Main.dart:

```
// Import Statements
import 'package:beyou/features/notes/notes.dart';

import 'features/notes/presentation/pages/home_page.dart';

// Main Function
void main() {
  runApp(const MyApp());
}

// Main App Class
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    // App Initialization

    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.deepOrange,
      ),
      // home: NoteDetailPage(
      //   note: Note(
      //       title: 'Account Info',
      //       description: 'email: sample@info.com\npass:newacc1290',
      //       category: 'personal',
      //       noteNumber: 1),
      // ));

      home: MyHomePage(),
```

```
    );
  }
}
```

Shared.dart:

```
export 'presentation/widgets/custom_text_field.dart';
export 'package:google_fonts/google_fonts.dart';
```

custom-field.dart:

```
import 'dart:ffi';

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

class custom_text_field extends StatelessWidget {
 final String label;
 final int? maxLines;
 final TextInputType? inputType;
 const custom_text_field({
  super.key,
  required this.titleController,
  required this.label,
  required this.maxLines,
  this.inputType,
 });

 final TextEditingController titleController;

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

```
    padding: const EdgeInsets.symmetric(
     vertical: 8.0,
    ),
    child: TextField(
     controller: titleController,
     maxLines: maxLines,
     style: GoogleFonts.poppins(
      fontWeight: FontWeight.w400,
      color: Colors.black54,
     ),
     keyboardType: inputType,
     decoration: InputDecoration(
      constraints: const BoxConstraints(maxHeight: 55),
      counterStyle: const TextStyle(color: Colors.black54),
      filled: true,
      fillColor: Colors.white,
      labelText: label,
      focusColor: Colors.black54,
      labelStyle: GoogleFonts.poppins(
        fontSize: 13, fontWeight: FontWeight.w400, color: Colors.black54),
     ),
    ),
   );
 }
}
```

Home.dart:

```
// Home Page Class
import 'package:flutter/cupertino.dart';
```

```
import '../../../../shared/shared.dart';
import '../../notes.dart';

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

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

class _MyHomePageState extends State<MyHomePage> with TickerProviderStateMixin {
  late final TextEditingController descriptionController;

  // State Variables
  List<Note> notes = [];
  int selectedFilterIndex = 0;
  List<String> filterItems = [];

  @override
  Widget build(BuildContext context) {
    // Main Scaffold
    return Scaffold(
      backgroundColor: Colors.black,
      appBar: CustomAppBar(
        bgColors: Colors.black,
        onProfilePressed: () {
          // Handle profile icon click
        },
        onAddPressed: () {
```

```
  //?detailed page navigation to be
},
leftData: const Icon(CupertinoIcons.person_fill),
RightData: const Icon(CupertinoIcons.add),
centerwidget: ScaleTransition(
 scale: Tween<double>(
  begin: 1.0,
  end: 1.1,
 ).animate(CurvedAnimation(
  parent: AnimationController(
   vsync: this,
   duration: const Duration(milliseconds: 150),
  ),
  curve: Curves.easeInOut,
 )),
 child: Row(
  crossAxisAlignment: CrossAxisAlignment.center,
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
   Text(
    "Morning,",
    style: GoogleFonts.raleway(
     fontSize: 16,
     fontWeight: FontWeight.w400,
    ),
   ),
   Text(
    " James",
    style: GoogleFonts.raleway(
     fontSize: 16,
     fontWeight: FontWeight.bold,
```

```
        ),
       ),
      ],
     ),
    ),
   ),
  ),
  body: Column(
   crossAxisAlignment: CrossAxisAlignment.start,
   children: [
    const LineBreak(color: Colors.white),
    YourNotes(notes: '/${notes.length}'),
    Padding(
     padding: const EdgeInsets.all(8.0),
     child: FilterList(
      filterItems: filterItems,
      selectedFilterIndex: selectedFilterIndex,
      onFilterSelected: (index) {
       setState(() {
        selectedFilterIndex = index;
       });
      },
     ),
    ),
    const SizedBox(height: 20),
    Expanded(
     child: ListView.builder(
      physics: const BouncingScrollPhysics(),
      itemCount: notes.isEmpty ? 1 : notes.length,
      itemBuilder: (context, index) {
       if (notes.isEmpty) {
        // If the list is empty, display a funny message
```

