# EXPERIMENT NO: - 02

**Name:-** Ronak Katariya **Class:-** D15A **Roll:No: -** 23

**AIM: -** To design Flutter UI by including common widgets.

---

## Theory: -

Each element on the screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the apps. And the structure of the code of apps is a tree of widgets.

When you made any alteration in the code, the widget rebuilds its description by calculating the difference of previous and current widget to determine the minimal changes for rendering in UI of the app.Widgets are nested with each other to build the app. It means the root of your app is itself a widget, and all the way down is a widget also. For example, a widget can display something, can define design, can handle interaction, etc.

The single child layout widget is a type of widget, which can have only **one child widget** inside the parent layout widget. These widgets can also contain special layout functionality. Flutter provides us many single child widgets to make the app UI attractive. If we use these widgets appropriately, it can save our time and makes the app code more readable.

The multiple child widgets are a type of widget, which contains **more than one child widget**, and the layout of these widgets are **unique**. For example, Row widget laying out of its child widget in a horizontal direction, and Column widget laying out of its child widget in a vertical direction. If we combine the Row and Column widget, then it can build any level of the complex widget.

## Type of Widgetss

- **StatefulWidget**

  A StatefulWidget has state information. It contains mainly two classes: the state object and the widget. It is dynamic because it can change the inner data during the widget lifetime. This widget does not have a build() method. It has createState() method, which returns a class that extends the Flutters State Class. The examples of the StatefulWidget are Checkbox, Radio, Slider, InkWell, Form, and TextField.

- **StatelessWidget**

  The StatelessWidget does not have any state information. It remains static throughout its lifecycle. The examples of the StatelessWidget are Text, Row, Column, Container, etc.

## Some of the commonly used widgets

Container – A box widget used for styling with padding, margins, colors, borders, and constraints. It helps in layout structuring and positioning.

Row & Column – Used to arrange widgets in horizontal (Row) or vertical (Column) orientation. They manage spacing, alignment, and distribution of child widgets.

Stack – Overlaps widgets on top of each other, useful for creating layered UIs like banners, tooltips, or floating elements.

Text – Displays text on the screen with customizable font size, color, alignment, and styling options

Image – Loads and displays images from assets, network, or memory with scaling, fit, properties.

Scaffold – Provides a basic layout structure with an app bar, body, floating action button, and bottom navigation.

ListView – A scrollable list widget that efficiently renders large amounts of dynamic content. Supports both vertical and horizontal scrolling.

GridView – Displays widgets in a grid format, useful for galleries, product listings, or dashboards. It supports dynamic column adjustments.

SizedBox – Used to create space between widgets or define fixed width and height for layout adjustments.

ElevatedButton – A button with elevation that provides a raised effect, customizable with color, shape, and click actions.

TextField – A user input field that supports text entry, keyboard configurations, validation.

AppBar – A top navigation bar that includes a title, actions, and menu icons, commonly used in Scaffold.

BottomNavigationBar – A bar at the bottom of the screen used for navigation between different app sections with icons and labels.

Drawer – A side navigation panel that slides out from the left, typically used for app menus and quick navigation.

Card – A material design component that displays content inside a box with elevation.

**Code: - quiz_summary.dart**

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

class QuizSummaryScreen extends
StatelessWidget {
 final int score;
 final int total;
 final List<Map<String, dynamic>>
userResponses;

 const QuizSummaryScreen({super.key,
required this.score, required this.total, required
this.userResponses});

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(title: const Text("Quiz
Summary")),
   body: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
     crossAxisAlignment:
CrossAxisAlignment.start,
     children: [
      Text("Your Score: $score / $total", style:
const TextStyle(fontSize: 26, fontWeight:
FontWeight.bold)),
      const SizedBox(height: 20),
      Expanded(
       child: ListView(
        children:
userResponses.map((response) => Card(
         child: ListTile(
          title: Text(response['question']),
          subtitle: Text("Your Answer:
${response['selected']} \nCorrect Answer:
${response['correct']}"),
          trailing: response['isCorrect'] ?
const Icon(Icons.check, color: Colors.green) :
const Icon(Icons.close, color: Colors.red),
         ),
        )).toList(),
       ),
      ),
      ElevatedButton(
       onPressed: () => Navigator.pop(context),
       child: const Text("Back to Categories"),
      ),
     ],
    ),



