



#7 Create the Course Detail Page

Introduction

When it comes to online learning, having a detailed course page can make all the difference. The course detail page is where potential students can learn more about the course, its content, instructors, and other details. This section will guide you through creating the course detail page for the educative.io clone with Flutter.

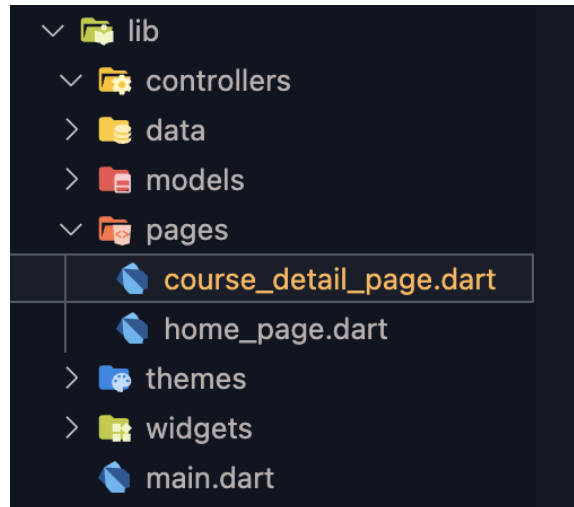
Let's get started! 🚀

Step-by-step guide

Okay, let's get started to create a Course Detail Page step by step.

Step 1 - Create a `course_detail_page.dart` file

Inside `lib/pages` directory create a new file called `course_detail_page.dart`.



Step 2 - Create a ConsumerStatefulWidget



`ConsumerStatefulWidget` is a widget provided by the Flutter Riverpod package. It is similar to the `StatefulWidget` widget, but with the added benefit of being able to listen to changes in the state of a provider. The `ConsumerStatefulWidget` widget is used to build widgets that depend on a provider's state. When the provider changes its state, the `ConsumerStatefulWidget` widget rebuilds, updating its state as necessary.

After creating the `course_detail_page.dart` file, the next step is to create a `ConsumerStatefulWidget` that will represent our course detail page. We will name it `CourseDetailPage`.

Widget `CourseDetailPage` has a constructor parameter that is useful for obtaining `Course` data from the `HomePage`. We will display this data on the `CourseDetailPage` so that users can see the details of the selected course.

We can access the `Course` data using the constructor parameter provided in the Widget `CourseDetailPage`. With this parameter, we can display information such as the course title, number of materials, level, and duration.

The `CourseDetailPage` has several sections that we need to add, such as the course information section, course description section, course content section, and author information section. Each section will display different information, so users can understand more details about the selected course.

The `CourseDetailPage` widget will return a `Scaffold` widget. And inside the `Scaffold` widget, we will call two parameters: `body` and `bottomNavigationBar`. In the `body`, we use a `CustomScrollView` widget and add a `SliverList` widget that contains additional widgets. These widgets will be added in the next step.

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

import '../models/course.dart';

class CourseDetailPage extends ConsumerStatefulWidget {
  const CourseDetailPage({super.key, required this.course});
  final Course course;

  @override
  ConsumerState<CourseDetailPage> createState() => _CourseDetailPageState();
}

class _CourseDetailPageState extends ConsumerState<CourseDetailPage> {
  @override
  Widget build(BuildContext context) {
    return const Scaffold(
      body: CustomScrollView(
        slivers: [
          // Course Thumbnail
          // Course Info
          // Course Description
          // Course Content
          // Course Author
        ],
      ),
      bottomNavigationBar: ,
    );
  }
}
```

Step 3 - Add body content (Course Thumbnail Section)

In this step, we add the body content to our `CustomScrollView` widget. First, we add the course thumbnail section using a `SliverAppBar` widget. This allows us to create a header with an image and a title.

Add the following code snippet to the `slivers` parameter list:

```
// Course Thumbnail
SliverAppBar(
```

```
background-color: MyColors.primary,
expandedHeight: MediaQuery.of(context).size.height * 0.33,
elevation: 0,
pinned: true,
title: Text(
  "Course Detail",
  style: MyTypography.titleSmall.copyWith(
    color: Colors.white,
  ),
),
flexibleSpace: FlexibleSpaceBar(
  background: Stack(
    children: [
      Opacity(
        opacity: 0.5,
        child: Image.network(
          widget.course.imageUrl,
          height: MediaQuery.of(context).size.height * 0.4,
          width: double.infinity,
          fit: BoxFit.cover,
        ),
      ),
      Container(
        height: MediaQuery.of(context).size.height * 0.4,
        width: double.infinity,
        decoration: BoxDecoration(
          color: Colors.black.withOpacity(0.8),
          gradient: const LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [
              Colors.transparent,
              Colors.black,
            ],
          ),
        ),
      ),
    ],
  ),
  Positioned(
    left: MediaQuery.of(context).size.width * 0.1,
    right: MediaQuery.of(context).size.width * 0.1,
    bottom: MediaQuery.of(context).size.width * 0.07,
    child: ClipRRect(
      borderRadius: BorderRadius.circular(10),
      child: Image.network(
        widget.course.imageUrl,
        width: MediaQuery.of(context).size.width * 0.8,
      ),
    ),
  ),
],
),
),
```

Don't forget to import the `MyColors` and `MyTypography` classes that are used in the `SliverAppBar` widget. They should be imported at the top of the file along with the other necessary imports.

