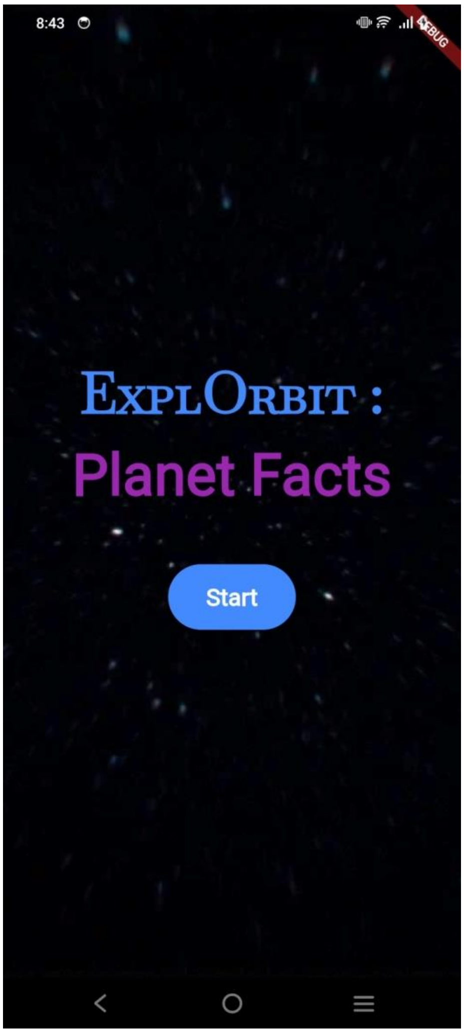
ExplOrbit: Planet Facts

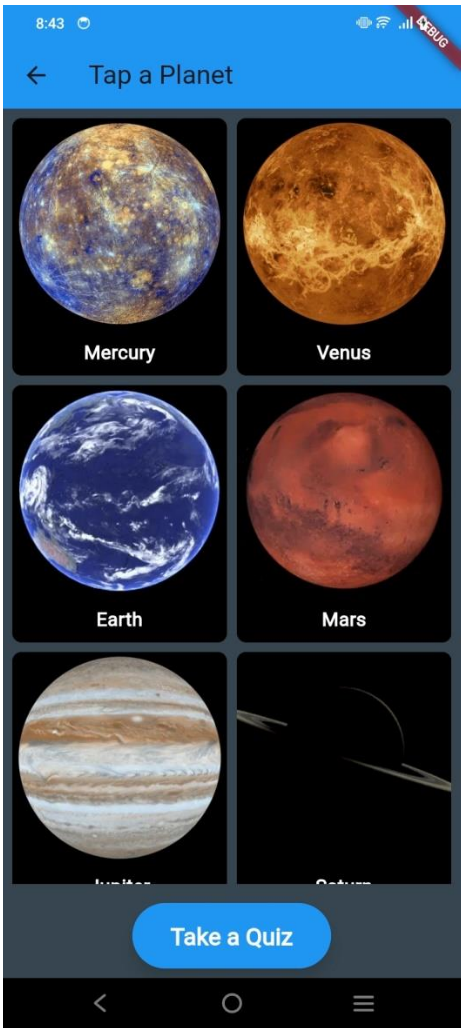
Altamia, Analyn

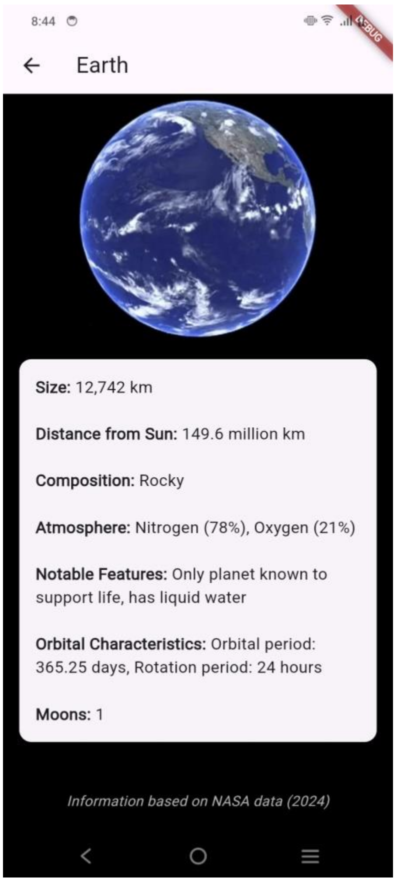
Arenal, Lorenz

Pagkaliwangan, Jarwin

Screenshot of Mobile Application







Size: 12,742 km

Distance from Sun: 149.6 million km

Composition: Rocky

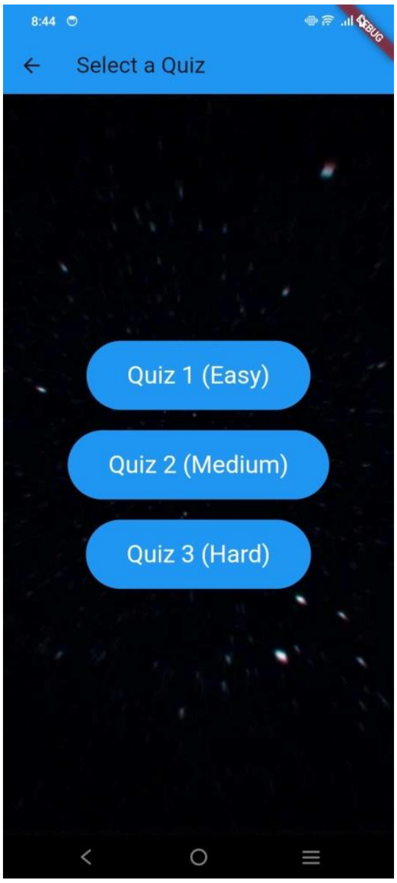
Atmosphere: Nitrogen (78%), Oxygen (21%)

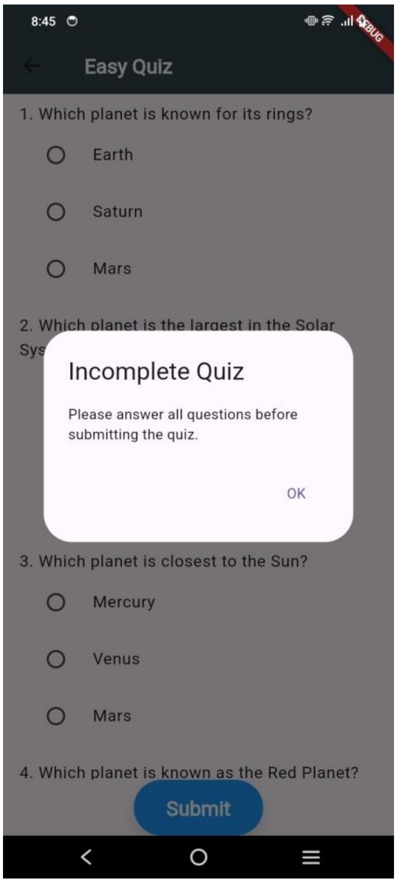
Notable Features: Only planet known to support life, has liquid water

Orbital Characteristics: Orbital period: 365.25 days, Rotation period: 24 hours

Moons: 1

Information based on NASA data (2024)





1. Which planet is known for its rings?

○ Earth

○ Saturn

○ Mars

2. Which planet is the largest in the Solar Sys

Incomplete Quiz

Please answer all questions before submitting the quiz.

OK

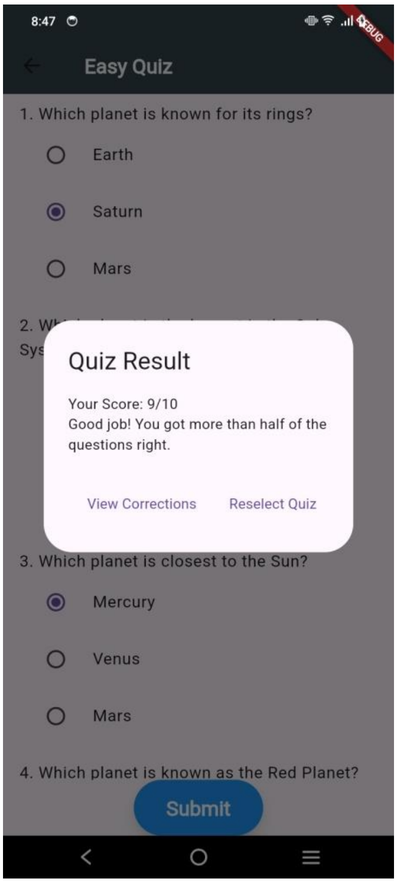
3. Which planet is closest to the Sun?

○ Mercury

○ Venus

○ Mars

4. Which planet is known as the Red Planet?



1. Which planet is known for its rings?

○ Earth

● Saturn

○ Mars

2. Wh

Sys

Quiz Result

Your Score: 9/10
Good job! You got more than half of the questions right.

View Corrections Reselect Quiz

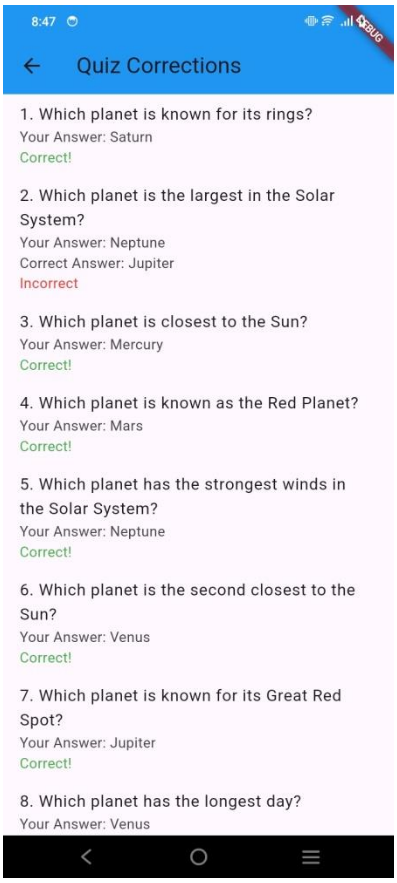
3. Which planet is closest to the Sun?