   ),
  );
 }
}
```

**Code: landing_page.dart**

```
import 'package:flutter/material.dart';
import 'package:quiz/widgets/bottom_navbar.dart';
import
'package:quiz/screens/category_selection_screen.dart'
;
import 'package:quiz/screens/leaderboard.dart';
import 'package:quiz/screens/profile.dart';

class LandingPage extends StatefulWidget {
 const LandingPage({super.key});

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

class _LandingPageState extends
State<LandingPage> {
 int _selectedIndex = 0;

 final List<Widget> _pages = [
  const HomeScreen(), // Landing Page UI
  const CategorySelectionScreen(),
  const LeaderboardPage(),
  const ProfilePage(),
 ];

 void _onItemTapped(int index) {
  setState(() {
   _selectedIndex = index;
  });
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   body: _pages[_selectedIndex],
   bottomNavigationBar: BottomNavBar(
    currentIndex: _selectedIndex,
    onTap: _onItemTapped,
   ),
  );
 }
}

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

 @override
 Widget build(BuildContext context) {
  return Scaffold(
 body: HomeContent(),
  );
 }
}

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

 @override
 Widget build(BuildContext context) {
  return Container(
   decoration: const BoxDecoration(
    gradient: LinearGradient(
     colors: [Color(0xFF6A11CB),
Color(0xFF2575FC)],
     begin: Alignment.topCenter,
     end: Alignment.bottomCenter,
    ),
   ),
   child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
     const Padding(
      padding: EdgeInsets.symmetric(horizontal: 20),
      child: Text(
       "Test Your Knowledge 🚀 ",
       style: TextStyle(
        fontSize: 26,
        fontWeight: FontWeight.bold,
        color: Colors.white,
       ),
       textAlign: TextAlign.center,
      ),
     ),
     const SizedBox(height: 20),
     _buildCategoryCards(context),
     const SizedBox(height: 20),
     ElevatedButton(
      onPressed: () {
       Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => const
CategorySelectionScreen()),
       );
      },
      style: ElevatedButton.styleFrom(
       backgroundColor: Colors.white,
       foregroundColor: Colors.blueAccent,
       shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30),
       ),
              padding: const
EdgeInsets.symmetric(horizontal: 40, vertical: 15),
            ),
            child: const Text("Start Quiz", style:
TextStyle(fontSize: 18)),
          ),
        ],
      ),
    );
  }

  /// ✅ Function to build category cards
  Widget _buildCategoryCards(BuildContext context)
{
    List<Map<String, String>> categories = [
      {"title": "CNS", "image": "assets/cns.jpg"},
      {"title": "OS", "image": "assets/os.jpg"},
      {"title": "DSA", "image": "assets/dsa.jpg"},
      {"title": "SQL", "image": "assets/sql.png"},
    ];

    return SizedBox(
      height: 200,
      child: ListView.builder(
        scrollDirection: Axis.horizontal,
        itemCount: categories.length,
        itemBuilder: (context, index) {
          return
_categoryCard(categories[index]["title"]!,
categories[index]["image"]!, context);
        },
      ),
    );
  }

  /// ✅ Category Card Widget
  Widget _categoryCard(String title, String
imagePath, BuildContext context) {
    return GestureDetector(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => const
CategorySelectionScreen()),
        );
      },
      child: Card(
        margin: const EdgeInsets.symmetric(horizontal: 10),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(20),
        ),
        elevation: 5,
        child: Container(
          width: 150,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(20),
            image: DecorationImage(
              image: AssetImage(imagePath),
              fit: BoxFit.cover,
            ),
          ),
          child: Center(
            child: Text(
              title,
              style: const TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
                color: Colors.white,
                backgroundColor: Colors.black54,
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

