Main.dart

```
import 'package:flutter/material.dart';
import 'package:todo_app/screens/home.dart';
import 'package:todo_app/splashscreen.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Todo List',
      themeMode: ThemeMode.system,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      debugShowCheckedModeBanner: false,
      darkTheme: ThemeData(
        brightness: Brightness.dark,
      ),
      home: const SplashScreen(),
      routes: {
        '/home': (context) => const HomeScreen(),
        '/splash': (context) => const SplashScreen(),

      }
    );
  }
}
```

Splashscreen.dart :

```
import 'dart:async';

import 'package:flutter/material.dart';

import 'package:todo_app/screens/home.dart';



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

  @override
  Widget build(BuildContext context) {
    Timer(const Duration(seconds: 2), () {
      Navigator.of(context).pushReplacement(MaterialPageRoute(
        builder: (context) => const HomeScreen(),
      ));
    });

    return Scaffold(
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Image.asset(
                  "assets/images/icon-transparent.png",
                width: 100,
                height: 100,
              ),
            // const SizedBox(height: 20,),
            const Text(
                "...Simplifying your workflows",
              style: TextStyle(fontSize: 16.0, fontWeight: FontWeight.bold, color:
Colors.deepPurpleAccent,),
            ),
          ],
        ),
      ),
    );
  }
}
```

Home.dart :

```
import 'package:flutter/material.dart';
import 'package:todo_app/auth/login.dart';
import 'package:todo_app/screens/setting.dart';
import 'package:todo_app/screens/todoscreen.dart';
import 'package:todo_app/screens/notescreen.dart';

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  int _currentIndex = 0;

  // Define a list of screens to be displayed in the bottom navigation bar
  final List<Widget> _screens = [
    const ToDoScreen(),
    const NotesScreen(),
    const SettingsScreen(),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: _screens[_currentIndex],
      drawer: Drawer(
        child: ListView(
          padding: EdgeInsets.zero,
          children: [
            const DrawerHeader(
              decoration: BoxDecoration(
                color: Colors.deepPurple,
              ),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  CircleAvatar(
                    radius: 30,
                    backgroundImage: AssetImage('assets/images/icon.png'),
                  ),
                  SizedBox(height: 10),
                  Text(
                    'John Doe',
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 20,
                    ),
                  ),
     Text(
      "testemail@flutter.com"
   ),
    ],
   ),
  ),
  ListTile(
   leading: const Icon(Icons.home),
   title: const Text('Home'),
   onTap: () {
    Navigator.push(context,
     MaterialPageRoute(builder: (context) => const LoginScreen()),);
   },
  ),
  ListTile(
   leading: const Icon(Icons.person),
   title: const Text('Profile'),
   onTap: () {
    // Navigate to profile screen
    Navigator.pop(context);
   },
  ),
  ListTile(
   leading: const Icon(Icons.settings),
   title: const Text('Settings'),
   onTap: () {
    // Navigate to settings screen
    Navigator.pop(context);

   },
  ),
     ListTile(
      leading: const Icon(Icons.info),
      title: const Text('About'),
      onTap: () {
       // Navigate to about screen
       Navigator.pop(context);
      },
     ),
     const Divider(),
     ListTile(
      title: const Text('Log out'),
      onTap: () {
       // Log out logic
       Navigator.pop(context);
      },
     ),
    ],
   ),
  ),
  bottomNavigationBar: BottomNavigationBar(
   currentIndex: _currentIndex,
   onTap: (index) {
    setState(() {
     _currentIndex = index;
    });
   },
   items: const [
    BottomNavigationBarItem(
     icon: Icon(Icons.check_box),
     label: 'Todo',
     tooltip: 'Todo',
     ),
     BottomNavigationBarItem(
      icon: Icon(Icons.note),
      label: 'Notes',
      tooltip: 'Notes',
     ),
     BottomNavigationBarItem(
      icon: Icon(Icons.settings),
      label: 'Settings',
      tooltip: 'Settings',
     )
    ],
   ),
  );
 }
}
```

Notes.dart:

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

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

 @override
 State<NotesScreen> createState() => _NotesScreenState();
}

class _NotesScreenState extends State<NotesScreen> {
 List<String> notes = [
  'Note 1: This is the first note',
    'Note 2: This is the second note',
    'Note 3: This is the third note',
  ];

  void _addNote() {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => const NoteEditorScreen(),
      ),
    ).then((newNote) {
      if (newNote != null) {
        setState(() {
          notes.add(newNote);
        });
      }
    });
  }

  void _editNote(int index) {
    TextEditingController titleController = TextEditingController(text: notes[index].split(':')[0]);
    TextEditingController bodyController = TextEditingController(text: notes[index].split(':')[1]);