● Mercury

○ Venus

○ Mars

4. Which planet is known as the Red Planet?



8:47

Quiz Corrections

1. Which planet is known for its rings?
Your Answer: Saturn
Correct!

2. Which planet is the largest in the Solar System?
Your Answer: Neptune
Correct Answer: Jupiter
Incorrect

3. Which planet is closest to the Sun?
Your Answer: Mercury
Correct!

4. Which planet is known as the Red Planet?
Your Answer: Mars
Correct!

5. Which planet has the strongest winds in the Solar System?
Your Answer: Neptune
Correct!

6. Which planet is the second closest to the Sun?
Your Answer: Venus
Correct!

7. Which planet is known for its Great Red Spot?
Your Answer: Jupiter
Correct!

8. Which planet has the longest day?
Your Answer: Venus

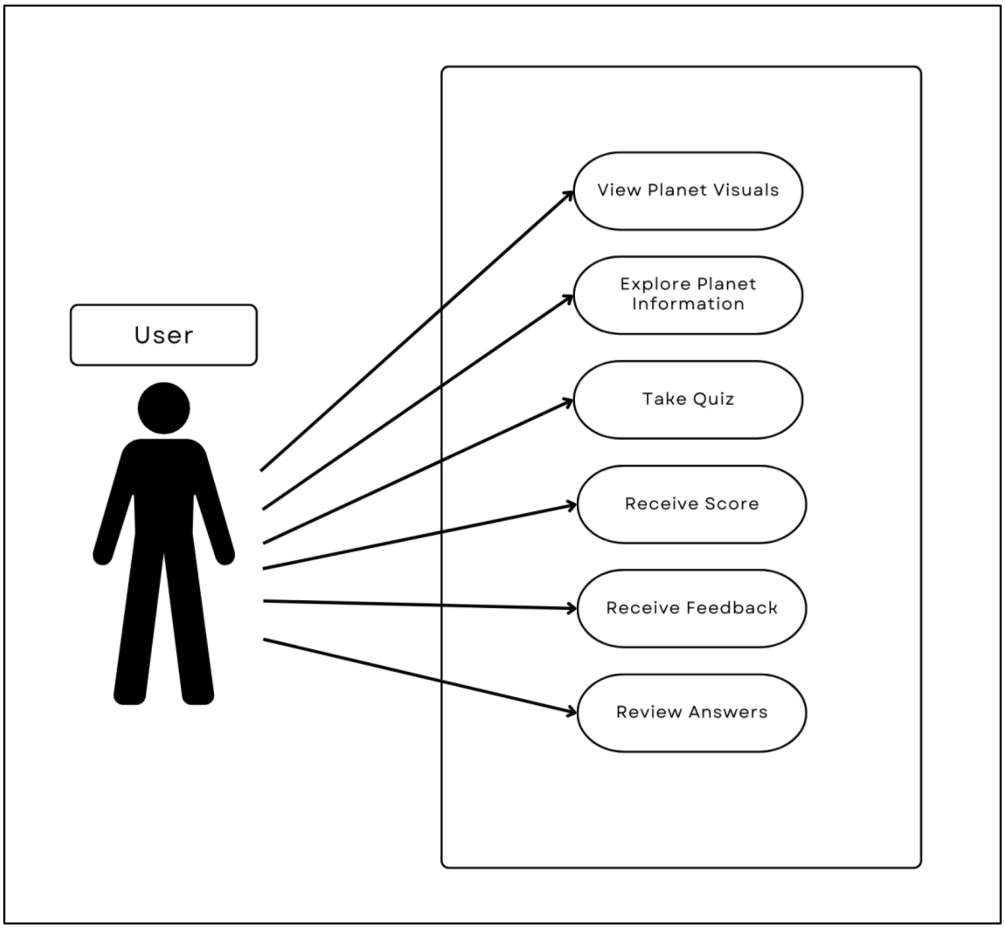
Summary of Functions

The app is designed to make learning about planets easy and engaging. It displays planets in a grid format, allowing users to explore each one and view important details such as their size, distance from the Sun, and other unique features. The app includes a quiz with questions at various difficulty levels so users can test their knowledge. Instant scores and feedback are provided after each quiz, helping users understand what they know and where they need to improve. Additionally, users can review which answers were correct or incorrect, making it a useful tool for reinforcing and expanding their knowledge of the planets in our solar system.

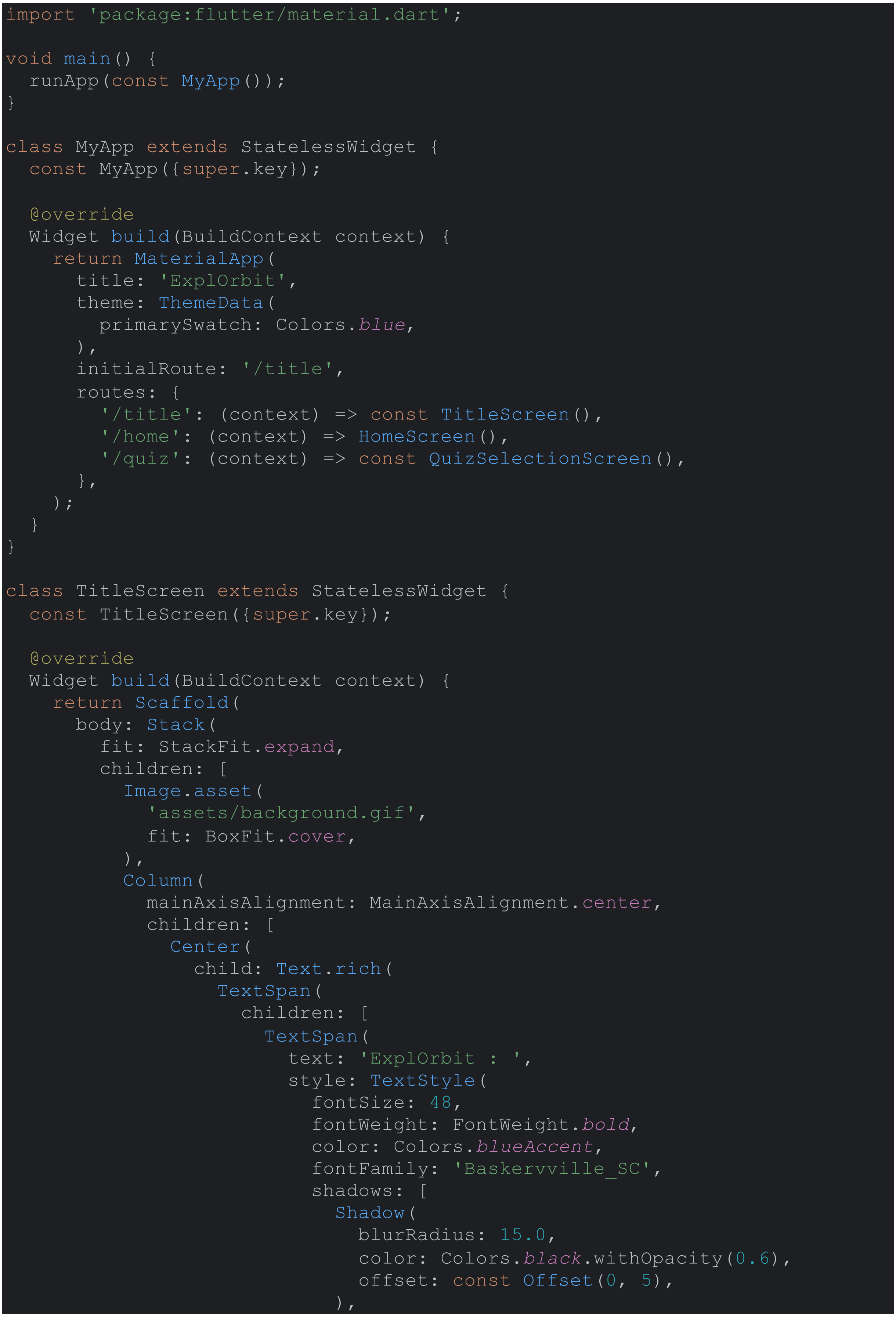
1. **Planet Selector & Visuals** - Explores different planets with visuals through a grid view.
2. **Exhibiting Information** - Displays detailed data on each planet, including size, distance from the Sun, composition, atmosphere, notable features, and orbital characteristics.

3. **Quiz** - Offers multiple-choice questions about planetary facts with difficulty levels (easyt, medium, and hard).
4. **Mandatory Completion Check** - Ensures users cannot submit a quiz or form until all required fields or questions are completed.
5. **Scoring and Feedback** - Scores and feedback are updated instantly as users answer questions or complete the quiz.
6. **Review & Corrections** - Displays option for viewing correct and incorrect answers to enhance learning.

