AIM:

Learning Resources

- Validating Input -

  https://www.youtube.com/watch?v=nCgGGfSZ7zY
- Custom Widget Development -

  https://www.geeksforgeeks.org/flutter-custom-widgets/
- Navigation and Routing -

  https://www.youtube.com/watch?v=C6nTXjQFVKI

Practical Task:

- Create an E-Book app with sections for book categories and a simple book reader UI.
- Implement navigation between categories and book details.

Topics Covered:

- Custom Widgets for Book Items
- Navigation and Routing
- Validating Input

THEORY:

□ **Category-Based Filtering**

- The screen filters books based on the category passed to it via the constructor.
- This is achieved using the where method to create a list of books (categoryBooks) that match the selected category.
- This approach ensures that only relevant books are displayed to the user.

- **Dynamic List Display**

- ListView.builder is used to efficiently display a scrollable list of books.
- It builds each item lazily, improving performance when dealing with a large number of books.
- This method enhances app performance by only rendering visible list items.

- **Book Details Display**

- Each book is displayed within a Card widget, giving a structured and organized look.
- ListTile is used within each Card to display:
  - **Title:** Displayed with a bold and slightly larger font.
  - **Author:** Shown in italic style for distinction.
  - **Trailing Icon:** An arrow icon indicating the navigation to the next screen.

State Management with GetX

GetX is a powerful and lightweight state management solution for Flutter. It provides a reactive programming model that automatically updates the UI when the state changes. In addition to state management, GetX offers navigation and dependency injection, making it an all-in-one package for managing the app's architecture. This reduces boilerplate code and improves the maintainability of the project.

Key Elements:

- **Controllers:** These manage business logic and state. For example, a WishlistController can handle adding or removing items from the wishlist, while a ProductDetailsController manages product details like selected quantity and favorite status.
- **Reactive Variables:** GetX uses reactive variables that automatically update the UI when their values change, ensuring a dynamic user experience.
- **Obx:** This widget listens to changes in reactive variables and rebuilds the UI accordingly.

Advantages of Using GetX:

- Reduces boilerplate code.
- Provides reactive state management.
- Offers an intuitive and clean navigation system.
- Enhances performance by efficiently managing state and dependencies.

Responsive Design

Responsive design ensures that the app's UI adapts seamlessly to different screen sizes and orientations. This improves user experience by maintaining consistency across various devices, from small smartphones to large tablets.

Techniques Used:

- **MediaQuery:** This is used to calculate screen dimensions, allowing dynamic sizing for padding, margins, and font sizes.
- **Percentage-Based Layouts:** Dimensions are calculated as percentages of screen width and height, ensuring consistent scaling on different devices.
- **Adaptive Widgets:** Widgets and UI components are designed to adjust their layout and size based on screen dimensions, providing a fluid and flexible interface.

Benefits:

- Consistent user experience across all devices.
- Enhanced usability and accessibility.
- Better adaptability to future devices with varying screen sizes.

Widgets Used

The app utilizes various Flutter widgets to create an intuitive and user-friendly interface. Each widget serves a specific purpose and enhances the overall user experience.

a) AppBar

- The AppBar is customized to have a gradient background, giving the UI a modern and visually appealing look.
- It includes navigation controls, such as a back button and a shopping cart icon, for seamless user navigation.

b) Image Display with Dialog

- Product images are displayed with an interactive feature that allows users to view them in full size.
- This is achieved by wrapping the images in a gesture detector that triggers a dialog, enhancing the product viewing experience.

c) Product Details

- Detailed product information is displayed, including the product name, price, and description.
- Reactive variables ensure that changes, such as selected quantity or favorite status, are instantly reflected in the UI.

Navigation with GetX

GetX simplifies navigation by providing a powerful routing system that eliminates the need for traditional navigation methods like Navigator.push() and Navigator.pop().

Features:

- **Named Routing:** This allows navigation using route names, making it easier to manage complex navigation flows.
- **Arguments Passing:** GetX simplifies passing data between screens, enhancing maintainability and reducing boilerplate code.
- **Transition Animations:** Built-in transition animations provide a smooth and visually appealing navigation experience.

Benefits:

- Cleaner and more maintainable navigation logic.
- Easier to implement complex navigation flows.
- Enhanced user experience with smooth transition animations.