```
import '../themes/colors.dart';
import '../themes/typography.dart';
```

If you successfully follow step 3, the view will look like the following screenshot:



Step 4 - Add body content (Course Info Section)

In step 4, we add the course information section to our `CustomScrollView` widget. This section includes the course title, total lesson, level, and duration. We use a

`SliverToBoxAdapter` widget to add a `Column` widget that contains these three pieces of information. We also use `buildCourseInfo` function to display the course duration as an icon and text.

Add the following code snippet to the `slivers` parameter list, right below the **Course Thumbnail Section** widget:

```
// Course Info
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 20,
    ),
    child: Column(
      children: [
        Text(
          widget.course.title,
          style: MyTypography.title,
        ),
        const SizedBox(height: 10),
        Row(
          children: [
            Wrap(
              crossAxisAlignment: WrapCrossAlignment.center,
              direction: Axis.horizontal,
              spacing: 20,
              runSpacing: 10,
              children: [
                ClipRRect(
                  borderRadius: BorderRadius.circular(15),
                  child: Container(
                    color: MyColors.primary.withOpacity(0.1),
                    padding: const EdgeInsets.symmetric(
                      horizontal: 10,
                      vertical: 5,
                    ),
                  ),
                child: Text(
                  widget.course.level,
                  style: MyTypography.bodySmall.copyWith(
                    color: MyColors.primary,
                  ),
                ),
              ],
            ),
            buildLessonInfo(
              Icons.library_books_outlined,
              "20 Lessons",
            ),
            buildLessonInfo(
```

```
Icons.av_timer,
    widget.course.duration,
),
],
),
],
),
],
),
),
),
```

The `buildCourseInfo` function takes two parameters: an `icon` and a `text`. It returns a `Row` widget containing an `Icon` widget and a `Text` widget. The `Icon` widget displays the provided icon, and the `Text` widget displays the provided text.

Here is the `buildCourseInfo` function:

```
Widget buildCourseInfo(IconData icon, String text) {
  return Row(
    children: [
      Icon(icon, size: 22, color: MyColors.black),
      const SizedBox(width: 5),
      Text(
        text,
        style: MyTypography.bodySmall,
      ),
    ],
  );
}
```

It is important to note that the `buildCourseInfo` function is located inside the `CourseDetailPage` widget class, along with the `build` method and other functions and variables used to build the page. This is because the function is specific to the `CourseDetailPage` and is not used in any other part of the app. Placing it inside the `CourseDetailPage` widget class keeps the code organized and easy to understand.

Populate the `course_detail_page.dart` file will look like this with `buildCourseInfo` function inside:

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

import '../models/course.dart';

class CourseDetailPage extends StatefulWidget {
```

```

const CourseDetailPage({super.key, required this.course});
final Course course;

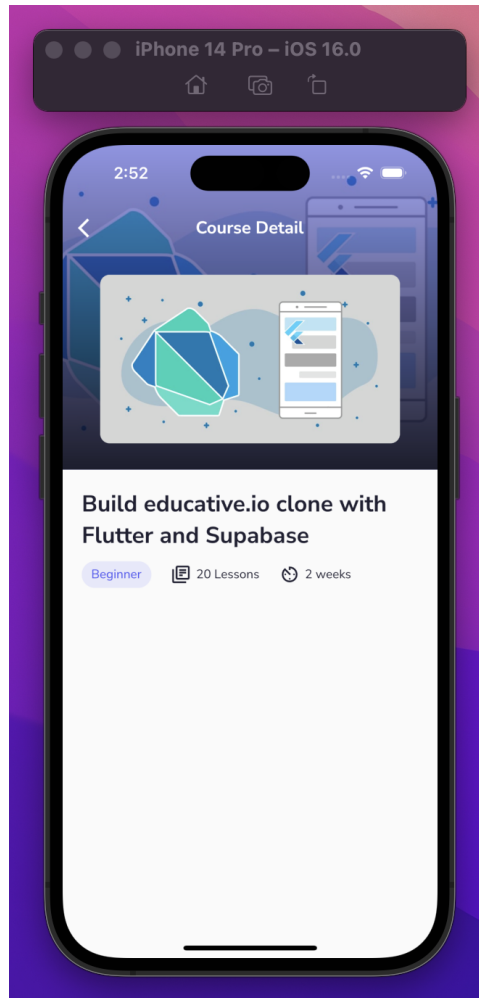
@override
State<CourseDetailPage> createState() => _CourseDetailPageState();
}

class _CourseDetailPageState extends State<CourseDetailPage> {
  @override
  Widget build(BuildContext context) {
    ...
  }

  // Build Course Info Widget
  Widget buildCourseInfo(IconData icon, String text) {
    return Row(
      children: [
        Icon(icon, size: 22, color: MyColors.black),
        const SizedBox(width: 5),
        Text(
          text,
          style: MyTypography.bodySmall,
        ),
      ],
    );
  }
}

```

If you successfully follow step 4, the view will look like the following screenshot:



Step 5 - Add body content (Course Description Section)

Next, we will add the course description section. We will use a `SliverToBoxAdapter` widget to add the course description to the `CustomScrollView` widget.