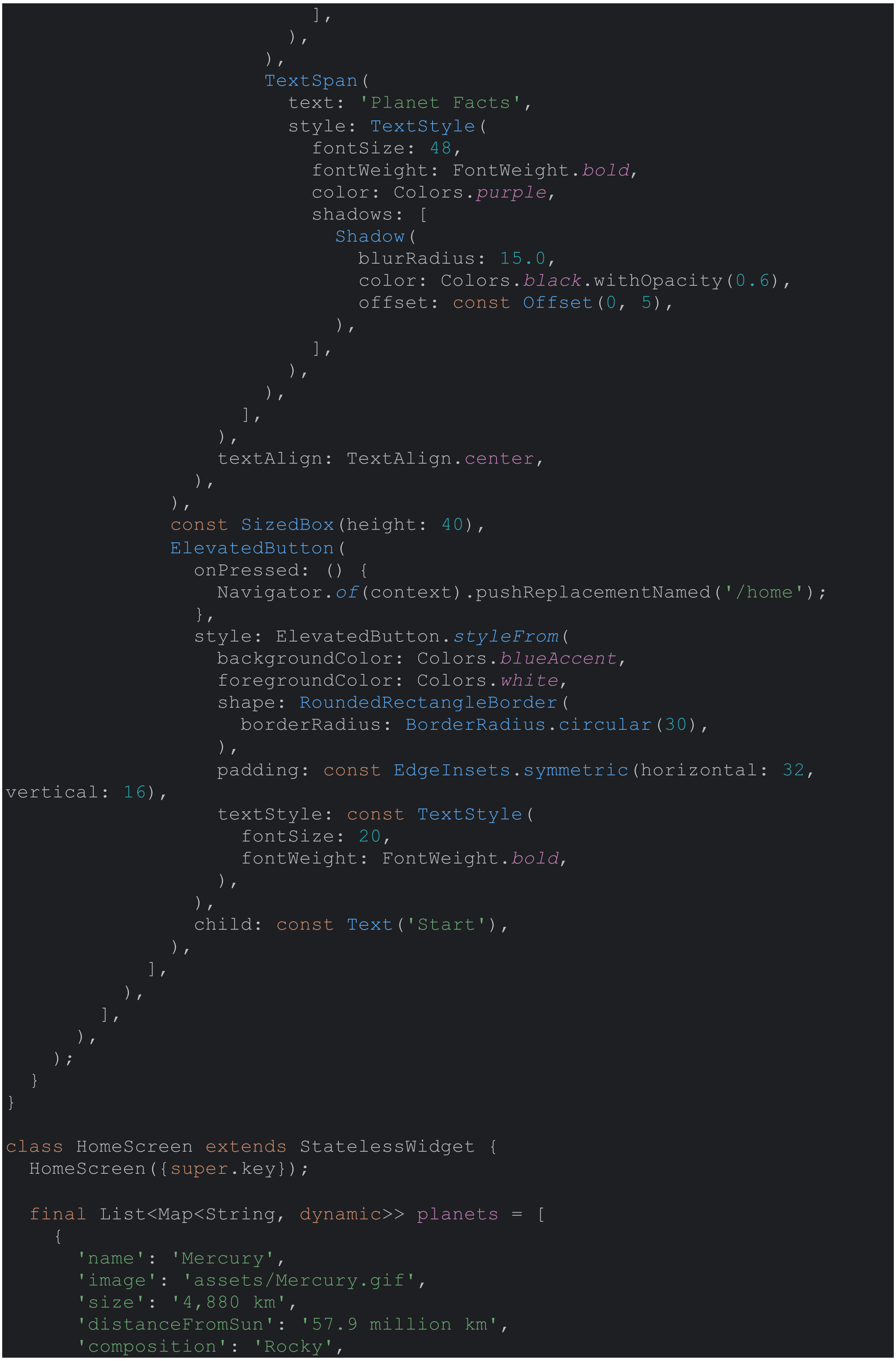
Use-case Diagram



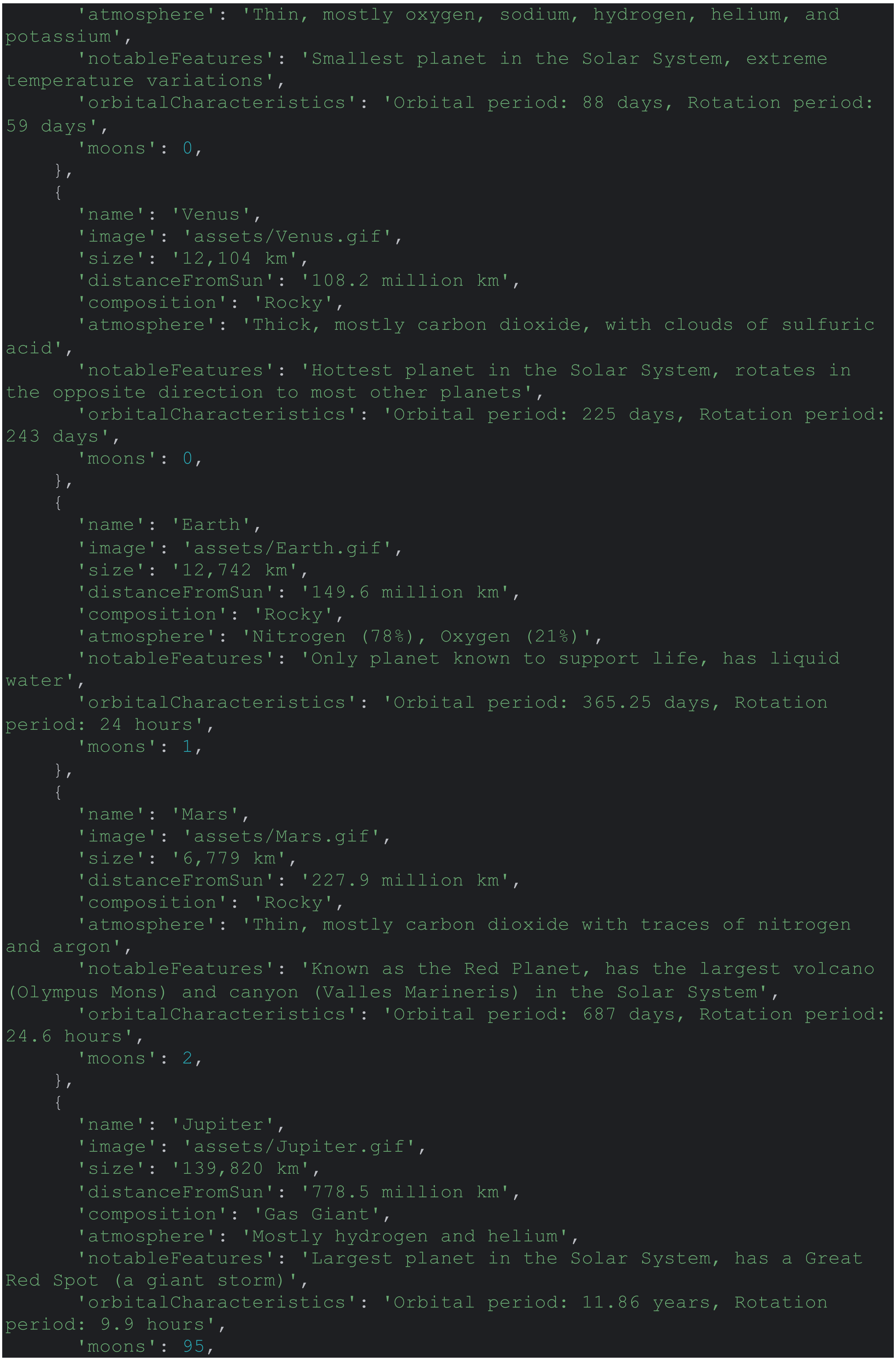
Source Code



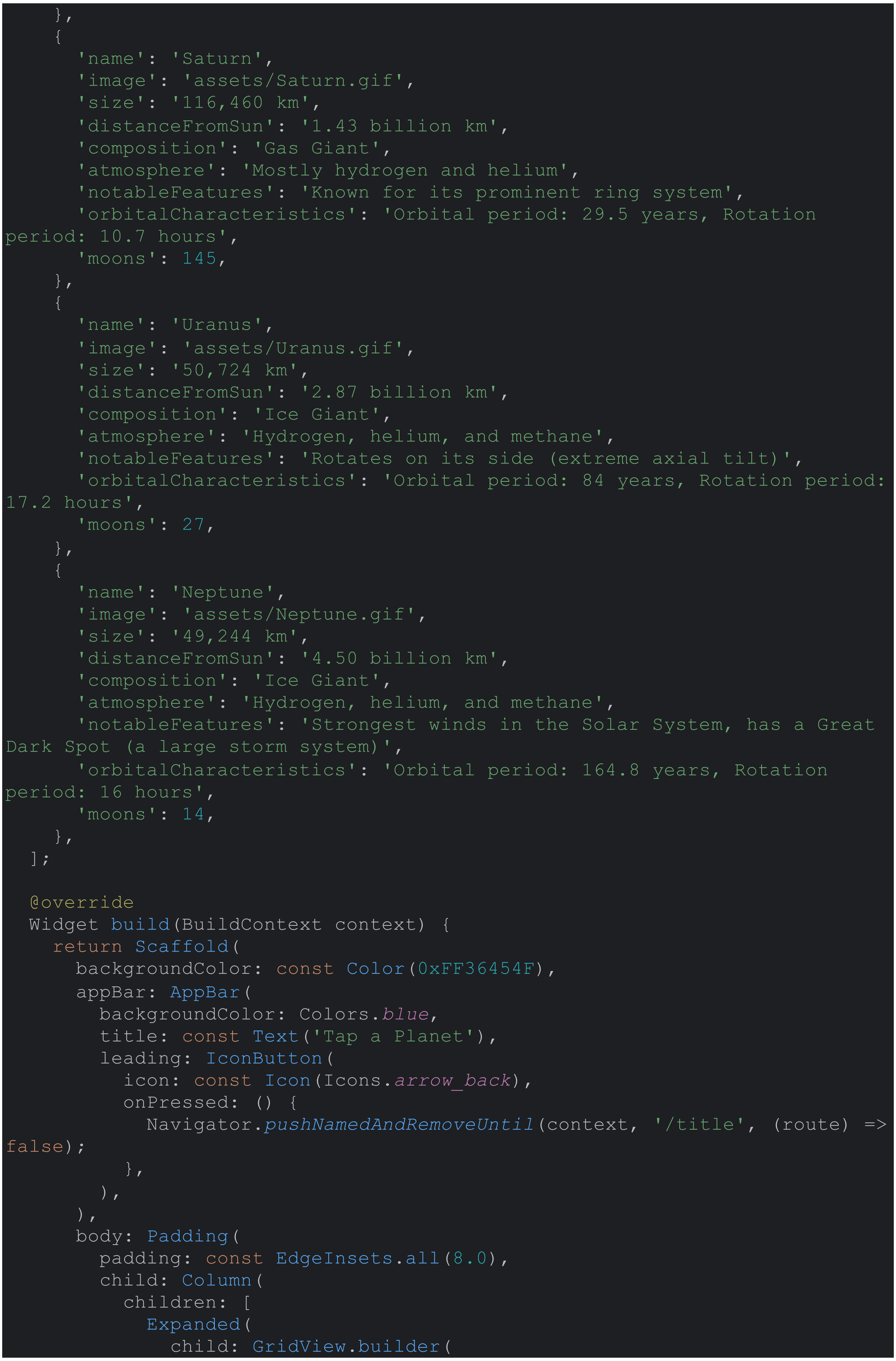
```dart
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'ExplOrbit',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      initialRoute: '/title',
      routes: {
        '/title': (context) => const TitleScreen(),
        '/home': (context) => HomeScreen(),
        '/quiz': (context) => const QuizSelectionScreen(),
      },
    );
  }
}

class TitleScreen extends StatelessWidget {
  const TitleScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        fit: StackFit.expand,
        children: [
          Image.asset(
            'assets/background.gif',
            fit: BoxFit.cover,
          ),
          Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Center(
                child: Text.rich(
                  TextSpan(
                    children: [
                      TextSpan(
                        text: 'ExplOrbit : ',
                        style: TextStyle(
                          fontSize: 48,
                          fontWeight: FontWeight.bold,
                          color: Colors.blueAccent,
                          fontFamily: 'Baskervville_SC',
                          shadows: [
                            Shadow(
                              blurRadius: 15.0,
                              color: Colors.black.withOpacity(0.6),
                              offset: const Offset(0, 5),
                            ),
```



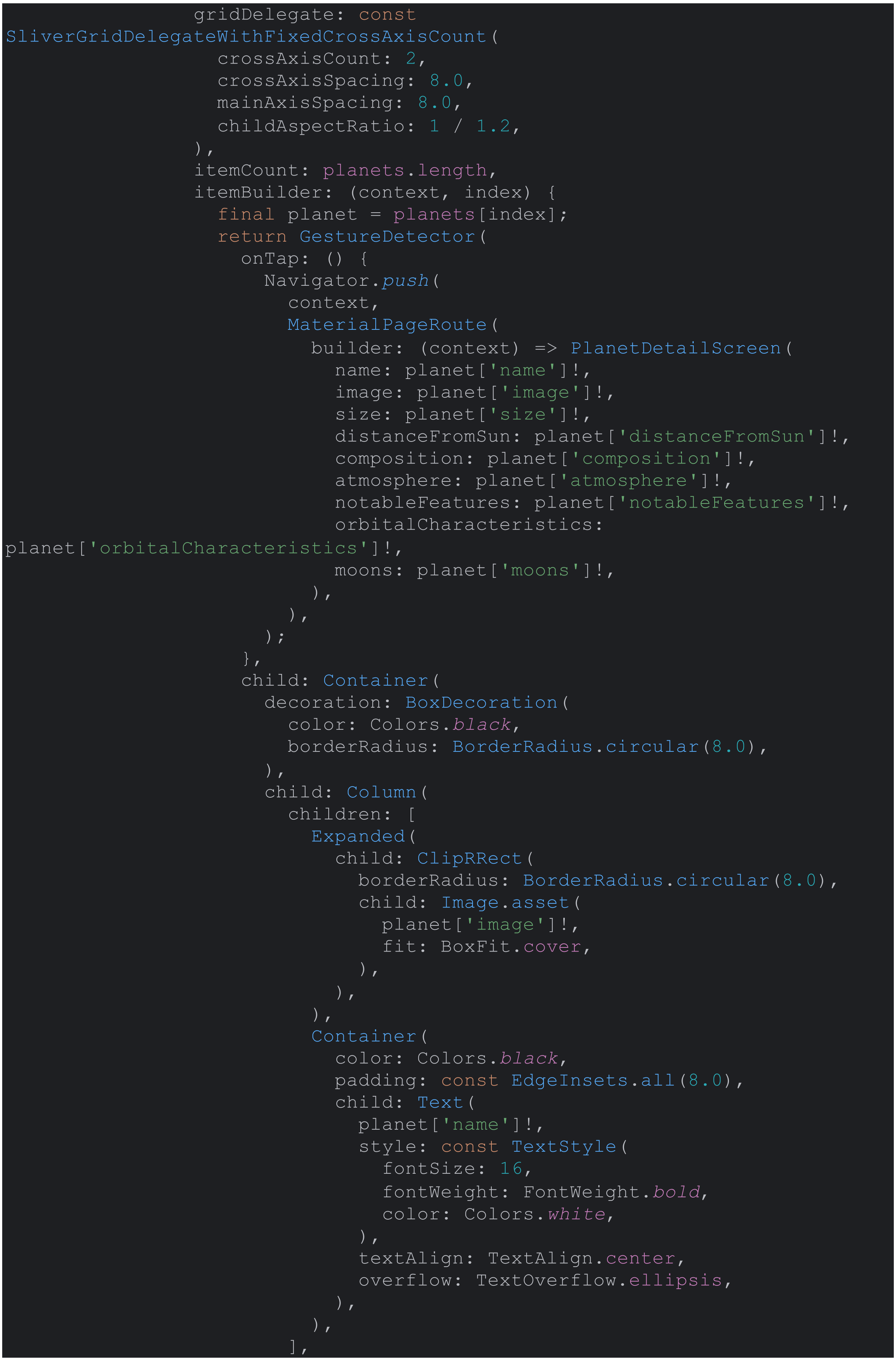
```
                          ],
                        ),
                      ),
                      TextSpan(
                        text: 'Planet Facts',
                        style: TextStyle(
                          fontSize: 48,
                          fontWeight: FontWeight.bold,
                          color: Colors.purple,
                          shadows: [
                            Shadow(
                              blurRadius: 15.0,
                              color: Colors.black.withOpacity(0.6),
                              offset: const Offset(0, 5),
                            ),
                          ],
                        ),
                      ),
                    ],
                  ),
                  textAlign: TextAlign.center,
                ),
              ),
              const SizedBox(height: 40),
              ElevatedButton(
                onPressed: () {
                  Navigator.of(context).pushReplacementNamed('/home');
                },
                style: ElevatedButton.styleFrom(
                  backgroundColor: Colors.blueAccent,
                  foregroundColor: Colors.white,
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(30),
                  ),
                  padding: const EdgeInsets.symmetric(horizontal: 32,
vertical: 16),
                  textStyle: const TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                child: const Text('Start'),
              ),
            ],
          ),
        ],
      ),
    );
  }
}

class HomeScreen extends StatelessWidget {
  HomeScreen({super.key});

  final List<Map<String, dynamic>> planets = [
    {
      'name': 'Mercury',
      'image': 'assets/Mercury.gif',
      'size': '4,880 km',
      'distanceFromSun': '57.9 million km',
      'composition': 'Rocky',
```