    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => NoteEditorScreen(titleController: titleController, bodyController:
bodyController),
      ),
    ).then((editedNote) {
      if (editedNote != null) {
```

```
   setState(() {
    notes[index] = editedNote;
   });
  }
 });
}

@override
Widget build(BuildContext context) {
 return Scaffold(
  appBar: AppBar(
   title: const Row(
    children: [
     Icon(FontAwesomeIcons.penToSquare),
     SizedBox(width: 8,),
     Text('Notes')
    ],
   ),
   systemOverlayStyle: SystemUiOverlayStyle.light,
   centerTitle: true,
   backgroundColor: Colors.deepPurple[400],
   elevation: 0,
   actions: [
    IconButton(
     icon: const Icon(Icons.search),
     onPressed: () {
      // Implement search functionality
     },
    ),
   ],
  ),
  body: notes.isEmpty
    ? const Center(
   child: Text(
    'No notes yet. Tap the + button to add a new note.',
    textAlign: TextAlign.center,
   ),
  )
    : ListView.builder(
   itemCount: notes.length,
   itemBuilder: (context, index) {
    final note = notes[index];
    return ListTile(
     title: Card(
      elevation: 3.0,
      margin: const EdgeInsets.symmetric(vertical: 8.0, horizontal: 16.0),
      shape: RoundedRectangleBorder(
       borderRadius: BorderRadius.circular(12.0),
      ),
      child: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
         Expanded(
          child: Text(
           note.split(':')[0],
           style: const TextStyle(fontSize: 16.0),
          ),
         ),
         IconButton(
          icon: const Icon(Icons.edit),
             onPressed: () {
              _editNote(index);
             },
            ),
            IconButton(
             icon: const Icon(Icons.delete),
             onPressed: () {
              setState(() {
               notes.removeAt(index);
              });
             },
            ),
           ],
          ),
         ),
        ),
       );
      },
     ),
     floatingActionButton: FloatingActionButton(
      onPressed: _addNote,
      child: const Icon(Icons.add),
     ),
    );
   }
  }

  class NoteEditorScreen extends StatelessWidget {
   final TextEditingController? titleController;
   final TextEditingController? bodyController;
```

```
const NoteEditorScreen({super.key, this.titleController, this.bodyController});

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Note Editor'),
      backgroundColor: Colors.deepPurple[400],
      elevation: 0,
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: [
          TextField(
            controller: titleController,
            decoration: const InputDecoration(
              hintText: 'Title',
              border: InputBorder.none,
            ),
            keyboardType: TextInputType.text,
            style: const TextStyle(fontSize: 24.0, fontWeight: FontWeight.bold),
          ),
          const SizedBox(height: 16.0),
          Expanded(
            child: TextField(
              controller: bodyController,
              decoration: const InputDecoration(
                hintText: 'Write your note...',
                border: InputBorder.none,
        ),
        keyboardType: TextInputType.multiline,
        maxLines: null,
        style: const TextStyle(fontSize: 16.0),
       ),
      ),
     ],
    ),
   ),
   floatingActionButton: FloatingActionButton(
    onPressed: () {
     Navigator.of(context).pop('${titleController?.text ?? ''}: ${bodyController?.text ?? ''}');
    },
    child: const Icon(Icons.save),
   ),
  );
 }