In this Step, we add the course description section using a `SliverToBoxAdapter` widget. We use the `ReadMoreText` widget to display the course description, which allows us to show a limited amount of text and then provide a "Show More" button if the text is too long. The `trimLength` parameter controls how much text is initially displayed, and the `trimMode` parameter controls how the text is trimmed. In this case, we are using `TrimMode.Length`, which truncates the text to a specified length. We also provide `trimCollapsedText` and `trimExpandedText` parameters to customize the text displayed on the "Show More" button.

Add the following code snippet to the `slivers` parameter list, right below the **Course Info Section** widget:

```

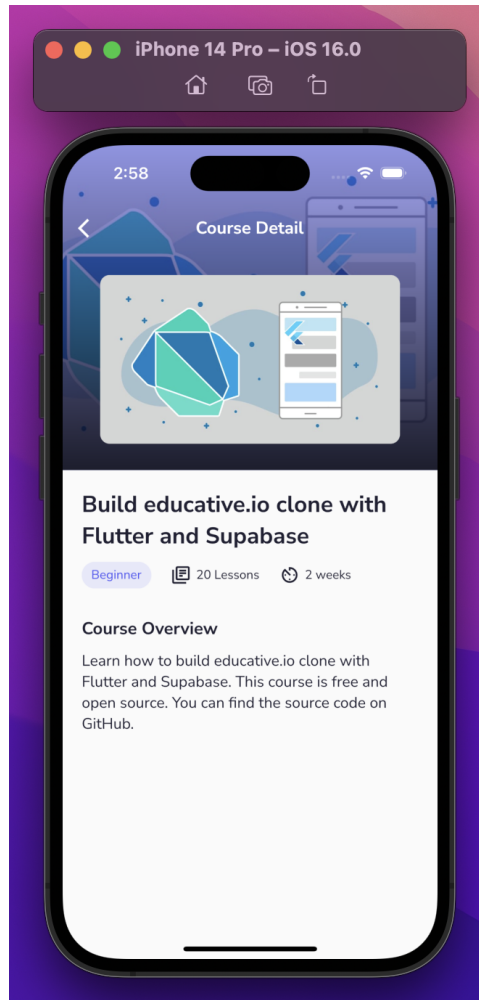
// Course Description
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 10,
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          'Course Overview',
          style: MyTypography.titleSmall,
        ),
        const SizedBox(height: 10),
        ReadMoreText(
          widget.course.description,
          style: MyTypography.body,
          trimMode: TrimMode.Length,
          trimCollapsedText: 'Show more',
          trimExpandedText: 'Show less',
          colorClickableText: MyColors.primary,
          trimLength: 180,
        ),
      ],
    ),
  ),
),

```

Don't forget to import `ReadMoreText` at the top of the file along with the other necessary imports.

```
import 'package:readmore/readmore.dart';
```

If you successfully follow step 5, the view will look like the following screenshot:



Step 6 - Add body content (Course Content Section)

After adding the course description section, we need to add the course content section.

Add the following code snippet to the `slivers` parameter list, right below the **Course Description Section** widget:

```
// Course Content
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 10,
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          'Course Contents',
```

```

                style: MyTypography.titleSmall,
            ),
            const SizedBox(height: 10),
            buildCourseContents(lessons),
        ],
    ),
),
),
),

```

The **Course Content** section is added to the `CustomScrollView` widget using a `SliverToBoxAdapter` widget. This section displays the course contents, and we use the `buildCourseContents` function to build the contents.

Step 6.1 - Define Lesson and LessonChild classes

To create the `buildCourseContents` function, we need to define a `Lesson` and `LessonChild` object or model.

To define the `Lesson` and `LessonChild` models, create a new file called `lesson.dart` inside the `lib/models` directory.

Add the following code snippet to define the `Lesson` and `LessonChild` classes in the `lib/models/lesson.dart` file:

```
class Lesson {
    final String id;
    final String title;
    final String courseId;
    final List<LessonChild> lessons;
    bool showDetail;

    Lesson({
        required this.title,
        required this.courseId,
        required this.id,
        required this.lessons,
        required this.showDetail,
    });

    factory Lesson.fromJson(Map<String, dynamic> json) {
        return Lesson(
            id: json['id'],
            title: json['title'],
            courseId: json['course_id'],
            lessons: json['lesson_child']
                .map<LessonChild>((lesson) => LessonChild.fromJson(lesson))
                .toList()
                .cast<LessonChild>(),
        );
    }
}
```

```

        showDetail: false,
    );
}
}

class LessonChild {
    final String id;
    final String title;
    final String content;
    bool? isCompleted;
    final String lessonId;
    final DateTime? createdAt;

    LessonChild({
        required this.id,
        required this.title,
        required this.content,
        required this.isCompleted,
        required this.lessonId,
        this.createdAt,
    });

    factory LessonChild.fromJson(Map<String, dynamic> json) {
        return LessonChild(
            id: json['id'],
            title: json['title'],
            content: json['content'],
            isCompleted: json['is_completed'] ?? false,
            lessonId: json['lesson_parent_id'],
            createdAt: DateTime.tryParse(json['created_at']),
        );
    }
}

```

Step 6.2 - Define the list of Lessons (dummy data)

Next, we create a `List` of `Lesson` objects to store the course contents.