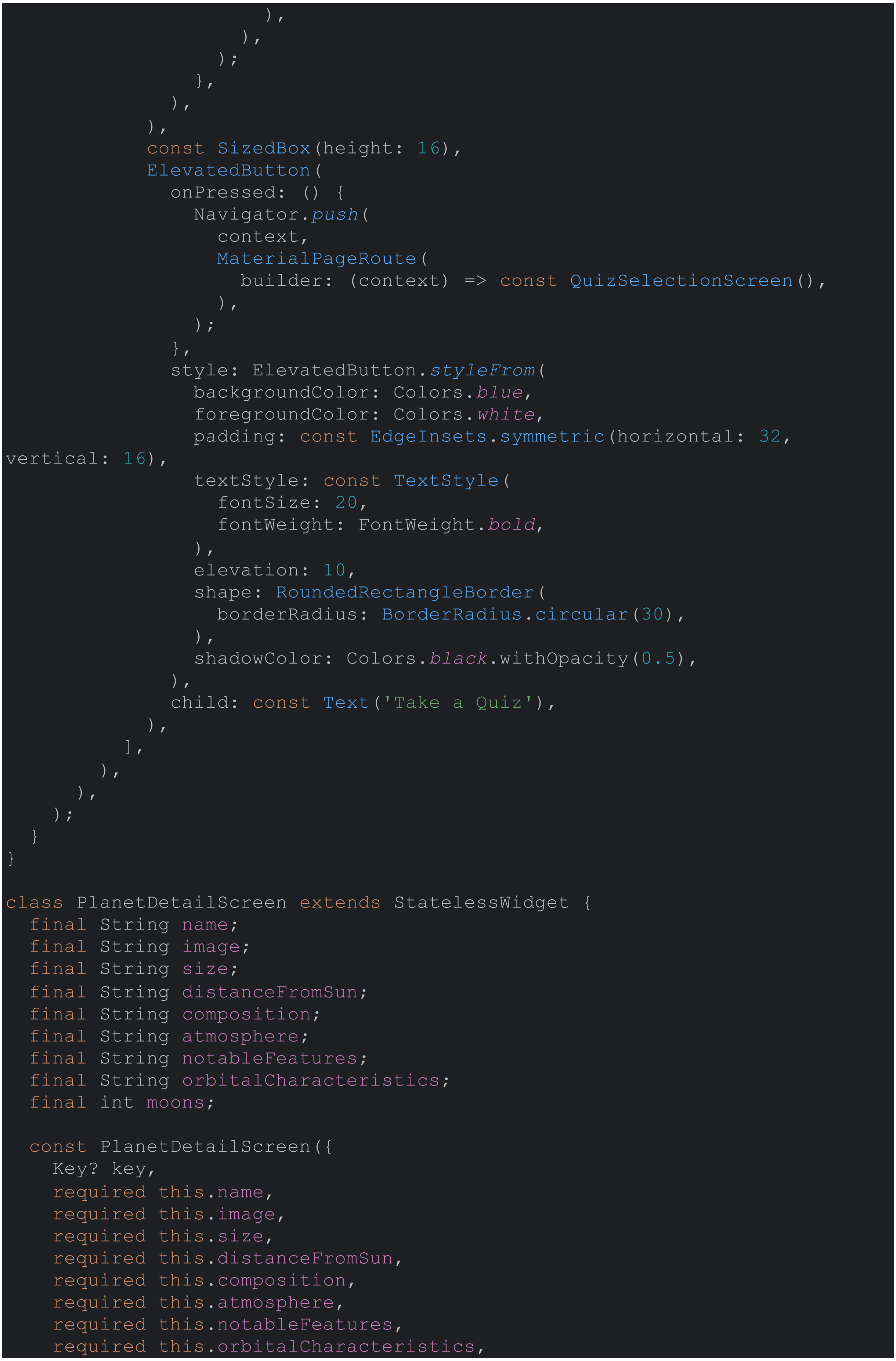
```
      'atmosphere': 'Thin, mostly oxygen, sodium, hydrogen, helium, and
potassium',
      'notableFeatures': 'Smallest planet in the Solar System, extreme
temperature variations',
      'orbitalCharacteristics': 'Orbital period: 88 days, Rotation period:
59 days',
      'moons': 0,
    },
    {
      'name': 'Venus',
      'image': 'assets/Venus.gif',
      'size': '12,104 km',
      'distanceFromSun': '108.2 million km',
      'composition': 'Rocky',
      'atmosphere': 'Thick, mostly carbon dioxide, with clouds of sulfuric
acid',
      'notableFeatures': 'Hottest planet in the Solar System, rotates in
the opposite direction to most other planets',
      'orbitalCharacteristics': 'Orbital period: 225 days, Rotation period:
243 days',
      'moons': 0,
    },
    {
      'name': 'Earth',
      'image': 'assets/Earth.gif',
      'size': '12,742 km',
      'distanceFromSun': '149.6 million km',
      'composition': 'Rocky',
      'atmosphere': 'Nitrogen (78%), Oxygen (21%)',
      'notableFeatures': 'Only planet known to support life, has liquid
water',
      'orbitalCharacteristics': 'Orbital period: 365.25 days, Rotation
period: 24 hours',
      'moons': 1,
    },
    {
      'name': 'Mars',
      'image': 'assets/Mars.gif',
      'size': '6,779 km',
      'distanceFromSun': '227.9 million km',
      'composition': 'Rocky',
      'atmosphere': 'Thin, mostly carbon dioxide with traces of nitrogen
and argon',
      'notableFeatures': 'Known as the Red Planet, has the largest volcano
(Olympus Mons) and canyon (Valles Marineris) in the Solar System',
      'orbitalCharacteristics': 'Orbital period: 687 days, Rotation period:
24.6 hours',
      'moons': 2,
    },
    {
      'name': 'Jupiter',
      'image': 'assets/Jupiter.gif',
      'size': '139,820 km',
      'distanceFromSun': '778.5 million km',
      'composition': 'Gas Giant',
      'atmosphere': 'Mostly hydrogen and helium',
      'notableFeatures': 'Largest planet in the Solar System, has a Great
Red Spot (a giant storm)',
      'orbitalCharacteristics': 'Orbital period: 11.86 years, Rotation
period: 9.9 hours',
      'moons': 95,
```



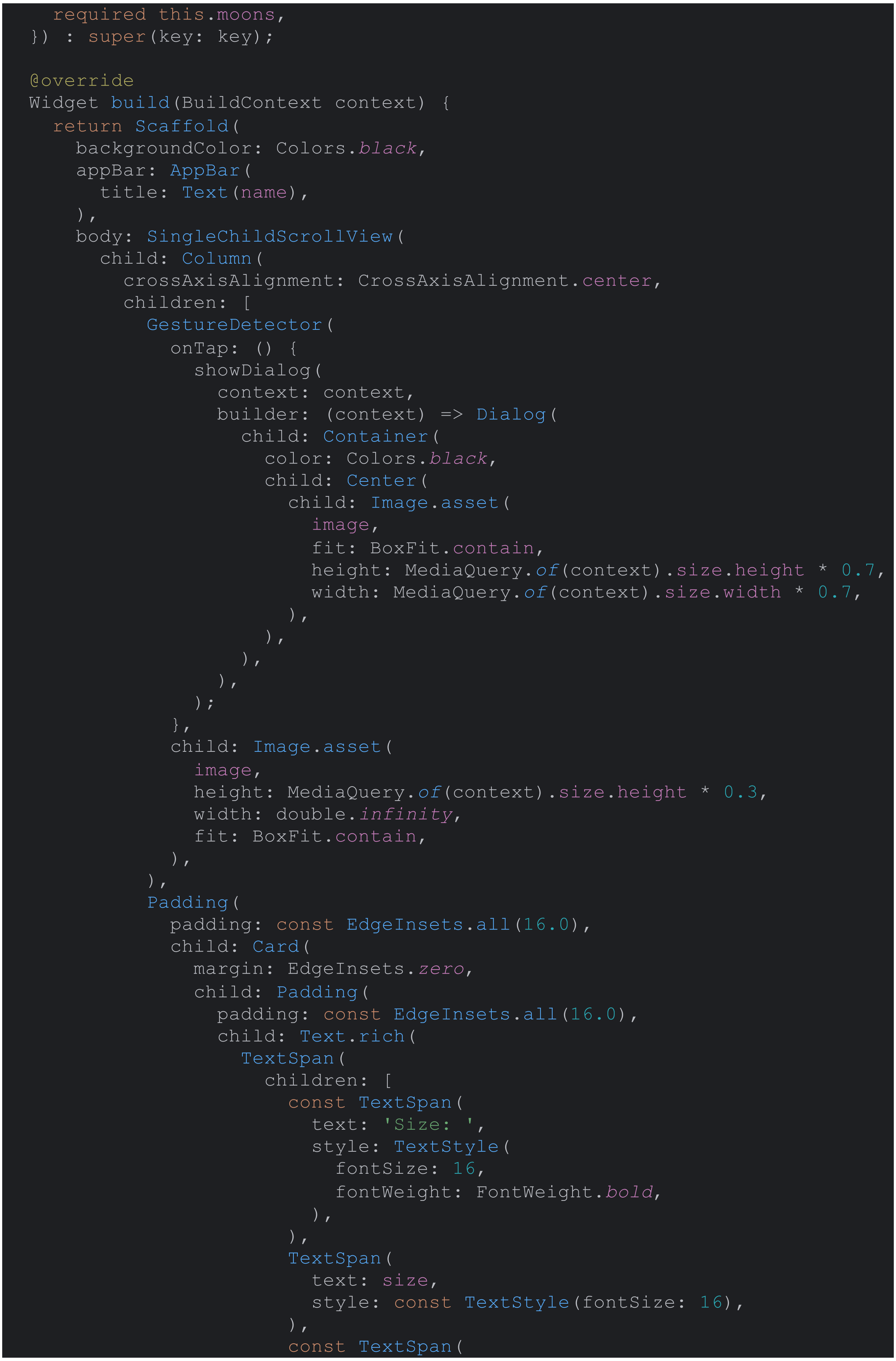
```
    },
    {
      'name': 'Saturn',
      'image': 'assets/Saturn.gif',
      'size': '116,460 km',
      'distanceFromSun': '1.43 billion km',
      'composition': 'Gas Giant',
      'atmosphere': 'Mostly hydrogen and helium',
      'notableFeatures': 'Known for its prominent ring system',
      'orbitalCharacteristics': 'Orbital period: 29.5 years, Rotation
period: 10.7 hours',
      'moons': 145,
    },
    {
      'name': 'Uranus',
      'image': 'assets/Uranus.gif',
      'size': '50,724 km',
      'distanceFromSun': '2.87 billion km',
      'composition': 'Ice Giant',
      'atmosphere': 'Hydrogen, helium, and methane',
      'notableFeatures': 'Rotates on its side (extreme axial tilt)',
      'orbitalCharacteristics': 'Orbital period: 84 years, Rotation period:
17.2 hours',
      'moons': 27,
    },
    {
      'name': 'Neptune',
      'image': 'assets/Neptune.gif',
      'size': '49,244 km',
      'distanceFromSun': '4.50 billion km',
      'composition': 'Ice Giant',
      'atmosphere': 'Hydrogen, helium, and methane',
      'notableFeatures': 'Strongest winds in the Solar System, has a Great
Dark Spot (a large storm system)',
      'orbitalCharacteristics': 'Orbital period: 164.8 years, Rotation
period: 16 hours',
      'moons': 14,
    },
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFF36454F),
      appBar: AppBar(
        backgroundColor: Colors.blue,
        title: const Text('Tap a Planet'),
        leading: IconButton(
          icon: const Icon(Icons.arrow_back),
          onPressed: () {
            Navigator.pushNamedAndRemoveUntil(context, '/title', (route) =>
false);
          },
        ),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Column(
          children: [
            Expanded(
              child: GridView.builder(
```