}
```

Todo.dart:

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

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

 @override
 State<ToDoScreen> createState() => _ToDoScreenState();
```

```
}

final player = AudioPlayer();
class _ToDoScreenState extends State<ToDoScreen> {
 List<Todo> todos = [
  Todo(
   title: 'Add more Todo by pressing + button',
   creationDate: DateTime.now(),
  ),
  Todo(
   title: 'Tap on a todo to edit it',
   creationDate: DateTime.now(),
  ),
  Todo(
   title: 'Swipe left or right to delete',
   creationDate: DateTime.now(),
  ),
  Todo(
   title: 'Check the todo checkbox to mark complete',
   creationDate: DateTime.now(),
   isCompleted: true,
   completionDate: DateTime.utc(2024, 01, 24, 4, 11, 48),
  ),
 ];
 void _addTodo() {
  TextEditingController controller = TextEditingController();
  showDialog(
   context: context,
   builder: (BuildContext context) {
    return AlertDialog(
     title: const Text('Add Todo'),
        content: TextField(
          controller: controller,
          decoration: const InputDecoration(hintText: 'Enter your todo'),
        ),
        actions: <Widget>[
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: const Text('Cancel'),
          ),
          TextButton(
            onPressed: () {
              setState(() {
                String title = controller.text;
                if (title.isNotEmpty) {
                  todos.add(Todo(
                    title: title,
                    creationDate: DateTime.now(),
                  ));
                }
              });
              Navigator.of(context).pop();
            },
            child: const Text('Add'),
          ),
        ],
      );
    },
  );
}
```

```
void _deleteTodo(int index) {
  setState(() {
    todos.removeAt(index);
  });
}
// void _completeTodo(int index) async {
//   await player.play('assets/sounds/tink.mp3' as Source);
// }
void _editTodo(int index) {
  TextEditingController controller = TextEditingController();
  controller.text = todos[index].title;
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: const Text('Edit Todo'),
        content: TextField(
          controller: controller,
          decoration: const InputDecoration(hintText: 'Enter your todo'),
        ),
        actions: <Widget>[
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: const Text('Cancel'),
          ),
          TextButton(
            onPressed: () {
              setState(() {
                String newTitle = controller.text;
        if (newTitle.isNotEmpty) {
         todos[index].title = newTitle;
        }
       });
       Navigator.of(context).pop();
      },
      child: const Text('Save'),
     ),
    ],
   );
  },
 );
}
@override
Widget build(BuildContext context) {
 return Scaffold(
  appBar: AppBar(
   centerTitle: true,
   backgroundColor: Colors.deepPurple[400],
   elevation: 0,
   actions: [
    IconButton(
     icon: const Icon(Icons.search),
     onPressed: () {
      // search functionality
     },
    ),
    IconButton(
     icon: const Icon(Icons.filter_alt),
     onPressed: () {
      // filter functionality
      },
    ),
  ],
  title: const Row(
    children: [
      Icon(FontAwesomeIcons.listCheck),
      SizedBox(width: 8),
      Text('To Do'),
    ],
  ),
),
body: todos.isEmpty
    ? Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      const Text(
        'No todos yet!',
        style: TextStyle(fontSize: 18),
      ),
      ElevatedButton(
        onPressed: _addTodo,
        child: const Text('Add Todo'),
      ),
    ],
  ),
)
    : Column(
  children: [
    Expanded(
      child: ListView.builder(
itemCount: todos.length,
itemBuilder: (context, index) {
 final todo = todos[index];
 return Dismissible(
  key: Key(todo.title),
  onDismissed: (direction) {
   _deleteTodo(index);
  },
  background: Container(
   color: Colors.red,
   alignment: Alignment.centerRight,
   padding: const EdgeInsets.only(right: 20.0),
   child: const Icon(Icons.delete, color: Colors.white),
  ),
  child: Card(
   elevation: 3.0,
   margin: const EdgeInsets.symmetric(vertical: 10.0),
   child: ListTile(
    title: Column(
     crossAxisAlignment: CrossAxisAlignment.start,
     children: [
      Text(
       todo.title,
       style: TextStyle(
        fontSize: 18.0,
        color: Colors.deepPurpleAccent,
        decoration: todo.isCompleted
          ? TextDecoration.lineThrough
          : null,
       ),
      ),
  const SizedBox(height: 8),
  Text(
   'Created: ${todo.creationDate.toString().split('.')[0]}',
   style: const TextStyle(
    fontSize: 14.0,
    color: Colors.grey,
   ),
  ),
  if (todo.isCompleted &&
    todo.completionDate != null)
   Text(
    'Completed: ${todo.completionDate.toString().split('.')[0]}',
    style: const TextStyle(
     fontSize: 14.0,
     color: Colors.grey,
    ),
   ),
 ],
),
leading: Checkbox(
 value: todo.isCompleted,
 onChanged: (bool? value) {
  setState(() {
   todo.isCompleted = value!;
   if (value == true) {
    // _completeTodo();
    todo.completionDate = DateTime.now();
   } else {
    todo.completionDate = null;
   }
  });
```

```
       },
      ),
      onTap: () {
       _editTodo(index);
      },
     ),
    ),
   );
  },
 ),
),
Container(
 padding: const EdgeInsets.all(8.0),
 color: Theme.of(context).brightness == Brightness.light
   ? Colors.grey[300]
   : Colors.grey[700],
 child: Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
   const Icon(Icons.check_circle, color: Colors.green),
   const SizedBox(width: 8),
   Text(
    '${todos.where((todo) => !todo.isCompleted).length} Remaining',
    style: TextStyle(
     fontSize: 16,
     color: Theme.of(context).brightness == Brightness.light
       ? Colors.black
       : Colors.white,
    ),
   ),
  ],
            ),
          ),
        ],
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _addTodo,
        tooltip: 'Add Todo',
        child: const Icon(Icons.add),
      ),
    );
  }
}
class Todo {
  String title;
  bool isCompleted;
  DateTime creationDate;
  DateTime? completionDate;

  Todo({
    required this.title,
    this.isCompleted = false,
    required this.creationDate,
    this.completionDate,
  });
}
```