

flutter pub add firebase\_storage

flutter pub add file\_picker

flutter build apk --release

**Note: all functions are inside \_MyHomePageState class**

main.dart:

A screenshot of a code editor with a dark background and light-colored text. The editor shows the main.dart file content. At the top left, there are three colored circles (red, yellow, green) representing window controls. The code is as follows:

```
1 void main() async {  
2   WidgetsFlutterBinding.ensureInitialized();  
3  
4   await Firebase.initializeApp(  
5     options: DefaultFirebaseOptions.currentPlatform,  
6   );  
7  
8   runApp(const MyApp());  
9 }  
10  
11 class MyApp extends StatelessWidget {  
12   const MyApp({super.key});  
13  
14   @override  
15   Widget build(BuildContext context) {  
16     return const MaterialApp(  
17       debugShowCheckedModeBanner: false,  
18       home: MyHomePage(),  
19     );  
20   }  
21 }
```

home\_screen.dart:

build function (1):

```
1
2 class MyHomePage extends StatefulWidget {
3   const MyHomePage({super.key});
4
5   @override
6   State<MyHomePage> createState() => _MyHomePageState();
7 }
8
9 class _MyHomePageState extends State<MyHomePage> {
10   List<String> fileLinks = [];
11   List<String> fileNames = [];
12
13   var storage = FirebaseStorage.instance;
14   var db = FirebaseDatabase.instance;
15
16   @override
17   Widget build(BuildContext context) {
18     return SafeArea(
19       child: Scaffold(
20         body: SingleChildScrollView(
21           child: Padding(
22             padding: const EdgeInsets.all(20),
23             child: Column(
24               children: [
25
```

build function (2)

```
1 // buttons
2
3 Row(
4   mainAxisAlignment: MainAxisAlignment.spaceEvenly,
5   children: [
6     // upload file button
7     ElevatedButton(
8       onPressed: () async {
9         await uploadFile();
10      },
11      child: const Text('Upload file')),
12
13    // retrieve links button
14    ElevatedButton(
15      onPressed: () async {
16        await retrieveLinks();
17      },
18      child: const Text('Retrieve links')),
19  ],
20 ),
21
22 const SizedBox(height: 40),
```

### build function (3)

```
1 // file urls list
2 if (fileLinks.isNotEmpty)
3   ListView.separated(
4     shrinkWrap: true,
5     physics: const NeverScrollableScrollPhysics(),
6     itemBuilder: (context, index) {
7       return Column(
8         children: [
9           // file name + delete button
10          Row(
11            children: [
12              // file name
13              Expanded(
14                flex: 8,
15                child: Text(
16                  fileNames[index],
17                  maxLines: 1,
18                  overflow: TextOverflow.ellipsis,
19                ),
20            ],
21            // delete button
22            Expanded(
23              flex: 2,
24              child: IconButton(
25                onPressed: () async {
26                  deleteLink(index);
27                },
28                icon: const Icon(
29                  Icons.delete,
30                  color: Colors.red,
31                ),
32            ),
33          ],
34        ),
35      ],
36    ),
37    // url
38    Text(
39      fileLinks[index],
40      // maxLines: 3,
41      // overflow: TextOverflow.fade,
42      style: const TextStyle(
43        color: Colors.blue,
44        decoration: TextDecoration.underline,
45      ),
46    ),
47  ],
48 ],
49 );
50 },
51 separatorBuilder: (context, index) {
52   return const SizedBox(height: 20);
53 },
54 itemCount: fileLinks.length,
55 ),
56 ],
57 ),
58 ),
59 ),
60 ),
61 );
62 }
```

Upload file function:

```
1 Future<void> uploadFile() async {
2     var result = await FilePicker.platform.pickFiles();
3
4     // if user picked a file
5     if (result != null) {
6         File file = File(result.files.single.path!);
7         String fileName = result.files.single.name;
8
9         // upload to storage
10        storage.ref('files/$fileName').putFile(file).then(
11            (snapshot) async {
12                String url = await snapshot.ref.getDownloadURL();
13
14                // write to database
15                db.ref('files').push().set(url).then((_) async {
16                    // add link to list
17                    fileLinks.add(url);
18                    fileNames.add(fileName);
19
20                    // update UI
21                    setState(() {});
22                });
23            },
24        );
25    }
26 }
```

Retrieve links function:

```
1 Future<void> retrieveLinks() async {
2     try {
3         final ListResult listResult = await storage.ref('files').listAll();
4
5         for (var ref in listResult.items) {
6             String url = await ref.getDownloadURL();
7             fileLinks.add(url);
8             fileNames.add(ref.name);
9         }
10
11         // update UI
12         setState(() {});
13     } catch (e) {}
14 }
```

Delete link function:

```
1 Future<void> deleteLink(int index) async {
2     try {
3         final snapshot = await db.ref('files').get();
4
5         if (snapshot.exists) {
6             var data = snapshot.value as Map<dynamic, dynamic>;
7
8             data.forEach(
9                 (key, value) async {
10                     if (value == fileLinks[index]) {
11                         // delete from database
12                         await db.ref('files').child(key).remove();
13
14                         // delete from storage
15                         await storage.refFromURL(fileLinks[index]).delete();
16
17                         // delete from List
18                         fileLinks.removeAt(index);
19                         fileNames.removeAt(index);
20
21                         // update UI
22                         setState(() {});
23                     }
24                 },
25             );
26         }
27     } catch (error) {}
28 }
29 }
```

لا تنسونا من صالح الدعاء