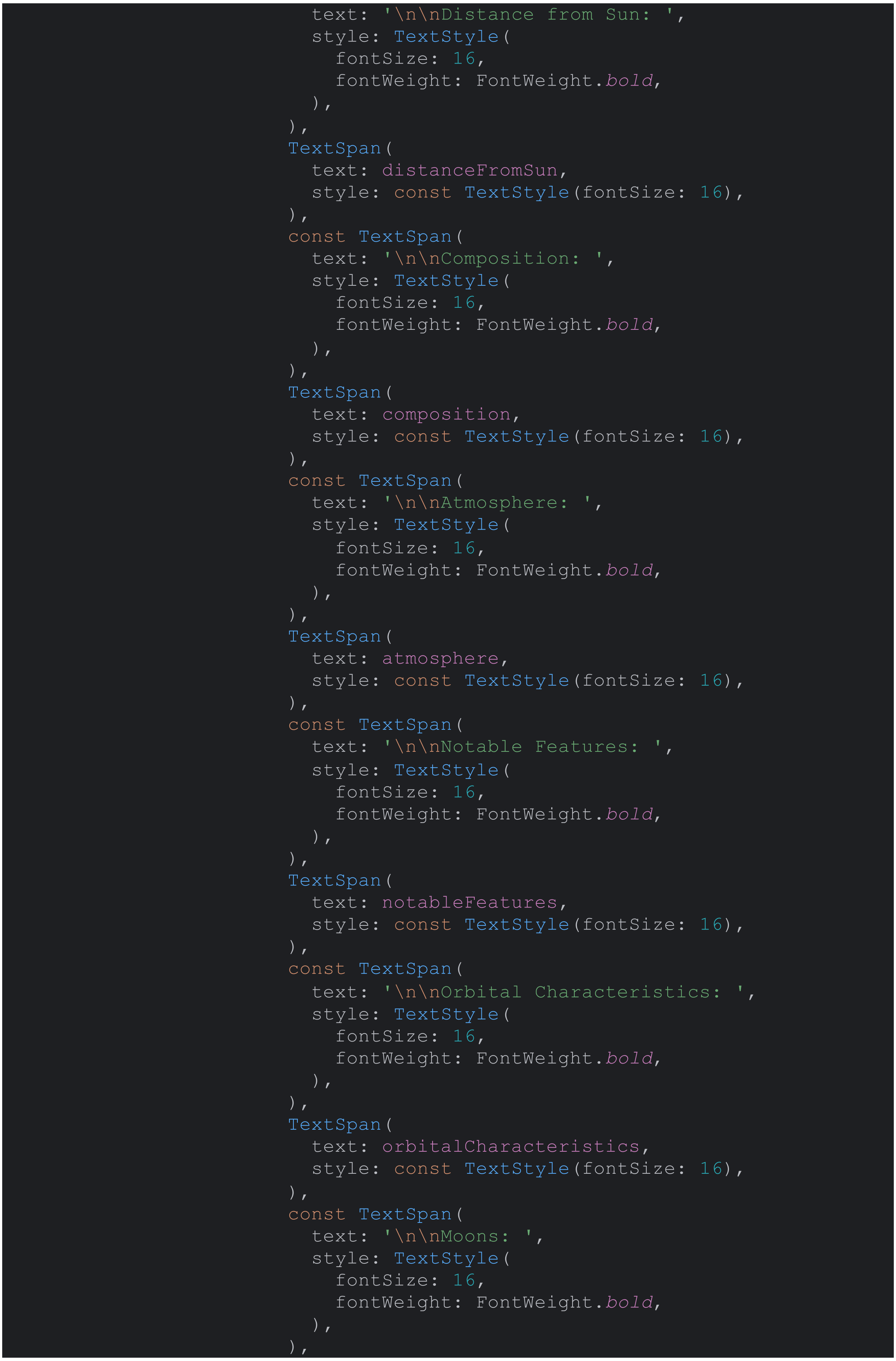
```
                gridDelegate: const
SliverGridDelegateWithFixedCrossAxisCount(
                  crossAxisCount: 2,
                  crossAxisSpacing: 8.0,
                  mainAxisSpacing: 8.0,
                  childAspectRatio: 1 / 1.2,
                ),
                itemCount: planets.length,
                itemBuilder: (context, index) {
                  final planet = planets[index];
                  return GestureDetector(
                    onTap: () {
                      Navigator.push(
                        context,
                        MaterialPageRoute(
                          builder: (context) => PlanetDetailScreen(
                            name: planet['name']!,
                            image: planet['image']!,
                            size: planet['size']!,
                            distanceFromSun: planet['distanceFromSun']!,
                            composition: planet['composition']!,
                            atmosphere: planet['atmosphere']!,
                            notableFeatures: planet['notableFeatures']!,
                            orbitalCharacteristics:
planet['orbitalCharacteristics']!,
                            moons: planet['moons']!,
                          ),
                        ),
                      );
                    },
                    child: Container(
                      decoration: BoxDecoration(
                        color: Colors.black,
                        borderRadius: BorderRadius.circular(8.0),
                      ),
                      child: Column(
                        children: [
                          Expanded(
                            child: ClipRRect(
                              borderRadius: BorderRadius.circular(8.0),
                              child: Image.asset(
                                planet['image']!,
                                fit: BoxFit.cover,
                              ),
                            ),
                          ),
                          Container(
                            color: Colors.black,
                            padding: const EdgeInsets.all(8.0),
                            child: Text(
                              planet['name']!,
                              style: const TextStyle(
                                fontSize: 16,
                                fontWeight: FontWeight.bold,
                                color: Colors.white,
                              ),
                              textAlign: TextAlign.center,
                              overflow: TextOverflow.ellipsis,
                            ),
                          ),
                        ],
```



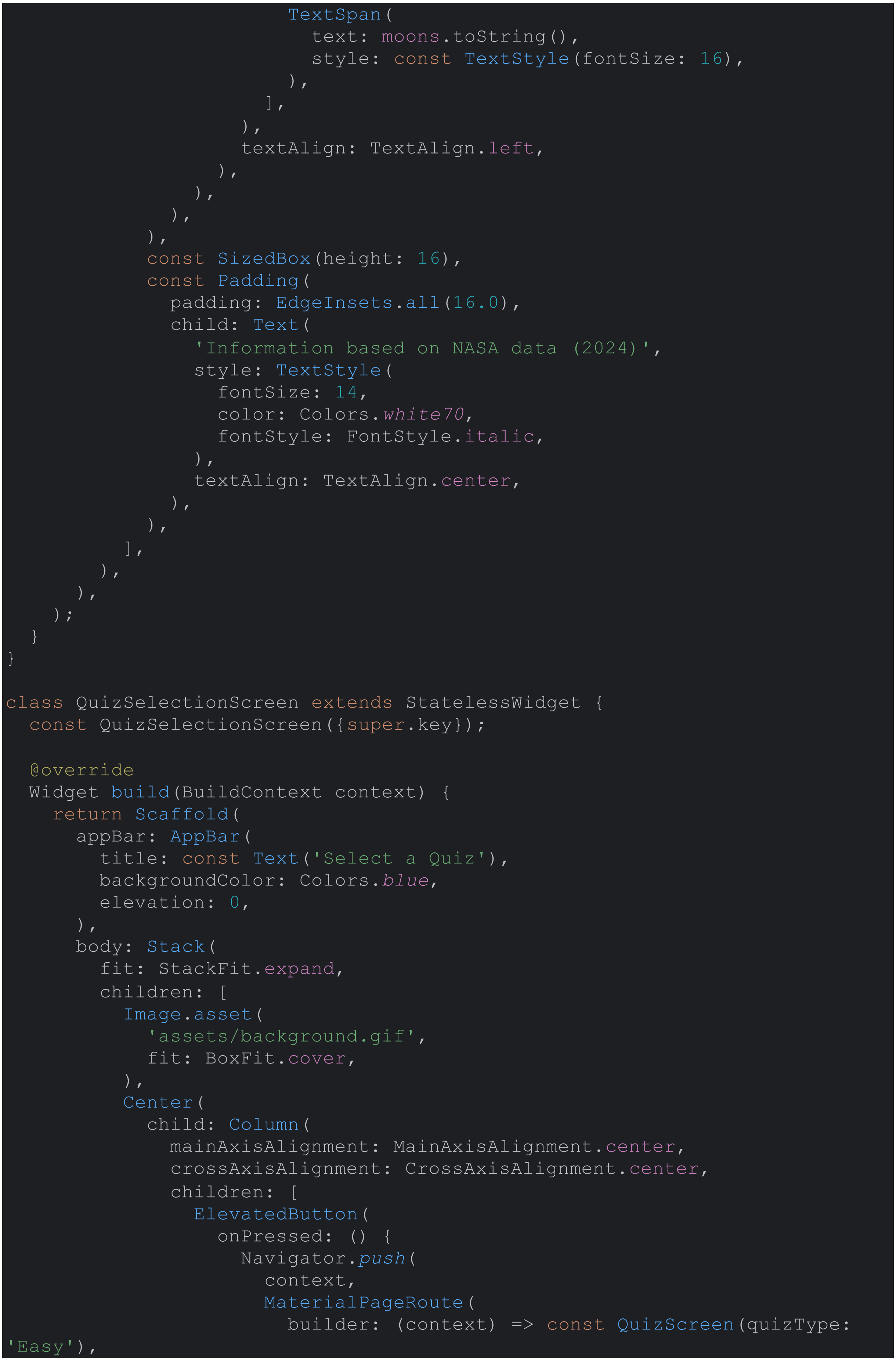
```
                      ),
                    ),
                  );
                },
              ),
            ),
            const SizedBox(height: 16),
            ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => const QuizSelectionScreen(),
                  ),
                );
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.blue,
                foregroundColor: Colors.white,
                padding: const EdgeInsets.symmetric(horizontal: 32,
vertical: 16),
                textStyle: const TextStyle(
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                ),
                elevation: 10,
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(30),
                ),
                shadowColor: Colors.black.withOpacity(0.5),
              ),
              child: const Text('Take a Quiz'),
            ),
          ],
        ),
      ),
    );
  }
}

class PlanetDetailScreen extends StatelessWidget {
  final String name;
  final String image;
  final String size;
  final String distanceFromSun;
  final String composition;
  final String atmosphere;
  final String notableFeatures;
  final String orbitalCharacteristics;
  final int moons;

  const PlanetDetailScreen({
    Key? key,
    required this.name,
    required this.image,
    required this.size,
    required this.distanceFromSun,
    required this.composition,
    required this.atmosphere,
    required this.notableFeatures,
    required this.orbitalCharacteristics,
```



