );
  }

  void _addNewCard() {
    final newCard = Flashcard(
      id: DateTime.now().toString(),
      question: 'New Question #$_questionCounter',
      answer: 'Answer to question #$_questionCounter. Add your own content
here!',
    );
```

```
    setState(() {
      _cards.insert(0, newCard);
      _questionCounter++;
    });
    _listKey.currentState?.insertItem(0);

    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: const Text('New card added!'),
        duration: const Duration(seconds: 1),
        behavior: SnackBarBehavior.floating,
      ),
    );
  }

  Widget _buildCardItem(Flashcard card, Animation<double> animation, int
index) {
    return SizeTransition(
      sizeFactor: animation,
      child: Dismissible(
        key: Key(card.id),
        direction: DismissDirection.endToStart,
        onDismissed: (direction) {
          _markAsLearned(index);
        },
        background: Container(
          alignment: Alignment.centerRight,
          padding: const EdgeInsets.only(right: 20),
          margin: const EdgeInsets.symmetric(horizontal: 16, vertical: 8),
          decoration: BoxDecoration(
            color: Colors.green,
            borderRadius: BorderRadius.circular(16),
          ),
          child: const Icon(Icons.check_circle, color: Colors.white, size:
32),
        ),
        child: FlashcardItem(card: card),
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
```

```
    return Scaffold(
      body: RefreshIndicator(
        onRefresh: _refreshCards,
        child: CustomScrollView(
          slivers: [
            SliverAppBar(
              expandedHeight: 140,
              floating: false,
              pinned: true,
              backgroundColor: Colors.deepPurple,
              flexibleSpace: FlexibleSpaceBar(
                title: Text(
                  'Learned: $_learnedCount',
                  style: const TextStyle(fontSize: 18, fontWeight:
FontWeight.bold),
                ),
                background: Container(
                  decoration: BoxDecoration(
                    gradient: LinearGradient(
                      colors: [Colors.deepPurple,
Colors.deepPurple.shade300],
                      begin: Alignment.topLeft,
                      end: Alignment.bottomRight,
                    ),
                  ),
                  child: Center(
                    child: Column(
                      mainAxisAlignment: MainAxisAlignment.center,
                      children: [
                        const SizedBox(height: 40),
                        const Icon(Icons.lightbulb_outline, size: 40, color:
Colors.white70),
                        const SizedBox(height: 8),
                        Text(
                          '${_cards.length} cards remaining',
                          style: const TextStyle(color: Colors.white70,
fontSize: 14),
                        ),
                      ],
                    ),
                  ),
                ),
              ),
```

```
            ),
            SliverToBoxAdapter(
              child: Padding(
                padding: const EdgeInsets.all(16),
                child: Text(
                  'Swipe left to mark as learned',
                  style: TextStyle(
                    color: Colors.grey.shade600,
                    fontSize: 14,
                    fontStyle: FontStyle.italic,
                  ),
                  textAlign: TextAlign.center,
                ),
              ),
            ),
            SliverPadding(
              padding: const EdgeInsets.only(bottom: 80),
              sliver: AnimatedList(
                key: _listKey,
                initialItemCount: _cards.length,
                itemBuilder: (context, index, animation) {
                  return _buildCardItem(_cards[index], animation, index);
                },
              ),
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton.extended(
        onPressed: _addNewCard,
        icon: const Icon(Icons.add),
        label: const Text('Add Card'),
        backgroundColor: Colors.deepPurple,
      ),
    );
  }
}

class FlashcardItem extends StatefulWidget {
  final Flashcard card;

  const FlashcardItem({Key? key, required this.card}) : super(key: key);
```

```dart
  @override
  State<FlashcardItem> createState() => _FlashcardItemState();
}

class _FlashcardItemState extends State<FlashcardItem> {
  bool _showAnswer = false;

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: () {
        setState(() {
          _showAnswer = !_showAnswer;
        });
      },
      child: AnimatedContainer(
        duration: const Duration(milliseconds: 300),
        margin: const EdgeInsets.symmetric(horizontal: 16, vertical: 8),
        padding: const EdgeInsets.all(20),
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: _showAnswer
                ? [Colors.green.shade400, Colors.green.shade600]
                : [Colors.blue.shade400, Colors.blue.shade600],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
          borderRadius: BorderRadius.circular(16),
          boxShadow: [
            BoxShadow(
              color: Colors.black.withOpacity(0.1),
              blurRadius: 10,
              offset: const Offset(0, 4),
            ),
          ],
        ),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Row(
              children: [
                Icon(
```

```
                _showAnswer ? Icons.check_circle_outline :
Icons.help_outline,
                color: Colors.white,
                size: 28,
              ),
              const SizedBox(width: 12),
              Expanded(
                child: Text(
                  _showAnswer ? 'Answer' : 'Question',
                  style: const TextStyle(
                    color: Colors.white70,
                    fontSize: 14,
                    fontWeight: FontWeight.w600,
                  ),
                ),
              ),
            ],
          ),
          const SizedBox(height: 16),
          AnimatedCrossFade(
            firstChild: Text(
              widget.card.question,
              style: const TextStyle(
                color: Colors.white,
                fontSize: 18,
                fontWeight: FontWeight.bold,
                height: 1.4,
              ),
            ),
            secondChild: Text(
              widget.card.answer,
              style: const TextStyle(
                color: Colors.white,
                fontSize: 16,
                height: 1.5,
              ),
            ),
            crossFadeState: _showAnswer
                ? CrossFadeState.showSecond
                : CrossFadeState.showFirst,
            duration: const Duration(milliseconds: 300),
          ),
          const SizedBox(height: 12),
```

```
            Text(
              _showAnswer ? 'Tap to see question' : 'Tap to reveal answer',
              style: const TextStyle(
                color: Colors.white60,
                fontSize: 12,
                fontStyle: FontStyle.italic,
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```