

ДОДАТОК А

home_page.dart:

```
appBar: AppBar(  
  backgroundColor: Colors.grey[900],  
  centerTitle: true,  
  title: (user == null)  
    ? const Text(  
      'Welcome to Ebook',  
      style: TextStyle(fontFamily: 'e-Ukraine', fontSize: 30),  
    )  
    : const Text(  
      'Welcome Back',  
      style: TextStyle(fontFamily: 'e-Ukraine', fontSize: 30),  
    ),  
),
```

search_screen.dart:

```
import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'dart:math' as math;  
  
import '../constant/constants.dart';  
import '../notifiers/app_notifier.dart';  
import '../models/book.dart';  
import 'detail_screen.dart';
```

```

class CustomSearchDelegate extends SearchDelegate {
  final _random = math.Random();

  String errorLink =
    "https://img.freepik.com/free-vector/funny-error-404-background-
design_1167-
219.jpg?w=740&t=st=1658904599~exp=1658905199~hmac=131d690585e96267b
bc45ca0978a85a2f256c7354ce0f18461cd030c5968011c";

  @override
  List<Widget>? buildActions(BuildContext context) {
    // TODO: implement buildActions
    return null;
  }

  @override
  Widget? buildLeading(BuildContext context) {
    // TODO: implement buildLeading

    return IconButton(
      onPressed: () {
        close(context, null);
      },
      icon: Icon(Icons.arrow_back_ios),
    );
  }

  @override

```



```

        boxColor: Colors.black,
    ),
),
);
},
leading: Image.network(
    snapshot.data?.items?[index].volumeInfo?.imageLinks
        ?.thumbnail ??
        errorLink,
),
title: Text(
    "${snapshot.data?.items![index].volumeInfo!.authors?.length != 0 ?
snapshot.data?.items![index].volumeInfo!.authors![0] : "Not Found"}",
    style: const TextStyle(color: Colors.grey),
),
subtitle: Text(
    "${snapshot.data?.items![index].volumeInfo?.title}",
    style: const TextStyle(color: Colors.white),
),
),
);
},
);
}
return const Center(
    child: CircularProgressIndicator(

```

```
        color: Colors.grey,
    ),
);
},
);
},
);
}
```

@override

```
Widget buildSuggestions(BuildContext context) {
  return Consumer<AppNotifier>(
    builder: (context, value, child) {
      return FutureBuilder(
        future: value.searchBookData(searchBook: "Biography"),
        builder: (context, AsyncSnapshot<Books> snapshot) {
          if (snapshot.hasError) {
            return const Center(
              child: Text("Opps! Try again later!"),
            );
          }
          if (snapshot.hasData) {
            return ListView.builder(
              itemCount: 10,
              itemBuilder: (context, index) {
```

```

return Container(
  color: Colors.grey[900],
  child: ListTile(
    style: ListTileStyle.list,
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => DetailsScreen(
            id: snapshot.data?.items?[index].id,
            boxColor: Colors.grey,
          )),
        ),
    leading: Image.network(
      "${snapshot.data?.items?[index].volumeInfo?.imageLinks?.thumb
      nail ?? errorLink}",
    ),
    title: Text(
      style: TextStyle(color: Colors.grey),
      "${snapshot.data?.items![index].volumeInfo!.authors?.length != 0
      ? snapshot.data?.items![index].volumeInfo!.authors![0] : "Not Found"}",
    ),
    subtitle: Text(
      "${snapshot.data?.items![index].volumeInfo?.title}",
      style: TextStyle(color: Colors.white),
    ),
  ),
)

```

```
    );  
  },  
);  
}  
return const Center(  
  child: CircularProgressIndicator(  
    color: Colors.grey,  
  ),  
);  
},  
);  
},  
);  
}  
}
```

adventure.dart:

```
ListView.builder(  
  shrinkWrap: true,  
  physics: const BouncingScrollPhysics(),  
  scrollDirection: Axis.horizontal,  
  itemCount: 10,  
  itemBuilder: (context, index)
```

categories.dart:

```
GridView.builder(  
  physics: const BouncingScrollPhysics(),  
  itemCount: categoriesName.length,  
  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
    childAspectRatio: 16 / 15,  
    crossAxisCount: 2,  
    crossAxisSpacing: 20,  
    mainAxisSpacing: 20),  
  itemBuilder: (context, index)
```

main.dart:

```
Future<void> main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);  
  runApp(const MyApp());  
}
```

sign_up_page.dart:

```
Future<void> _signUp() async {  
  final navigator = Navigator.of(context);  
  
  final isValid = formKey.currentState!.validate();  
  if (!isValid) return;  
  if (_passwordTextInputController.text !=  
    _passwordTextRepeatInputController.text) {  
    SnackBarService.showSnackBar(  
      context,
```



```
'Passwords should be the same',
true,
);
return;
}
try {
await FirebaseAuth.instance.createUserWithEmailAndPassword(
  email: _emailTextEditingController.text.trim(),
  password: _passwordTextEditingController.text.trim(),
);
} on FirebaseAuthException catch (e) {
  // ignore: avoid_print
  print(e.code);
  if (e.code == 'email-already-in-use') {
    SnackbarService.showSnackBar(
      context,
      'This E-mail is already in use',
      true,
    );
    return;
  } else {
    SnackbarService.showSnackBar(
      context,
      'Unknown error,please contact customer support',
      true,
```

```
);  
}  
}  
navigator.pushNamedAndRemoveUntil(  
  FirebaseStream.routeName,  
  (route) => false,  
);  
}
```