To define a list of `Lesson` objects, navigate to the `lib/data/dummy_data.dart` file and insert the following code snippet below the variable list of `Course`.

```

// List of dummy lessons
final List<Lesson> lessons = [
    Lesson(
        title: 'Introduction',
        id: '',
        courseId: '',
        lessons: [
            LessonChild(
                id: '',

```

```

        title: 'Overview',
        content: '',
        isCompleted: false,
        lessonId: '',
        createdAt: DateTime.now(),
    ),
],
showDetail: false,
),
Lesson(
    title: 'Setting Up the Project',
    id: '',
    courseId: '',
    lessons: [
        LessonChild(
            id: '',
            title: 'Setting Up the Database',
            content: '',
            isCompleted: false,
            lessonId: '',
            createdAt: DateTime.now(),
        ),
    ],
    showDetail: false,
),
Lesson(
    title: 'Building the Backend',
    id: '',
    courseId: '',
    lessons: [
        LessonChild(
            id: '',
            title: 'Building the API',
            content: '',
            isCompleted: false,
            lessonId: '',
            createdAt: DateTime.now(),
        ),
    ],
    showDetail: false,
),
Lesson(
    title: 'Building the Frontend',
    id: '',
    courseId: '',
    lessons: [
        LessonChild(
            id: '',
            title: 'Building the UI',
            content: '',
            isCompleted: false,
            lessonId: '',
            createdAt: DateTime.now(),
        ),
    ],

```

```

    ],
    showDetail: false,
  ),
];

```

And don't forget to import `Lesson` and `LessonChild` classes at the top of the file along with the other necessary imports.

Populate the `lib/data/dummy_data.dart` will look like this:

```

import '../models/course.dart';
import '../models/lesson.dart'; // Add this line to import lesson models

// List of dummy courses
final List<Course> courses = [
  ...
];

// List of dummy lessons
final List<Lesson> lessons = [
  ...
];

```

Step 6.3 - Create the `buildCourseContents` function

The `buildCourseContents()` function in the code snippet you provided returns an `ExpansionPanelList` widget. This widget is used to create a list of expandable panels, where each panel contains a lesson title and duration. When a panel is clicked, it expands to show additional details about the lesson.

To create the `ExpansionPanelList`, the function takes in a list of `Lesson` objects as an input parameter, and then maps each lesson to an `ExpansionPanel` widget. The `ExpansionPanel` widget has a header and a body, which contain the lesson title and duration, respectively.

In the `buildCourseContents()` function, the header of each `ExpansionPanel` is wrapped in a `ListTile` widget, which provides a nice-looking title with a trailing icon that indicates whether the panel is expanded or collapsed. When the `ExpansionPanel` is expanded, the body of the panel is displayed, which shows additional details about the lesson, including a description and the lesson's content.

Overall, the `buildCourseContents()` function is a useful helper function for creating a list of expandable panels that display course content. It's a great way to organize and display

information about a course in a visually appealing and user-friendly way.

Here is the `buildCourseContents` function:

```
Widget buildCourseContents(List<Lesson> lessons) {
  return ExpansionPanelList(
    elevation: 0,
    expandedHeaderPadding: EdgeInsets.zero,
    expansionCallback: (panelIndex, isExpanded) {
      setState(() {
        lessons[panelIndex].showDetail = !isExpanded;
      });
    },
    children: lessons.map<ExpansionPanel>((Lesson lesson) {
      return ExpansionPanel(
        isExpanded: lesson.showDetail,
        canTapOnHeader: true,
        backgroundColor: Colors.transparent,
        headerBuilder: (context, isExpanded) {
          int index = lessons.indexOf(lesson) + 1;

          return ClipRRect(
            child: ListTile(
              leading: ClipRRect(
                borderRadius: BorderRadius.circular(15),
                child: Container(
                  color: MyColors.primary.withOpacity(0.1),
                  height: 25,
                  width: 25,
                  alignment: Alignment.center,
                  child: Text(
                    index.toString(),
                    style: MyTypography.bodySmall.copyWith(
                      color: MyColors.primary,
                    ),
                  ),
                ),
              ),
              minLeadingWidth: 0,
              title: Text(
                lesson.title,
                style: MyTypography.body,
              ),
            ),
          );
        },
        body: ListView.separated(
          padding: const EdgeInsets.symmetric(
            horizontal: 20,
            vertical: 16,
          ),
          shrinkWrap: true,
```



```

physics: const NeverScrollableScrollPhysics(),
itemCount: lesson.lessons.length,
separatorBuilder: (context, index) {
  return const Divider();
},
itemBuilder: (context, index) {
  return Row(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      if (lesson.lessons[index].isCompleted == true)
        const Icon(
          Icons.check_circle,
          size: 20,
          color: Colors.green,
        )
      else
        Icon(
          Icons.radio_button_unchecked,
          size: 20,
          color: MyColors.black,
        ),
      const SizedBox(width: 20),
      Expanded(
        child: Text(
          lesson.lessons[index].title,
          style: MyTypography.body,
        ),
      ),
    ],
  );
},
}).toList(),
);
}