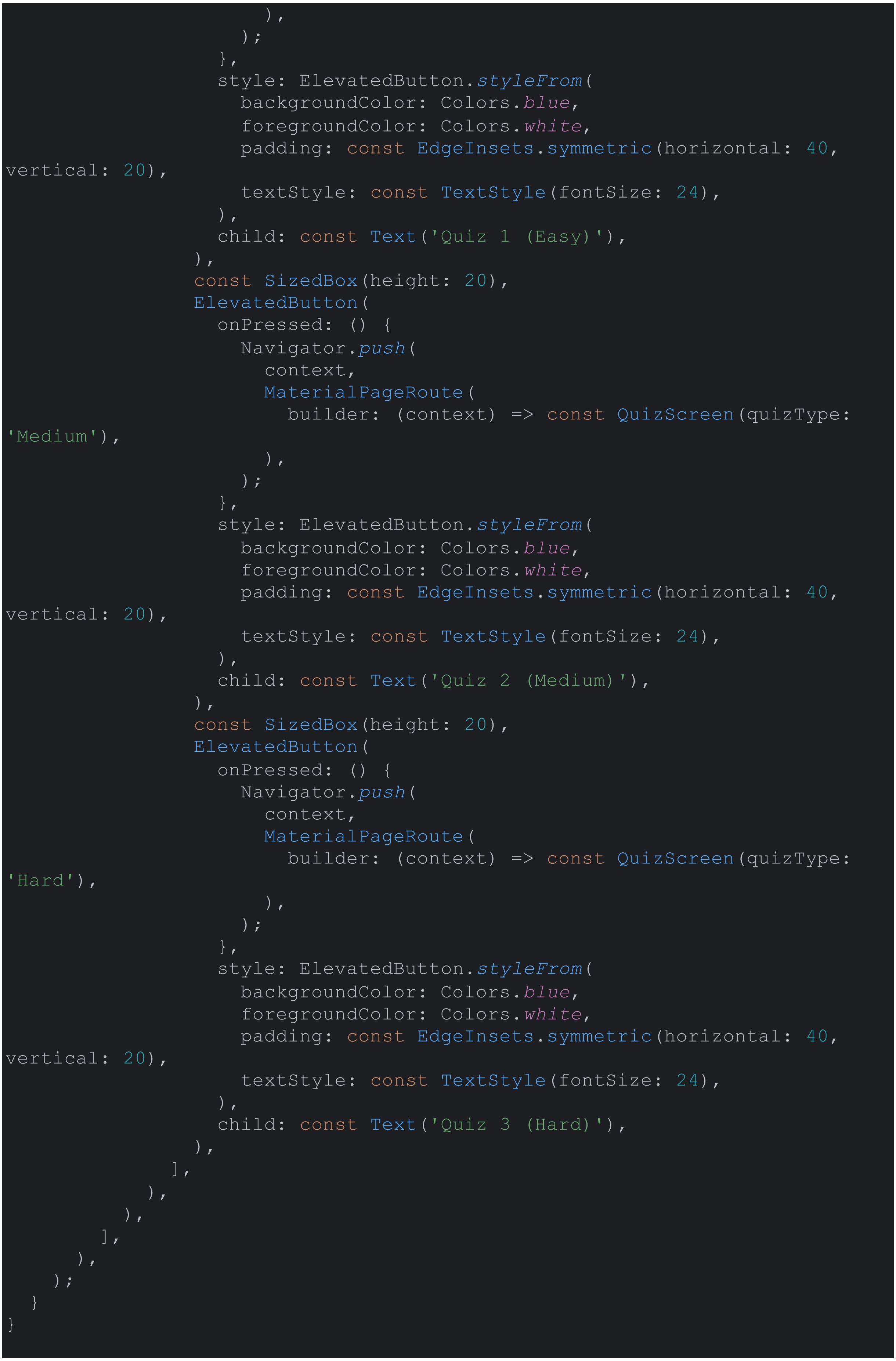
```
    required this.moons,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black,
      appBar: AppBar(
        title: Text(name),
      ),
      body: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            GestureDetector(
              onTap: () {
                showDialog(
                  context: context,
                  builder: (context) => Dialog(
                    child: Container(
                      color: Colors.black,
                      child: Center(
                        child: Image.asset(
                          image,
                          fit: BoxFit.contain,
                          height: MediaQuery.of(context).size.height * 0.7,
                          width: MediaQuery.of(context).size.width * 0.7,
                        ),
                      ),
                    ),
                  ),
                );
              },
              child: Image.asset(
                image,
                height: MediaQuery.of(context).size.height * 0.3,
                width: double.infinity,
                fit: BoxFit.contain,
              ),
            ),
            Padding(
              padding: const EdgeInsets.all(16.0),
              child: Card(
                margin: EdgeInsets.zero,
                child: Padding(
                  padding: const EdgeInsets.all(16.0),
                  child: Text.rich(
                    TextSpan(
                      children: [
                        const TextSpan(
                          text: 'Size: ',
                          style: TextStyle(
                            fontSize: 16,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                        TextSpan(
                          text: size,
                          style: const TextStyle(fontSize: 16),
                        ),
                        const TextSpan(
```



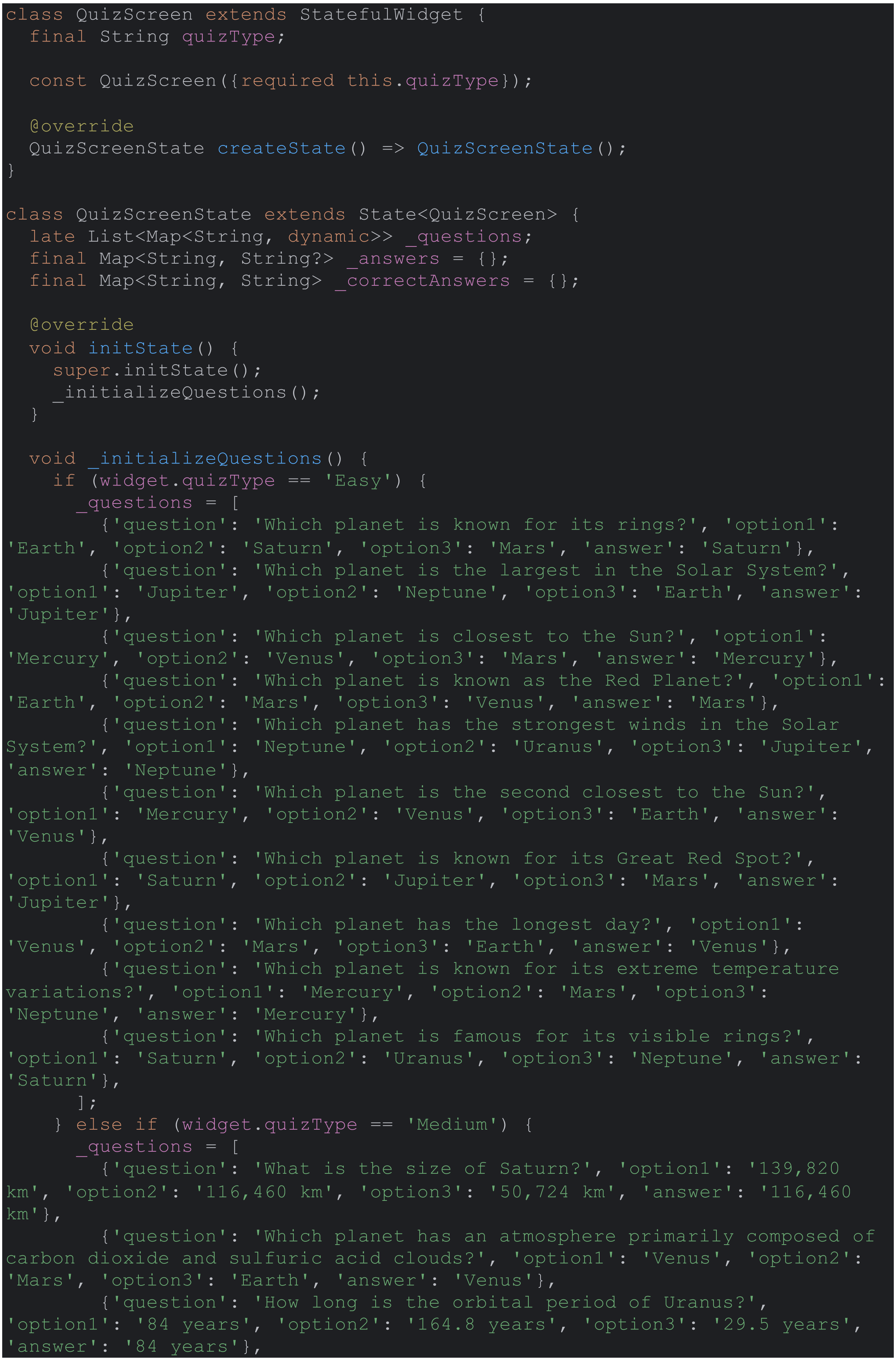
```
                          text: '\n\nDistance from Sun: ',
                          style: TextStyle(
                            fontSize: 16,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                        TextSpan(
                          text: distanceFromSun,
                          style: const TextStyle(fontSize: 16),
                        ),
                        const TextSpan(
                          text: '\n\nComposition: ',
                          style: TextStyle(
                            fontSize: 16,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                        TextSpan(
                          text: composition,
                          style: const TextStyle(fontSize: 16),
                        ),
                        const TextSpan(
                          text: '\n\nAtmosphere: ',
                          style: TextStyle(
                            fontSize: 16,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                        TextSpan(
                          text: atmosphere,
                          style: const TextStyle(fontSize: 16),
                        ),
                        const TextSpan(
                          text: '\n\nNotable Features: ',
                          style: TextStyle(
                            fontSize: 16,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                        TextSpan(
                          text: notableFeatures,
                          style: const TextStyle(fontSize: 16),
                        ),
                        const TextSpan(
                          text: '\n\nOrbital Characteristics: ',
                          style: TextStyle(
                            fontSize: 16,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
                        TextSpan(
                          text: orbitalCharacteristics,
                          style: const TextStyle(fontSize: 16),
                        ),
                        const TextSpan(
                          text: '\n\nMoons: ',
                          style: TextStyle(
                            fontSize: 16,
                            fontWeight: FontWeight.bold,
                          ),
                        ),
```



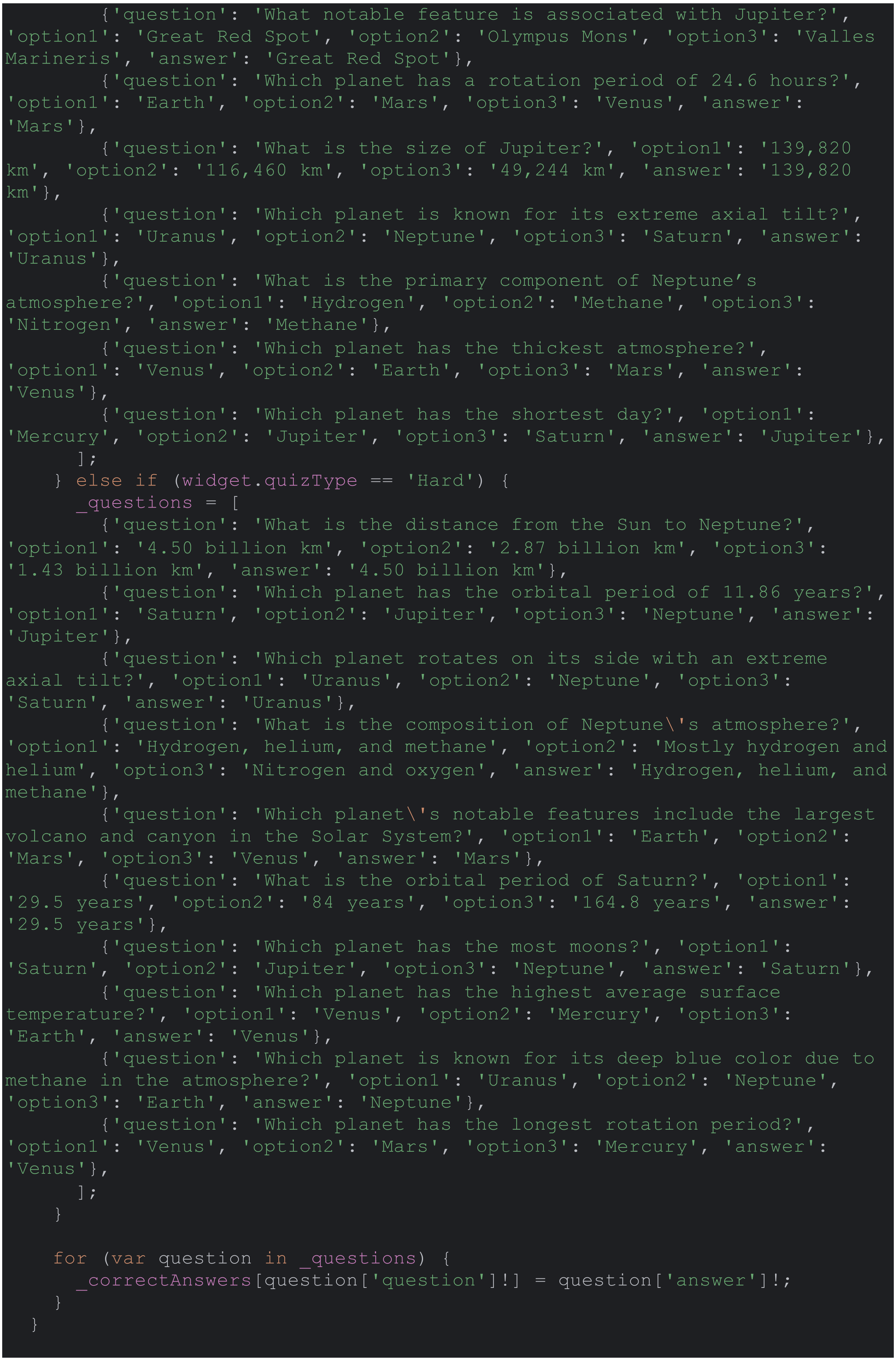
```
                        TextSpan(
                          text: moons.toString(),
                          style: const TextStyle(fontSize: 16),
                        ),
                      ],
                    ),
                    textAlign: TextAlign.left,
                  ),
                ),
              ),
            ),
            const SizedBox(height: 16),
            const Padding(
              padding: EdgeInsets.all(16.0),
              child: Text(
                'Information based on NASA data (2024)',
                style: TextStyle(
                  fontSize: 14,
                  color: Colors.white70,
                  fontStyle: FontStyle.italic,
                ),
                textAlign: TextAlign.center,
              ),
            ),
          ],
        ),
      ),
    );
  }
}

class QuizSelectionScreen extends StatelessWidget {
  const QuizSelectionScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Select a Quiz'),
        backgroundColor: Colors.blue,
        elevation: 0,
      ),
      body: Stack(
        fit: StackFit.expand,
        children: [
          Image.asset(
            'assets/background.gif',
            fit: BoxFit.cover,
          ),
          Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                ElevatedButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => const QuizScreen(quizType:
'Easy'),
```