**Code: -**

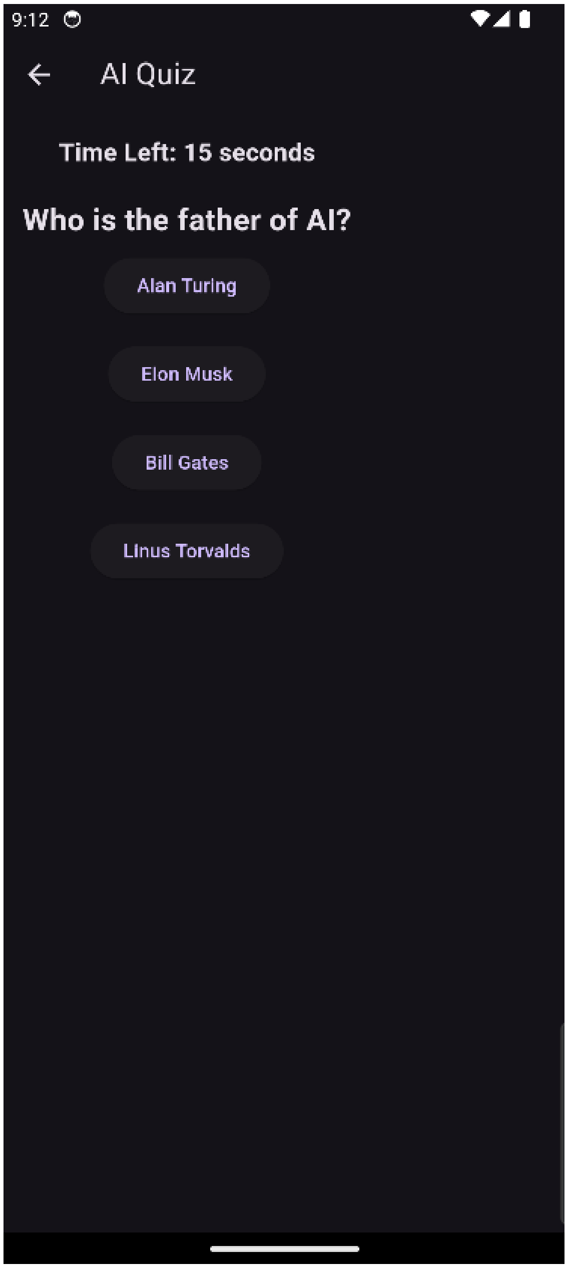
**quiz_screen.dart**

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:quiz/screens/quiz_summary.dart';

class QuizScreen extends StatefulWidget {
  final String category;
  const QuizScreen({super.key, required this.category});

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

class _QuizScreenState extends State<QuizScreen> {
  int currentQuestion = 0;
  int score = 0;
  late Timer timer;
  int timeLeft = 20;
  bool answered = false;
  List<Map<String, dynamic>> userResponses = [];

  final Map<String, List<Map<String, dynamic>>> questionBank = {
    "CNS": [
      {
        "question": "What is a firewall?",
        "options": ["Security Device", "OS", "Protocol", "Network"],
        "answer": "Security Device"
      },
      {
        "question": "What is encryption?",
        "options": ["Encoding data", "Deleting data", "Sending messages", "Accessing files"],
        "answer": "Encoding data"
      },
      {
        "question": "What is a VPN?",
        "options": ["Virtual Private Network", "Variable Public Network", "Visual Private Network", "Virtual Protected Network"],
        "answer": "Virtual Private Network"
      },
      {
        "question": "What does IDS stand for?",
        "options": ["Intrusion Detection System", "Internal Data Security", "Internet Data Service", "Integrated Defense System"],
        "answer": "Intrusion Detection System"
      },
      {
        "question": "What is a DDoS attack?",
        "options": ["Distributed Denial of Service", "Direct Denial of Service", "Distributed Data Service", "Direct Data Service"],
        "answer": "Distributed Denial of Service"
      },
      {
        "question": "What is a proxy server?",
        "options": ["Intermediary for requests", "Data storage device", "Network protocol", "Firewall type"],
        "answer": "Intermediary for requests"
      },
      {
        "question": "What does SSL stand for?",
        "options": ["Secure Sockets Layer", "Standard Security Layer", "Simple Sockets Layer", "Secure System Layer"],
        "answer": "Secure Sockets Layer"
      },
      {
        "question": "What is malware?",
        "options": ["Malicious software", "Machine learning software", "Management software", "Multi-layer software"],
        "answer": "Malicious software"
      },
      {
        "question": "What is phishing?",
        "options": ["Fraudulent attempt to obtain sensitive info","Data encryption method","Network security measure","Type of firewall"],
        "answer":"Fraudulent attempt to obtain sensitive info"
      },
  {
      "question":"What is a botnet?",
      "options":["Network of infected
devices","Type of firewall","Data encryption
method","Software application"],
      "answer":"Network of infected devices"
    }
   ],
   "OS": [
    {"question": "What is CPU scheduling?",
"options": ["Round Robin", "Mutex", "Binary
Search", "Recursion"], "answer": "Round Robin"},
    {"question": "What does OS stand for?",
"options": ["Operating System", "Open Source",
"Output System", "Offline Storage"], "answer":
"Operating System"},
   ],
   "Cloud Computing": [
    {"question": "What is SaaS?", "options":
["Software as a Service", "Storage as a Service",
"Security as a Service", "System as a Service"],
"answer": "Software as a Service"},
    {"question": "AWS stands for?", "options":
["Amazon Web Services", "Azure Web Solutions",
"Advanced Web Server", "Automated Web
Security"], "answer": "Amazon Web Services"},
   ],
   "AI": [
    {"question": "Who is the father of AI?",
"options": ["Alan Turing", "Elon Musk", "Bill
Gates", "Linus Torvalds"], "answer": "Alan
Turing"},
    {"question": "What is NLP?", "options":
["Natural Language Processing", "New Logic
Programming", "Network Layer Protocol", "Next
Level Prediction"], "answer": "Natural Language
Processing"},
   ],
  };