CODE:

main.dart

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'E-Book App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: CategoriesScreen(),
    );
  }
}
```

Book.dart

```
class Book {
  final String id;
  final String title;
  final String author;
  final String category;
  final String pdfName;
  final String imageUrl;

  Book({
    required this.id,
    required this.title,
    required this.author,
    required this.category,
    required this.pdfName,
    required this.imageUrl,
  });
}

final List<Book> books = [
  Book(
    id: '1',
    title: 'Flutter Succinctly',
    author: 'John Doe',
    category: 'Programming',
    pdfName: 'flutter-succinctly.pdf',
    imageUrl:
        'https://imgs.search.brave.com/GphBt536rNlgnJZ4yExDiBzfKBs9AVhlDPIhoM3RrkY/rs:fit:500:0:0:0/g:ce/aHR0cHM6Ly9mbHV0/dGVyYXdlc29tZS5j/b20vY29udGVudC9p/bWFnZXMvMjAxOS8w/NC9GbHV0dGVyLWlu/LUFjdGlvbi5qcGc',
  ),
  Book(
    id: '2',
    title: 'Cooking 101',
    author: 'Jane Smith',
    category: 'Cooking',
    pdfName: 'flutter-succinctly.pdf',
    imageUrl:
      'https://imgs.search.brave.com/GphBt536rNlgnJZ4yExDiBzfKBs9AVhlDPIhoM3RrkY/rs:fit:500:0:0:0/g:ce/aHR0cHM6Ly9mbHV0/dGVyYXdlc29tZS5j/b20vY29udGVudC9p/bWFnZXMvMjAxOS8w/NC9GbHV0dGVyLWlu/LUFjdGlvbi5qcGc',
  ),
  Book(
    id: '3',
    title: 'Introduction to Machine Learning',
    author: 'Alice Johnson',
    category: 'Programming',
    pdfName: 'flutter-succinctly.pdf',
    imageUrl:
      'https://imgs.search.brave.com/GphBt536rNlgnJZ4yExDiBzfKBs9AVhlDPIhoM3RrkY/rs:fit:500:0:0:0/g:ce/aHR0cHM6Ly9mbHV0/dGVyYXdlc29tZS5j/b20vY29udGVudC9p/bWFnZXMvMjAxOS8w/NC9GbHV0dGVyLWlu/LUFjdGlvbi5qcGc',
  ),
  Book(
    id: '4',
    title: 'Travel Guide: Europe',
    author: 'Michael Brown',
    category: 'Travel',
    pdfName: 'flutter-succinctly.pdf',
    imageUrl:
      'https://imgs.search.brave.com/GphBt536rNlgnJZ4yExDiBzfKBs9AVhlDPIhoM3RrkY/rs:fit:500:0:0:0/g:ce/aHR0cHM6Ly9mbHV0/dGVyYXdlc29tZS5j/b20vY29udGVudC9p/bWFnZXMvMjAxOS8w/NC9GbHV0dGVyLWlu/LUFjdGlvbi5qcGc',
  ),
  Book(
    id: '5',
    title: 'Mindfulness and Meditation',
    author: 'Sophia Lee',
    category: 'Self-help',
    pdfName: 'flutter-succinctly.pdf',
    imageUrl:
      'https://imgs.search.brave.com/GphBt536rNlgnJZ4yExDiBzfKBs9AVhlDPIhoM3RrkY/rs:fit:500:0:0:0/g:ce/aHR0cHM6Ly9mbHV0/dGVyYXdlc29tZS5j/b20vY29udGVudC9p/bWFnZXMvMjAxOS8w/NC9GbHV0dGVyLWlu/LUFjdGlvbi5qcGc',
  ),
];
```

categories_screen.dart

```
import 'package:flutter/material.dart';
import 'book_list_screen.dart';
import '../models/book.dart';