```

It is important to note that the `buildCourseContents` function is located inside the `CourseDetailPage` widget class, along with the `build` method, `buildCourseInfo` function, and variables used to build the page. This is because the function is specific to the `CourseDetailPage` and is not used in any other part of the app.

Populate the `course_detail_page.dart` file will look like this with `buildCourseContents` function inside:

```

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

import '../data/dummy_data.dart'. // Add this line to import dummy data

```

```

import '../models/course.dart';
import '../models/lesson.dart';
import '../themes/colors.dart';
import '../themes/typography.dart';

class CourseDetailPage extends StatefulWidget {
  const CourseDetailPage({super.key, required this.course});
  final Course course;

  @override
  State<CourseDetailPage> createState() => _CourseDetailPageState();
}

class _CourseDetailPageState extends State<CourseDetailPage> {
  @override
  Widget build(BuildContext context) {
    ...
  }

  // Build Course Info Widget
  Widget buildCourseInfo(IconData icon, String text) {
    ...
  }

  // Build Course Contents Widget
  Widget buildCourseContents(List<Lesson> lessons) {
    ...
  }
}

```

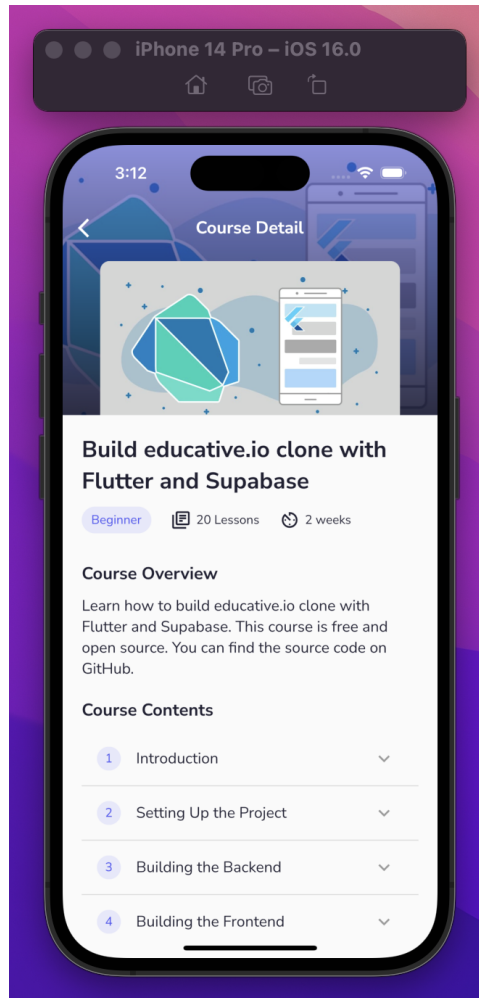
Make sure not to forget import `dummy_data.dart` file at the top of the file along with the other necessary imports on the `course_detail_page.dart` file.

```

import '../data/dummy_data.dart';

```

If you successfully follow step 6, the view will look like the following screenshot:



Step 7 - Add body content (Course Author Section)

Next, we need to add the course author section. We will use a `SliverToBoxAdapter` widget to add the course author information to the `CustomScrollView` widget.

Add the following code snippet to the `slivers` parameter list, right below the **Course Contents Section** widget:

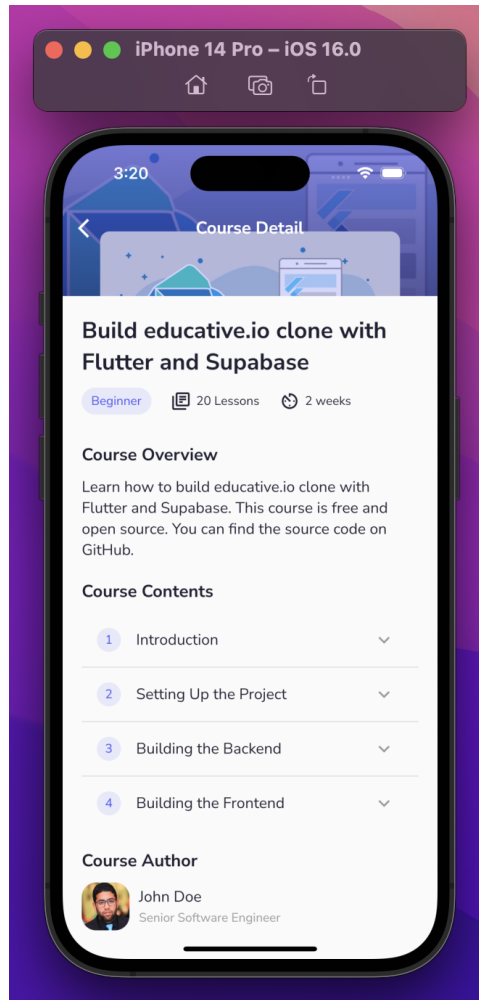
```
// Course Author
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 10,
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
```

```

Text(
  'Course Author',
  style: MyTypography.titleSmall,
),
const SizedBox(height: 10),
Row(
  children: [
    ClipRRect(
      borderRadius: BorderRadius.circular(15),
      child: Image.network(
        widget.course.authorAvatarUrl,
        width: 50,
        height: 50,
      ),
    ),
    const SizedBox(width: 10),
    Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          widget.course.authorName,
          style: MyTypography.body,
        ),
        const SizedBox(height: 2),
        Text(
          widget.course.authorJob,
          style: MyTypography.bodySmall.copyWith(
            color: Colors.grey,
          ),
        ),
      ],
    ),
  ],
),
const SizedBox(height: 20)
],
),
),
),
),

```

If you successfully follow step 7, the view will look like the following screenshot:



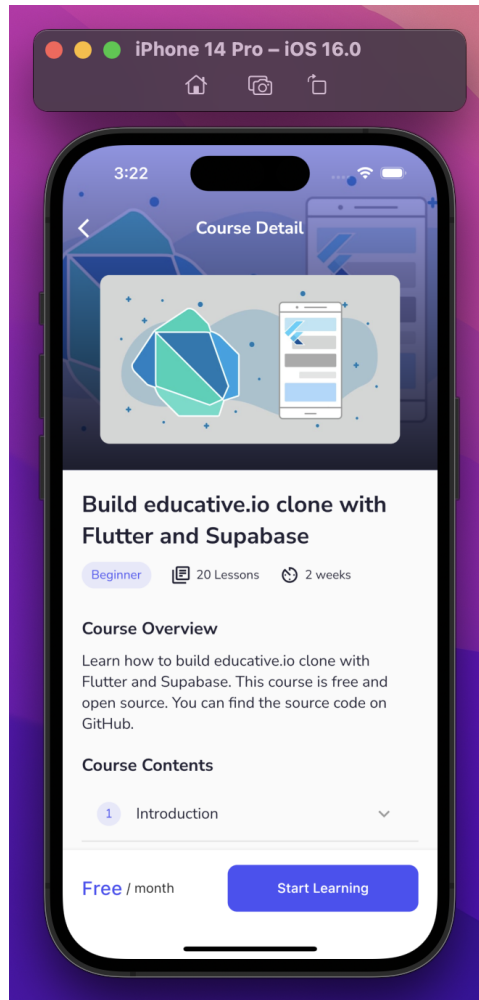
Step 8 - Add the bottom navigation bar

Finally, we need to add a bottom navigation bar to the `Scaffold` widget. To do this, we will use a `BottomAppBar` widget and add some additional widgets to it. The bottom navigation bar will include the course price and a button to start learning the course.

Overall, the bottom navigation bar is an important component of the course detail page, as it allows users to easily see the course price and navigate to the first lesson.

Here's the code snippet for the bottom navigation bar that needs to be added to the `Scaffold` widget:

```
// Bottom Nvaigation Bar
bottomNavigationBar: BottomAppBar(
  padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 10),
  child: SizedBox(
    height: 60,
```

Step 9 - Implementation of navigation between screens (HomePage → CourseDetailPage)

In the last step, we will implement the navigation between the `HomePage` and `CourseDetailPage`. When a user selects a course from the `HomePage`, they should be taken to the corresponding `CourseDetailPage`. To do this, we will need to make a few changes to the `HomePage` widget.

First, we need to use the `Navigator.push()` method to navigate to the `CourseDetailPage` when a user taps on a course.

Here's the code snippet that needs to be added to `onPressed` parameter of the `CourseCard` widget inside `lib/pages/home_page.dart` file:

```
Navigator.push(  
  context,
```

```

        MaterialPageRoute(
          builder: (context) => CourseDetailPage(
            course: courses[index],
          ),
        ),
      );

```

Here's the updated code for the `HomePage` widget in `lib/pages/home_page.dart` file:

```

import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:google_fonts/google_fonts.dart';

import '../data/dummy_data.dart';
import '../themes/colors.dart';
import '../widgets/course_card.dart';
import 'course_detail_page.dart'; // Import course_detail_page.dart file

class HomePage extends ConsumerWidget {
  const HomePage({super.key});

  @override
  Widget build(BuildContext context, WidgetRef ref) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.white,
        automaticallyImplyLeading: false,
        title: Row(
          children: [
            SvgPicture.asset(
              'assets/logo.svg',
              width: 24,
              color: MyColors.primary,
            ),
            const SizedBox(width: 5),
            Text(
              'educative',
              style: GoogleFonts.nunitoSans(
                color: MyColors.black,
                fontWeight: FontWeight.w500,
              ),
            ),
          ],
        ),
      ),
      actions: [
        IconButton(
          icon: const Icon(Icons.search),
          color: Colors.grey,
          onPressed: () {},
        ),
      ],
    );
  }
}