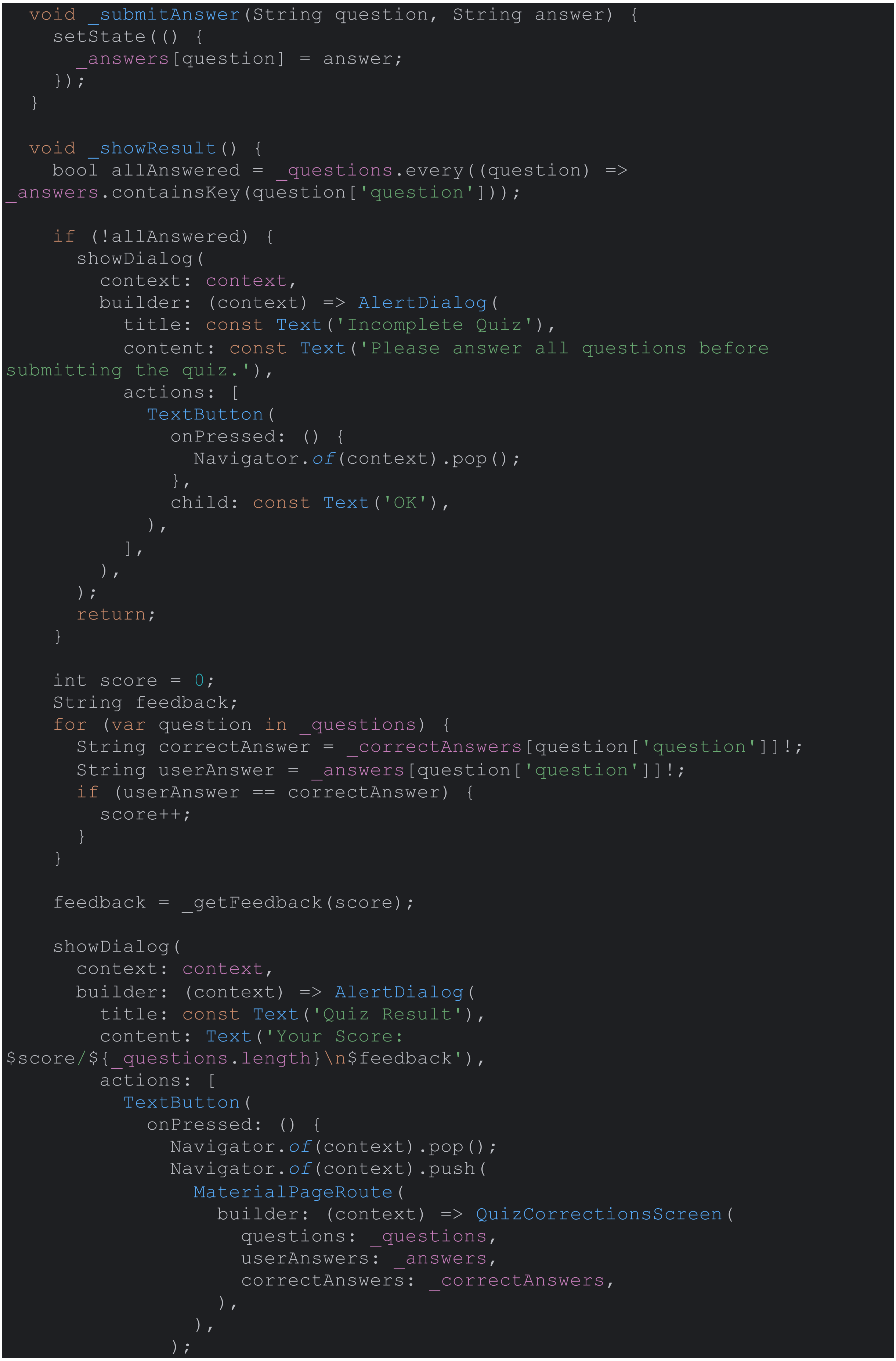
```
                      ),
                    );
                  },
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.blue,
                    foregroundColor: Colors.white,
                    padding: const EdgeInsets.symmetric(horizontal: 40,
vertical: 20),

                    textStyle: const TextStyle(fontSize: 24),
                  ),
                  child: const Text('Quiz 1 (Easy)'),
                ),
                const SizedBox(height: 20),
                ElevatedButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => const QuizScreen(quizType:
'Medium'),

                      ),
                    );
                  },
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.blue,
                    foregroundColor: Colors.white,
                    padding: const EdgeInsets.symmetric(horizontal: 40,
vertical: 20),

                    textStyle: const TextStyle(fontSize: 24),
                  ),
                  child: const Text('Quiz 2 (Medium)'),
                ),
                const SizedBox(height: 20),
                ElevatedButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => const QuizScreen(quizType:
'Hard'),

                      ),
                    );
                  },
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.blue,
                    foregroundColor: Colors.white,
                    padding: const EdgeInsets.symmetric(horizontal: 40,
vertical: 20),

                    textStyle: const TextStyle(fontSize: 24),
                  ),
                  child: const Text('Quiz 3 (Hard)'),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```



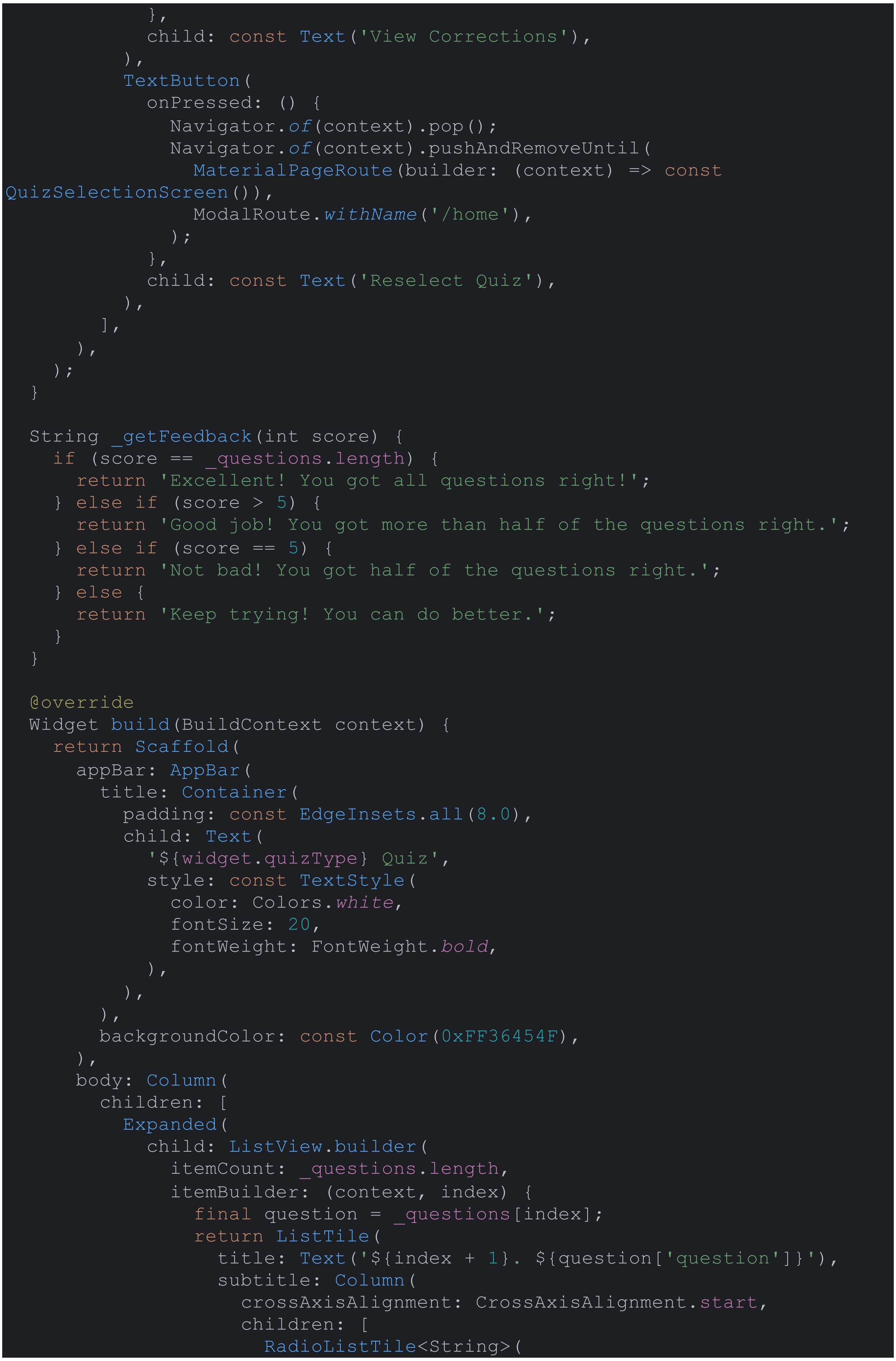
```
class QuizScreen extends StatefulWidget {
  final String quizType;

  const QuizScreen({required this.quizType});

  @override
  QuizScreenState createState() => QuizScreenState();
}

class QuizScreenState extends State<QuizScreen> {
  late List<Map<String, dynamic>> _questions;
  final Map<String, String?> _answers = {};
  final Map<String, String> _correctAnswers = {};