  late List<Map<String, dynamic>> questions;

  @override
  void initState() {
   super.initState();
   questions = questionBank[widget.category] ?? [];
   timer = Timer.periodic(const Duration(seconds:
1), (timer) {
    if (timeLeft == 0) {
     nextQuestion();
    } else {
```

```
setState(() {
     timeLeft--;
    });
   }
  });
 }

 void nextQuestion() {
  if (currentQuestion < questions.length - 1) {
   setState(() {
    currentQuestion++;
    timeLeft = 20;
    answered = false;
   });
  } else {
   Navigator.pushReplacement(
    context,
    MaterialPageRoute(
     builder: (_) => QuizSummaryScreen(score: score,
total: questions.length, userResponses: userResponses),
    ),
   );
  }
 }

 void selectAnswer(String selected) {
  if (!answered) {
   bool isCorrect = selected ==
questions[currentQuestion]['answer'];
   if (isCorrect) score += 1;

   userResponses.add({
    "question": questions[currentQuestion]['question'],
    "selected": selected,
    "correct": questions[currentQuestion]['answer'],
    "isCorrect": isCorrect,
   });

   setState(() {
    answered = true;
   });
  }
 }

 @override
 void dispose() {
  timer.cancel();
  super.dispose();
 }
```

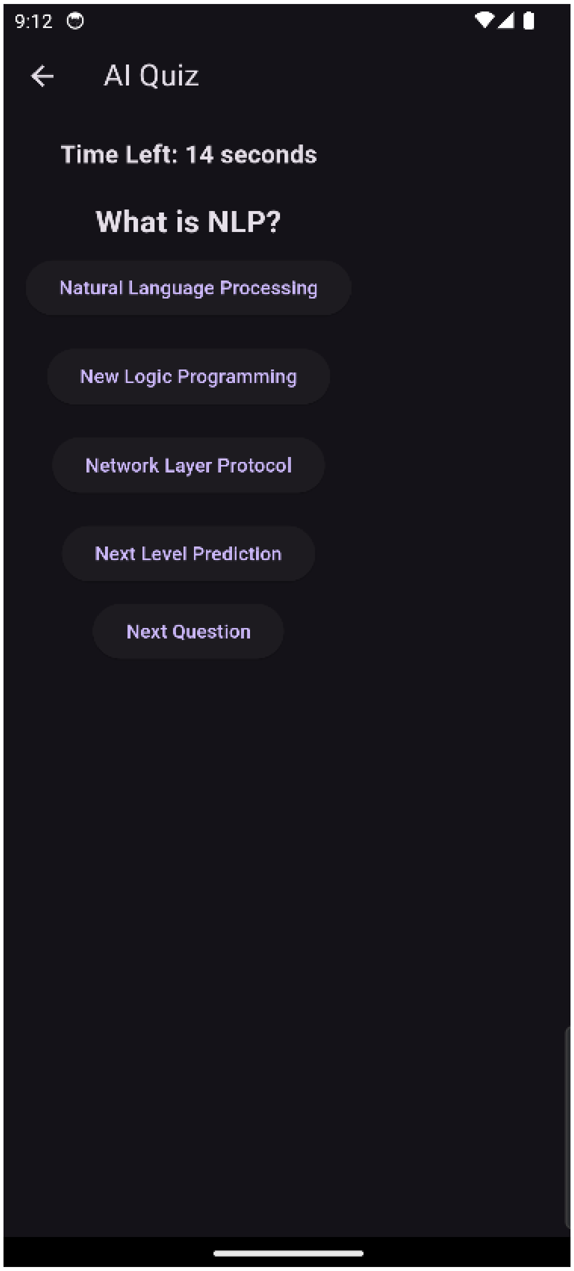
```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("${widget.category} Quiz")),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Text("Time Left: $timeLeft seconds", style: const TextStyle(fontSize: 18, fontWeight: FontWeight.bold)),
            const SizedBox(height: 20),
            Text(questions[currentQuestion]['question'], style: const TextStyle(fontSize: 22, fontWeight: FontWeight.bold)),