```



```

        IconButton(
          padding: const EdgeInsets.only(right: 10),
          icon: const Icon(Icons.account_circle),
          color: Colors.grey,
          iconSize: 30,
          onPressed: () {},
        ),
      ],
    ),
    body: ListView.separated(
      itemCount: courses.length,
      padding: const EdgeInsets.only(
        top: 20,
        left: 20,
        right: 20,
        bottom: 30,
      ),
      itemBuilder: (context, index) {
        return CourseCard(
          courses[index],
          onPressed: () {
            // Navigate to Course Detail Page
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => CourseDetailPage(
                  course: courses[index],
                ),
              ),
            );
          },
        );
      },
      separatorBuilder: (context, index) {
        return const SizedBox(height: 10);
      },
    ),
  );
}

```

Now when a user taps on a `CourseCard` widget, they will be taken to the corresponding `CourseDetailPage`.

Full `course_detail_page.dart` file

Here is the full code for the `CourseDetailPage` widget in the `course_detail_page.dart` file:

```

import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:readmore/readmore.dart';

import '../data/dummy_data.dart';
import '../models/course.dart';
import '../models/lesson.dart';
import '../themes/colors.dart';
import '../themes/typography.dart';

class CourseDetailPage extends ConsumerStatefulWidget {
  const CourseDetailPage({super.key, required this.course});
  final Course course;

  @override
  ConsumerState<CourseDetailPage> createState() => _CourseDetailPageState();
}

class _CourseDetailPageState extends ConsumerState<CourseDetailPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: CustomScrollView(
        slivers: [
          // Course Thumbnail
          SliverAppBar(
            backgroundColor: MyColors.primary,
            expandedHeight: MediaQuery.of(context).size.height * 0.33,
            elevation: 0,
            pinned: true,
            title: Text(
              "Course Detail",
              style: MyTypography.titleSmall.copyWith(
                color: Colors.white,
              ),
            ),
          ),
          flexibleSpace: FlexibleSpaceBar(
            background: Stack(
              children: [
                Opacity(
                  opacity: 0.5,
                  child: Image.network(
                    widget.course.imageUrl,
                    height: MediaQuery.of(context).size.height * 0.4,
                    width: double.infinity,
                    fit: BoxFit.cover,
                  ),
                ),
                Container(
                  height: MediaQuery.of(context).size.height * 0.4,
                  width: double.infinity,
                  decoration: BoxDecoration(

```

```

        color: Colors.black.withOpacity(0.8),
        gradient: const LinearGradient(
          begin: Alignment.topCenter,
          end: Alignment.bottomCenter,
          colors: [
            Colors.transparent,
            Colors.black,
          ],
        ),
      ),
    ),
    Positioned(
      left: MediaQuery.of(context).size.width * 0.1,
      right: MediaQuery.of(context).size.width * 0.1,
      bottom: MediaQuery.of(context).size.width * 0.07,
      child: ClipRRect(
        borderRadius: BorderRadius.circular(10),
        child: Image.network(
          widget.course.imageUrl,
          width: MediaQuery.of(context).size.width * 0.8,
        ),
      ),
    ),
  ],
),
),
),
),
// Course Info
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 20,
    ),
    child: Column(
      children: [
        Text(
          widget.course.title,
          style: MyTypography.title,
        ),
        const SizedBox(height: 10),
        Row(
          children: [
            Wrap(
              crossAxisAlignment: WrapCrossAlignment.center,
              direction: Axis.horizontal,
              spacing: 20,
              runSpacing: 10,
              children: [
                ClipRRect(
                  borderRadius: BorderRadius.circular(15),
                  child: Container(
                    color: MyColors.primary.withOpacity(0.1),
                    padding: const EdgeInsets.symmetric(

```

```

        horizontal: 10,
        vertical: 5,
      ),
      child: Text(
        widget.course.level,
        style: MyTypography.bodySmall.copyWith(
          color: MyColors.primary,
        ),
      ),
    ),
  ),
  buildCourseInfo(
    Icons.library_books_outlined,
    "20 Lessons",
  ),
  buildCourseInfo(
    Icons.av_timer,
    widget.course.duration,
  ),
],
),
],
),
],
),
),
// Course Description
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 10,
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          'Course Overview',
          style: MyTypography.titleSmall,
        ),
        const SizedBox(height: 10),
        ReadMoreText(
          widget.course.description,
          style: MyTypography.body,
          trimMode: TrimMode.Length,
          trimCollapsedText: 'Show more',
          trimExpandedText: 'Show less',
          colorClickableText: MyColors.primary,
          trimLength: 180,
        ),
      ],
    ),
  ),
),
),