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

  void _initializeQuestions() {
    if (widget.quizType == 'Easy') {
      _questions = [
        {'question': 'Which planet is known for its rings?', 'option1':
'Earth', 'option2': 'Saturn', 'option3': 'Mars', 'answer': 'Saturn'},
        {'question': 'Which planet is the largest in the Solar System?',
'option1': 'Jupiter', 'option2': 'Neptune', 'option3': 'Earth', 'answer':
'Jupiter'},
        {'question': 'Which planet is closest to the Sun?', 'option1':
'Mercury', 'option2': 'Venus', 'option3': 'Mars', 'answer': 'Mercury'},
        {'question': 'Which planet is known as the Red Planet?', 'option1':
'Earth', 'option2': 'Mars', 'option3': 'Venus', 'answer': 'Mars'},
        {'question': 'Which planet has the strongest winds in the Solar
System?', 'option1': 'Neptune', 'option2': 'Uranus', 'option3': 'Jupiter',
'answer': 'Neptune'},
        {'question': 'Which planet is the second closest to the Sun?',
'option1': 'Mercury', 'option2': 'Venus', 'option3': 'Earth', 'answer':
'Venus'},
        {'question': 'Which planet is known for its Great Red Spot?',
'option1': 'Saturn', 'option2': 'Jupiter', 'option3': 'Mars', 'answer':
'Jupiter'},
        {'question': 'Which planet has the longest day?', 'option1':
'Venus', 'option2': 'Mars', 'option3': 'Earth', 'answer': 'Venus'},
        {'question': 'Which planet is known for its extreme temperature
variations?', 'option1': 'Mercury', 'option2': 'Mars', 'option3':
'Neptune', 'answer': 'Mercury'},
        {'question': 'Which planet is famous for its visible rings?',
'option1': 'Saturn', 'option2': 'Uranus', 'option3': 'Neptune', 'answer':
'Saturn'},
      ];
    } else if (widget.quizType == 'Medium') {
      _questions = [
        {'question': 'What is the size of Saturn?', 'option1': '139,820
km', 'option2': '116,460 km', 'option3': '50,724 km', 'answer': '116,460
km'},
        {'question': 'Which planet has an atmosphere primarily composed of
carbon dioxide and sulfuric acid clouds?', 'option1': 'Venus', 'option2':
'Mars', 'option3': 'Earth', 'answer': 'Venus'},
        {'question': 'How long is the orbital period of Uranus?',
'option1': '84 years', 'option2': '164.8 years', 'option3': '29.5 years',
'answer': '84 years'},
```



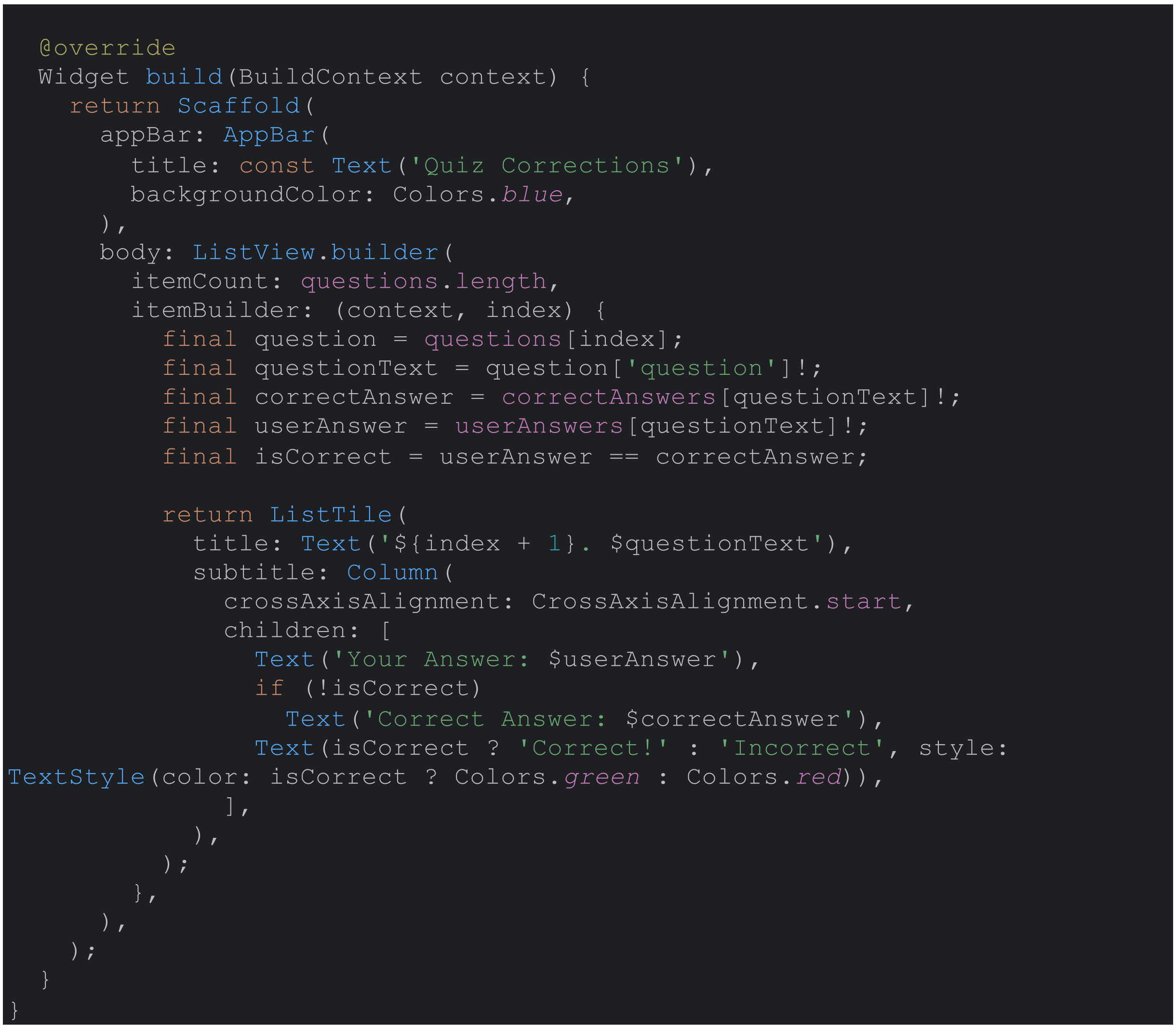
```
        {'question': 'What notable feature is associated with Jupiter?',
'option1': 'Great Red Spot', 'option2': 'Olympus Mons', 'option3': 'Valles
Marineris', 'answer': 'Great Red Spot'},
        {'question': 'Which planet has a rotation period of 24.6 hours?',
'option1': 'Earth', 'option2': 'Mars', 'option3': 'Venus', 'answer':
'Mars'},
        {'question': 'What is the size of Jupiter?', 'option1': '139,820
km', 'option2': '116,460 km', 'option3': '49,244 km', 'answer': '139,820
km'},
        {'question': 'Which planet is known for its extreme axial tilt?',
'option1': 'Uranus', 'option2': 'Neptune', 'option3': 'Saturn', 'answer':
'Uranus'},
        {'question': 'What is the primary component of Neptune’s
atmosphere?', 'option1': 'Hydrogen', 'option2': 'Methane', 'option3':
'Nitrogen', 'answer': 'Methane'},
        {'question': 'Which planet has the thickest atmosphere?',
'option1': 'Venus', 'option2': 'Earth', 'option3': 'Mars', 'answer':
'Venus'},
        {'question': 'Which planet has the shortest day?', 'option1':
'Mercury', 'option2': 'Jupiter', 'option3': 'Saturn', 'answer': 'Jupiter'},
      ];
    } else if (widget.quizType == 'Hard') {
      _questions = [
        {'question': 'What is the distance from the Sun to Neptune?',
'option1': '4.50 billion km', 'option2': '2.87 billion km', 'option3':
'1.43 billion km', 'answer': '4.50 billion km'},
        {'question': 'Which planet has the orbital period of 11.86 years?',
'option1': 'Saturn', 'option2': 'Jupiter', 'option3': 'Neptune', 'answer':
'Jupiter'},
        {'question': 'Which planet rotates on its side with an extreme
axial tilt?', 'option1': 'Uranus', 'option2': 'Neptune', 'option3':
'Saturn', 'answer': 'Uranus'},
        {'question': 'What is the composition of Neptune\'s atmosphere?',
'option1': 'Hydrogen, helium, and methane', 'option2': 'Mostly hydrogen and
helium', 'option3': 'Nitrogen and oxygen', 'answer': 'Hydrogen, helium, and
methane'},
        {'question': 'Which planet\'s notable features include the largest
volcano and canyon in the Solar System?', 'option1': 'Earth', 'option2':
'Mars', 'option3': 'Venus', 'answer': 'Mars'},
        {'question': 'What is the orbital period of Saturn?', 'option1':
'29.5 years', 'option2': '84 years', 'option3': '164.8 years', 'answer':
'29.5 years'},
        {'question': 'Which planet has the most moons?', 'option1':
'Saturn', 'option2': 'Jupiter', 'option3': 'Neptune', 'answer': 'Saturn'},
        {'question': 'Which planet has the highest average surface
temperature?', 'option1': 'Venus', 'option2': 'Mercury', 'option3':
'Earth', 'answer': 'Venus'},
        {'question': 'Which planet is known for its deep blue color due to
methane in the atmosphere?', 'option1': 'Uranus', 'option2': 'Neptune',
'option3': 'Earth', 'answer': 'Neptune'},
        {'question': 'Which planet has the longest rotation period?',
'option1': 'Venus', 'option2': 'Mars', 'option3': 'Mercury', 'answer':
'Venus'},
      ];
    }

    for (var question in _questions) {
      _correctAnswers[question['question']!] = question['answer']!;
    }
  }
```



```
  void _submitAnswer(String question, String answer) {
    setState(() {
      _answers[question] = answer;
    });
  }

  void _showResult() {
    bool allAnswered = _questions.every((question) =>
_answers.containsKey(question['question']));

    if (!allAnswered) {
      showDialog(
        context: context,
        builder: (context) => AlertDialog(
          title: const Text('Incomplete Quiz'),
          content: const Text('Please answer all questions before
submitting the quiz.'),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: const Text('OK'),
            ),
          ],
        ),
      );
      return;
    }

    int score = 0;
    String feedback;
    for (var question in _questions) {
      String correctAnswer = _correctAnswers[question['question']]!;
      String userAnswer = _answers[question['question']]!;
      if (userAnswer == correctAnswer) {
        score++;
      }
    }

    feedback = _getFeedback(score);

    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: const Text('Quiz Result'),
        content: Text('Your Score:
$score/${_questions.length}\n$feedback'),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
              Navigator.of(context).push(
                MaterialPageRoute(
                  builder: (context) => QuizCorrectionsScreen(
                    questions: _questions,
                    userAnswers: _answers,
                    correctAnswers: _correctAnswers,
                  ),
                ),
              );
```



```
            },
            child: const Text('View Corrections'),
          ),
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
              Navigator.of(context).pushAndRemoveUntil(
                MaterialPageRoute(builder: (context) => const
QuizSelectionScreen()),
                ModalRoute.withName('/home'),
              );
            },
            child: const Text('Reselect Quiz'),
          ),
        ],
      ),
    );
  }

  String _getFeedback(int score) {
    if (score == _questions.length) {
      return 'Excellent! You got all questions right!';
    } else if (score > 5) {
      return 'Good job! You got more than half of the questions right.';
    } else if (score == 5) {
      return 'Not bad! You got half of the questions right.';
    } else {
      return 'Keep trying! You can do better.';
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Container(
          padding: const EdgeInsets.all(8.0),
          child: Text(
            '${widget.quizType} Quiz',
            style: const TextStyle(
              color: Colors.white,
              fontSize: 20,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
        backgroundColor: const Color(0xFF36454F),
      ),
      body: Column(
        children: [
          Expanded(
            child: ListView.builder(
              itemCount: _questions.length,
              itemBuilder: (context, index) {
                final question = _questions[index];
                return ListTile(
                  title: Text('${index + 1}. ${question['question']}'),
                  subtitle: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      RadioListTile<String>(
                        title: Text(question['option1']!),
                        value: question['option1']!,
                        groupValue: _answers[question['question']] ?? '',
                        onChanged: (value) =>
_submitAnswer(question['question']!, value!),
                      ),
                      RadioListTile<String>(
                        title: Text(question['option2']!),
                        value: question['option2']!,
                        groupValue: _answers[question['question']] ?? '',
                        onChanged: (value) =>
_submitAnswer(question['question']!, value!),
                      ),
                      RadioListTile<String>(
                        title: Text(question['option3']!),
                        value: question['option3']!,
                        groupValue: _answers[question['question']] ?? '',
                        onChanged: (value) =>
_submitAnswer(question['question']!, value!),
                      ),
                    ],
                  ),
                );
              },
            ),
          ),
          ElevatedButton(
            onPressed: _showResult,
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.blue,
              foregroundColor: Colors.white,
              padding: const EdgeInsets.symmetric(horizontal: 32, vertical:
16),
              textStyle: const TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
              ),
              elevation: 10,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(30),
              ),
              shadowColor: Colors.black.withOpacity(0.5),
            ),
            child: const Text('Submit'),
          ),
        ],
      ),
    );
  }
}

class QuizCorrectionsScreen extends StatelessWidget {
  final List<Map<String, dynamic>> questions;
  final Map<String, String?> userAnswers;
  final Map<String, String> correctAnswers;

  const QuizCorrectionsScreen({
    required this.questions,
    required this.userAnswers,
    required this.correctAnswers,
  });
```



```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Quiz Corrections'),
        backgroundColor: Colors.blue,
      ),
      body: ListView.builder(
        itemCount: questions.length,
        itemBuilder: (context, index) {
          final question = questions[index];
          final questionText = question['question']!;
          final correctAnswer = correctAnswers[questionText]!;
          final userAnswer = userAnswers[questionText]!;
          final isCorrect = userAnswer == correctAnswer;

          return ListTile(
            title: Text('${index + 1}. $questionText'),
            subtitle: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text('Your Answer: $userAnswer'),
                if (!isCorrect)
                  Text('Correct Answer: $correctAnswer'),
                Text(isCorrect ? 'Correct!' : 'Incorrect', style:
TextStyle(color: isCorrect ? Colors.green : Colors.red)),
              ],
            ),
          );
        },
      ),
    );
  }
}
```