class CategoriesScreen extends StatelessWidget {
  final categories = books.map((book) => book.category).toSet().toList();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Book Categories',
            style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
        backgroundColor: const Color.fromARGB(255, 135, 230, 247),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
          itemCount: categories.length,
          itemBuilder: (context, index) {
            return Card(
              elevation: 5,
              color: const Color.fromARGB(255, 224, 240, 243),
              margin: EdgeInsets.symmetric(vertical: 8, horizontal: 5),
              child: ListTile(
                hoverColor: const Color.fromARGB(255, 101, 99, 137),
                contentPadding:
                    EdgeInsets.symmetric(vertical: 10, horizontal: 15),
                title: Text(
                  categories[index],
                  style: TextStyle(fontSize: 18, fontWeight: FontWeight.w500),
                ),
                trailing: Icon(Icons.arrow_forward,
                    color: const Color.fromARGB(255, 11, 105, 111)),
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
                          BookListScreen(category: categories[index]),
                    ),
                  );
                },
              ),
            );
          },
        ),
      ),
    );
  }
}
```

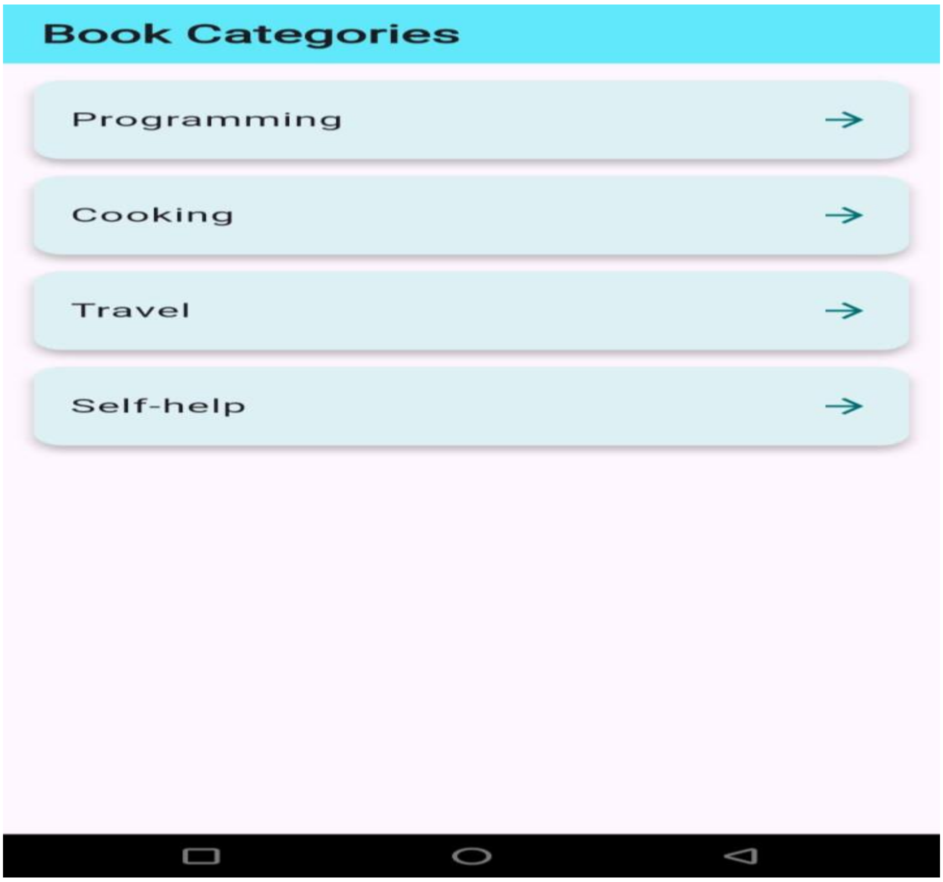
book_list_screen.dart

```
import 'package:flutter/material.dart';
import '../models/book.dart';
import 'book_reader_screen.dart';

class BookListScreen extends StatelessWidget {
  final String category;
  BookListScreen({required this.category});

  @override
  Widget build(BuildContext context) {
    final categoryBooks =
        books.where((book) => book.category == category).toList();

    return Scaffold(
      appBar: AppBar(
        title: Text(category,
            style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
        backgroundColor: const Color.fromARGB(255, 135, 230, 247),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
          itemCount: categoryBooks.length,
          itemBuilder: (context, index) {
            return Card(
              color: const Color.fromARGB(255, 224, 240, 243),
              elevation: 5,
              margin: EdgeInsets.symmetric(vertical: 8, horizontal: 5),
              child: ListTile(
                contentPadding:
                    EdgeInsets.symmetric(vertical: 10, horizontal: 15),
                title: Text(
                  categoryBooks[index].title,
                  style: TextStyle(fontSize: 18, fontWeight: FontWeight.w500),
                ),
                subtitle: Text(
                  categoryBooks[index].author,
                  style: TextStyle(fontSize: 16, fontStyle: FontStyle.italic),
                ),
                trailing: Icon(Icons.arrow_forward,
                    color: const Color.fromARGB(255, 11, 105, 111)),
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
                          BookReaderScreen(book: categoryBooks[index]),
                    ),
                  );
                },
              ),
            );
          },
        ),
      ),
    );
  }
}
```

book_reader_screen.dart

```
import 'package:flutter/material.dart';
import 'package:pdfx/pdfx.dart';
import '../models/book.dart';

class BookReaderScreen extends StatefulWidget {
  final Book book;
  BookReaderScreen({required this.book});

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

class _BookReaderScreenState extends State<BookReaderScreen> {
  late PdfController _pdfController;

  @override
  void initState() {
    super.initState();
    _pdfController = PdfController(
      document: PdfDocument.openAsset('assets/${widget.book.pdfName}'),
    );
  }