...questions[currentQuestion]['options'].map((option
      }
) => Padding(
              padding: const EdgeInsets.symmetric(vertical: 8.0),
              child: ElevatedButton(
                onPressed: () => selectAnswer(option),
                child: Text(option),
              ),
            )),
            if (answered)
              ElevatedButton(
                onPressed: nextQuestion,
                child: const Text("Next Question"),
              ),
          ],
        ),
      ),
    );
  }
}
```

**OUTPUT: -**



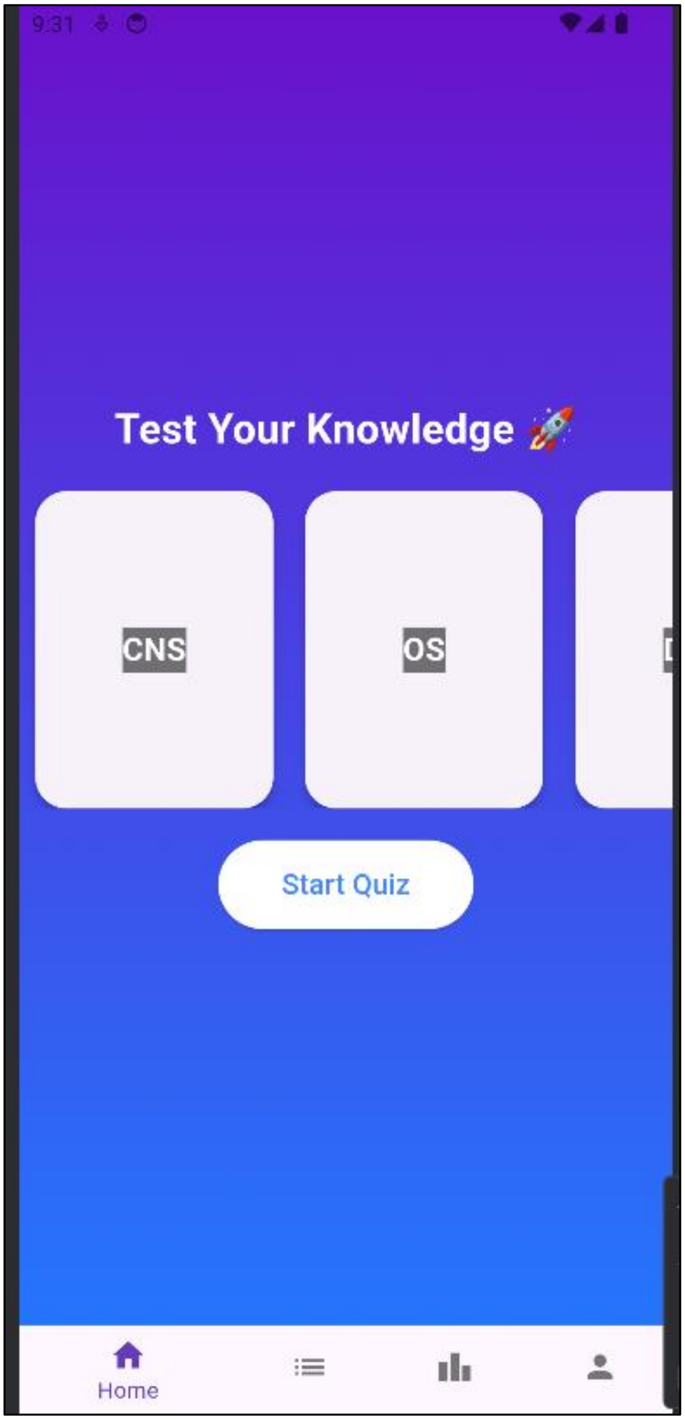
Time Left: 15 seconds

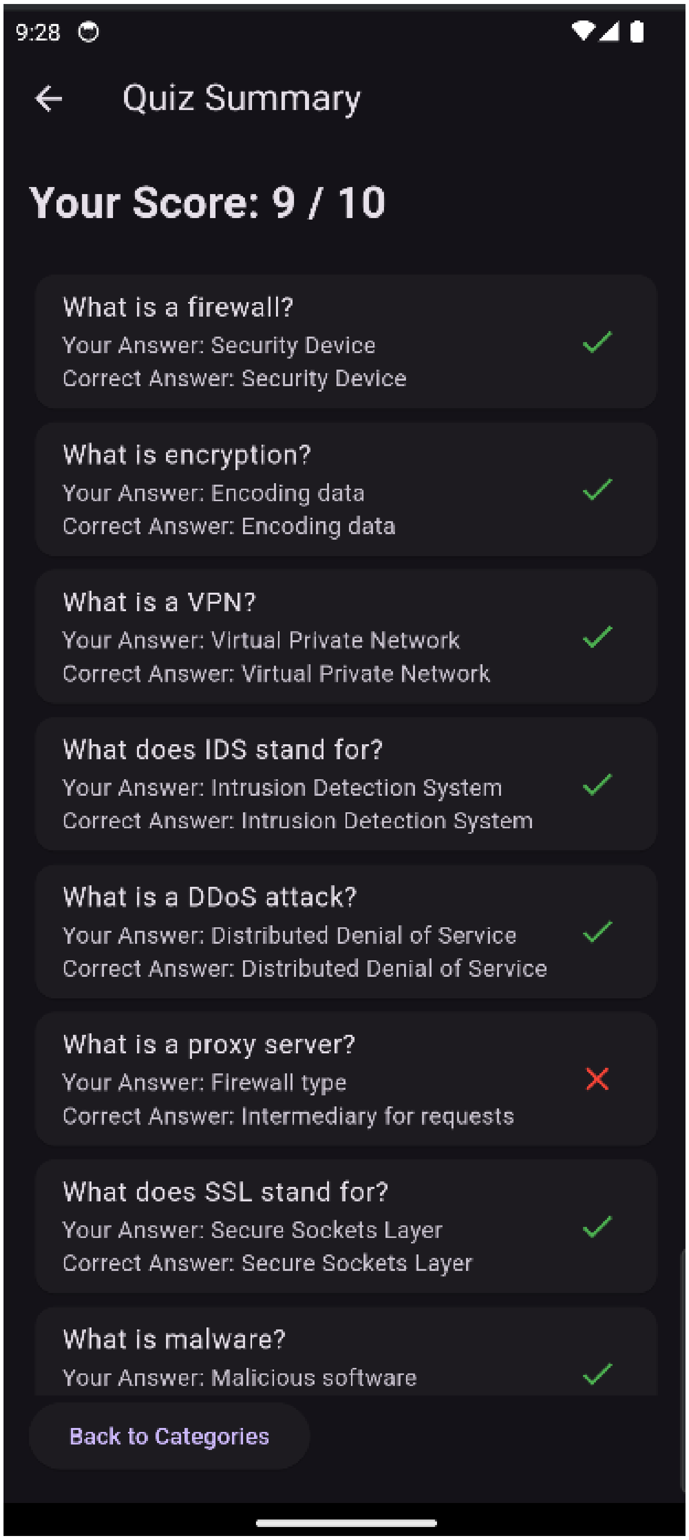
Who is the father of AI?



Time Left: 14 seconds

What is NLP?


9:31
Test Your Knowledge 🚀
CNS
OS
Start Quiz
Home



Quiz Summary

Your Score: 9 / 10

What is a firewall?
Your Answer: Security Device
Correct Answer: Security Device ✓

What is encryption?
Your Answer: Encoding data
Correct Answer: Encoding data ✓

What is a VPN?
Your Answer: Virtual Private Network
Correct Answer: Virtual Private Network ✓

What does IDS stand for?
Your Answer: Intrusion Detection System
Correct Answer: Intrusion Detection System ✓

What is a DDoS attack?
Your Answer: Distributed Denial of Service
Correct Answer: Distributed Denial of Service ✓

What is a proxy server?
Your Answer: Firewall type
Correct Answer: Intermediary for requests ✗

What does SSL stand for?
Your Answer: Secure Sockets Layer
Correct Answer: Secure Sockets Layer ✓

What is malware?
Your Answer: Malicious software ✓