```
return Column(
 mainAxisAlignment: MainAxisAlignment.center,
 crossAxisAlignment: CrossAxisAlignment.center,
 children: [
  Padding(
    padding: EdgeInsets.all(16.0),
    child: NotePreviewWidget(
     note: Note(
       title: 'Account Info',
       description:
         'email: sample@info.com\npass:newacc1290',
       category: 'personal',
       noteNumber: 1),
     onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(
         builder: (context) => NoteDetailPage(
           note: Note(
             title: 'Account Info',
             description:
               'email: sample@info.com\npass:newacc1290',
             category: 'personal',
             noteNumber: 1)),
        ));
     },
    )),
  LineBreak(
   color: Colors.white70,
  )
 ],
```

```
                );
              } else {
                // If the list is not empty, display the NotePreviewWidget
                return NotePreviewWidget(
                  note: notes[index],
                  onTap: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => NoteDetailPage(
                          note: notes[index],
                        ),
                      ),
                    );
                  },
                );
              }
            },
          ),
        ),
      ],
    ),
  );
 }
}
```

Note.dart:

```
// ignore_for_file: sized_box_for_whitespace

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

```
import 'package:speech_to_text/speech_to_text.dart' as stt;

import '../../notes.dart';

class NoteDetailPage extends StatefulWidget {
 final Note note;

 const NoteDetailPage({Key? key, required this.note}) : super(key: key);

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

class _NoteDetailPageState extends State<NoteDetailPage> {
 final stt.SpeechToText _speech = stt.SpeechToText();
 String voiceInput = '';

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   resizeToAvoidBottomInset: false,
   backgroundColor: Colors.white,
   appBar: CustomAppBar(
    bgColors: Colors.white,
    onProfilePressed: () {
     // Handle profile icon click
    },
    onAddPressed: () {},
    leftData: GestureDetector(
     onTap: () {
      Navigator.pop(context);
```

```
    },
    child: const Icon(
      CupertinoIcons.back,
      color: Colors.black,
      size: 20,
    ),
  ),
  RightData: Padding(
    padding: const EdgeInsets.only(
      right: 8.0,
    ),
    child: Text(
      'delete',
      style: GoogleFonts.poppins(
        fontSize: 13,
        fontWeight: FontWeight.w300,
        color: Colors.black,
      ),
    ),
  ),
  centerwidget: Container(),
),
body: Column(
  mainAxisAlignment: MainAxisAlignment.start,
  crossAxisAlignment: CrossAxisAlignment.center,
  children: [
    const LineBreak(color: Colors.black),
    const SizedBox(height: 20),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 15.0),
      child: Row(
```

```
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
     Text(
      '18/11/24',
      style: GoogleFonts.poppins(
       color: Colors.black54,
       fontSize: 13,
      ),
     ),
     Text(
      '#${widget.note.category}',
      style: GoogleFonts.poppins(
       color: Colors.black54,
       fontSize: 13,
      ),
     )
    ],
   ),
  ),
 ),
 const SizedBox(height: 45),
 Container(
  child: Column(
    children: [
     SizedBox(
      height: MediaQuery.of(context).size.height * 0.35,
      child: EditableTextWidget(
       initialTitle: widget.note.title,
       initialDescription: widget.note.description,
      ),
     ),
     SizedBox(
```

```
height: MediaQuery.of(context).size.height * 0.35,
child: Column(
 mainAxisAlignment: MainAxisAlignment.center,
 children: [
  GestureDetector(
   onTap: () {
    _startListening();
   },
   child: Container(
     height: 50,
     width: 50,
     decoration: BoxDecoration(
       color: Colors.deepOrange,
       borderRadius: BorderRadius.circular(50)),
     child: const Center(
      child: Icon(
       CupertinoIcons.mic,
       color: Colors.white,
       size: 30,
      ),
     )),
  ),
  const SizedBox(height: 20),

  Text('Voice Input: $voiceInput',
    style: GoogleFonts.poppins(
     color: Colors.black54,
     fontWeight: FontWeight.w500,
     fontSize: 15,
    )), // Display real-time voice input
 ],
```

```
            ),
          ),
        ],
      ),
    ),
    const SizedBox(height: 20),

    // Speech to text feature
  ],
 ),
);
}

void _startListening() async {
 bool available = await _speech.initialize(
  onStatus: (status) {
   print('Speech recognition status: $status');
  },
  onError: (errorNotification) {
   print('Speech recognition error: $errorNotification');
  },
 );

 if (available) {
  _speech.listen(
   onResult: (result) {
    setState(() {
     voiceInput = result.recognizedWords ?? '';
    });
   },
   listenFor: const Duration(minutes: 5), // Set duration to 5 minutes
```

```
      );
    } else {
      print('Speech recognition not available');
    }
  }
}
```