  @override
  void dispose() {
    _pdfController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.book.title,
            style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
        backgroundColor: const Color.fromARGB(255, 135, 230, 247),
      ),
      body: PdfView(
        controller: _pdfController,
      ),
    );
  }
}
```

OUTPUT:



Book Categories

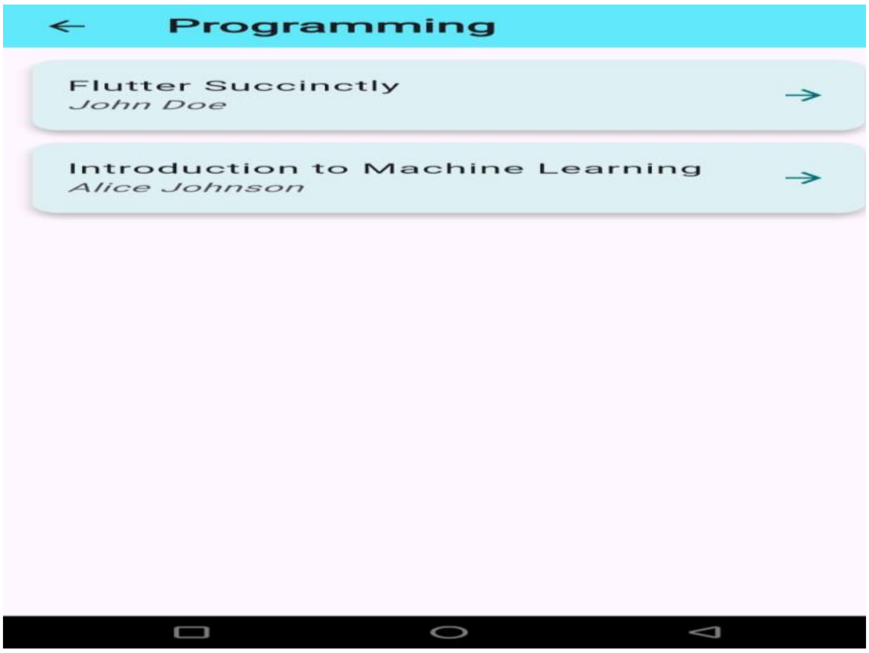
Programming

Cooking

Travel

Self-help

Figure 1:A E-Book-app, displays Book categories

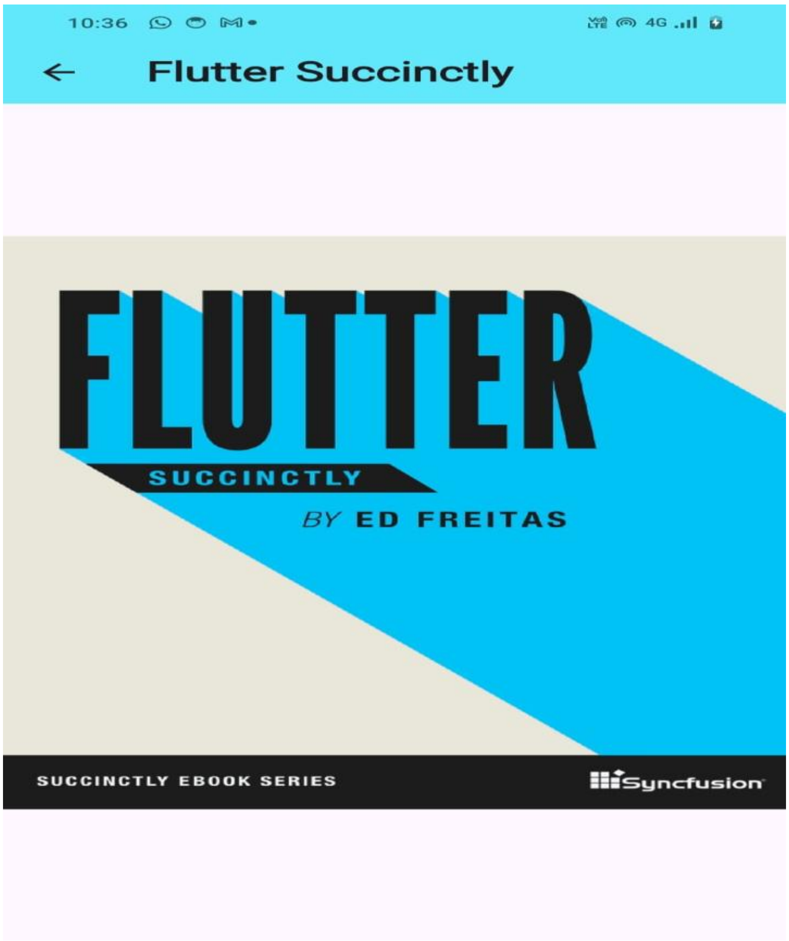


Programming

Flutter Succinctly
John Doe

Introduction to Machine Learning
Alice Johnson

Figure 2:It displays books of "Programming" category



10:36 4G

Flutter Succinctly

Figure 3:It displays "flutter succincity" book in pdf format

Latest Applications:

- Digital Education and eLearning

- Self-Publishing and Online Distribution
- Corporate Training and Development

Learning Outcome:

- I learned to develop cross-platform mobile apps using Flutter and gain experience in integrating PDF and eBook viewing functionalities and implemented smooth navigation and efficient state management.