```

```

),
// Course Content
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 10,
    ),
  ),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'Course Contents',
        style: MyTypography.titleSmall,
      ),
      const SizedBox(height: 10),
      buildCourseContents(lessons),
    ],
  ),
),
),
// Course Author
SliverToBoxAdapter(
  child: Padding(
    padding: const EdgeInsets.symmetric(
      horizontal: 20,
      vertical: 10,
    ),
  ),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'Course Author',
        style: MyTypography.titleSmall,
      ),
      const SizedBox(height: 10),
      Row(
        children: [
          ClipRRect(
            borderRadius: BorderRadius.circular(15),
            child: Image.network(
              widget.course.authorAvatarUrl,
              width: 50,
              height: 50,
            ),
          ),
          const SizedBox(width: 10),
          Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                widget.course.authorName,
                style: MyTypography.body,
              ),
            ],
          ),
        ],
      ),
    ],
  ),
),

```

```

        const SizedBox(height: 2),
        Text(
          widget.course.authorJob,
          style: MyTypography.bodySmall.copyWith(
            color: Colors.grey,
          ),
        ),
      ],
    ),
  ],
),
const SizedBox(height: 20)
],
),
),
],
),
// Bottom Navigation Bar
bottomNavigationBar: BottomAppBar(
  padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 10),
  child: SizedBox(
    height: 60,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Row(
          children: [
            Text(
              widget.course.price,
              style: Theme.of(context).textTheme.titleLarge?.copyWith(
                color: MyColors.primary,
              ),
            ),
            const Text(' / month'),
          ],
        ),
        SizedBox(
          width: 200,
          child: ElevatedButton(
            onPressed: () {
              // TODO: Navigate to Lesson Page
            },
            style: ElevatedButton.styleFrom(
              elevation: 0,
              padding: const EdgeInsets.symmetric(vertical: 16),
              minimumSize: const Size(double.infinity, 0),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10),
              ),
              backgroundColor: MyColors.primary,
            ),
            child: const Text('Start Learning'),
          ),
        ),
      ],
    ),
  ),
),

```

```

        ),
      ],
    ),
  ),
);
}

Widget buildCourseInfo(IconData icon, String text) {
  return Row(
    children: [
      Icon(icon, size: 22, color: MyColors.black),
      const SizedBox(width: 5),
      Text(
        text,
        style: MyTypography.bodySmall,
      ),
    ],
  );
}

Widget buildCourseContents(List<Lesson> lessons) {
  return ExpansionPanelList(
    elevation: 0,
    expandedHeaderPadding: EdgeInsets.zero,
    expansionCallback: (panelIndex, isExpanded) {
      setState(() {
        lessons[panelIndex].showDetail = !isExpanded;
      });
    },
    children: lessons.map<ExpansionPanel>((Lesson lesson) {
      return ExpansionPanel(
        isExpanded: lesson.showDetail,
        canTapOnHeader: true,
        backgroundColor: Colors.transparent,
        headerBuilder: (context, isExpanded) {
          int index = lessons.indexOf(lesson) + 1;

          return ClipRRect(
            child: ListTile(
              leading: ClipRRect(
                borderRadius: BorderRadius.circular(15),
                child: Container(
                  color: MyColors.primary.withOpacity(0.1),
                  height: 25,
                  width: 25,
                  alignment: Alignment.center,
                  child: Text(
                    index.toString(),
                    style: MyTypography.bodySmall.copyWith(
                      color: MyColors.primary,
                    ),
                  ),
                ),
            ),
          );
        },
      );
    },
  );
}

```

```

        ),
        minLeadingWidth: 0,
        title: Text(
            lesson.title,
            style: MyTypography.body,
        ),
    ),
);
},
body: ListView.separated(
    padding: const EdgeInsets.symmetric(
        horizontal: 20,
        vertical: 16,
    ),
    shrinkWrap: true,
    physics: const NeverScrollableScrollPhysics(),
    itemCount: lesson.lessons.length,
    separatorBuilder: (context, index) {
        return const Divider();
    },
    itemBuilder: (context, index) {
        return Row(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
                if (lesson.lessons[index].isCompleted == true)
                    const Icon(
                        Icons.check_circle,
                        size: 20,
                        color: Colors.green,
                    )
                else
                    Icon(
                        Icons.radio_button_unchecked,
                        size: 20,
                        color: MyColors.black,
                    ),
                const SizedBox(width: 20),
                Expanded(
                    child: Text(
                        lesson.lessons[index].title,
                        style: MyTypography.body,
                    ),
                ),
            ],
        );
    },
).toList(),
);
}
}

```


Testing the App

Let's test our app. Run the application on an emulator or device using the terminal command. The command `flutter run` will build the app and install it on your device.

<https://www.loom.com/share/6f7378eac2894490b89573d8a0f15e1d>

Conclusion

In this tutorial, we learned how to create a course detail page for an educative.io clone using Flutter. We covered topics such as building the UI with Slivers, creating a custom `ExpansionPanelList`, and navigating between screens. We also added features like a `ReadMoreText` widget and a bottom navigation bar. In the next section, we will create the lesson page. In the next section, we will create the lesson page, where users can navigate through the lessons of the course and mark their progress.

Happy coding!