

```
1 //
2 // JokeFinderApp.swift
3 // JokeFinder
4 //
5
6 import SwiftUI
7
8 @main
9 struct JokeFinderApp: App {
10     var body: some Scene {
11         WindowGroup {
12             BrowsingView()
13         }
14     }
15 }
16
17
18 //
19 // Functions.swift
20 // JokeFinder
21 //
22
23 import Foundation
24
25 func filtering(
26     listOfSavedJokes: [SavedJoke],
27     on searchText: String
28 ) -> [SavedJoke] {
29
30     if searchText.isEmpty {
31         return listOfSavedJokes
32     }
33
34     var filteredListOfJokes: [SavedJoke] = []
35
36     for savedJoke in listOfSavedJokes {
37
38         if savedJoke.comments.lowercased().contains(searchText.lowercased()) ||
39             savedJoke.joke.setup.lowercased().contains(searchText.lowercased()) ||
40             savedJoke.joke.punchline.lowercased().contains(searchText.lowercased())
41         {
42
43             filteredListOfJokes.append(savedJoke)
44
45         }
46     }
47
48     return filteredListOfJokes
49 }
50
51 }
52
53
54 //
55 // Joke.swift
56 // JokeFinder
57 //
58
59 import Foundation
```

```

60
61 struct Joke: Identifiable, Decodable {
62     var type: String
63     var setup: String
64     var punchline: String
65     var id: Int
66 }
67
68 let first = Joke(
69     type: "general",
70     setup: "You see, mountains aren't just funny.",
71     punchline: "They are hill areas.",
72     id: 408
73 )
74
75 let second = Joke(
76     type: "general",
77     setup: "How do you get two whales in a car?",
78     punchline: "Start in England and drive West.",
79     id: 126
80 )
81
82 let third = Joke(
83     type: "general",
84     setup: "Why did the invisible man turn down the job offer?",
85     punchline: "He couldn't see himself doing it.",
86     id: 52
87 )
88
89
90 //
91 // SavedJoke.swift
92 // JokeFinder
93 //
94
95 import Foundation
96
97 struct SavedJoke: Identifiable {
98     let id = UUID()
99     let joke: Joke
100    let comments: String
101 }
102
103 let firstSavedJoke = SavedJoke(
104     joke: first,
105     comments: "Use when hanging out with Geography teachers!"
106 )
107
108 let secondSavedJoke = SavedJoke(
109     joke: second,
110     comments: "Another great joke to share with Geography teachers."
111 )
112
113 let thirdSavedJoke = SavedJoke(
114     joke: third,
115     comments: "This one is a bit of riddle; remember to try it out on younger cousins in the family."
116 )
117
118 let exampleSavedJokes = [firstSavedJoke, secondSavedJoke, thirdSavedJoke]

```

```

119
120
121 //
122 // JokeSource.json
123 // JokeFinder
124 //
125
126 [
127     {"type":"general","setup":"What did the fish say when it hit the wall?","punchline":"Dam.","id":1},
128     {"type":"general","setup":"How do you make a tissue dance?","punchline":"You put a little boogie on it.","id":2},
129     {"type":"general","setup":"What's Forrest Gump's password?","punchline":"1Forrest1","id":3},
130     {"type":"general","setup":"What do you call a belt made out of watches?","punchline":"A waist of time.","id":4},
131     {"type":"general","setup":"Why can't bicycles stand on their own?","punchline":"They are two tired","id":5},
132     {"type":"general","setup":"How does a train eat?","punchline":"It goes chew, chew","id":6},
133     {"type":"general","setup":"What do you call a singing Laptop?","punchline":"A Dell","id":7},
134     {"type":"general","setup":"How many lips does a flower have?","punchline":"Tulips","id":8},
135     {"type":"general","setup":"How do you organize an outer space party?","punchline":"You planet","id":9},
136     {"type":"general","setup":"What kind of shoes does a thief wear?","punchline":"Sneakers","id":10},
137     {"type":"general","setup":"What's the best time to go to the dentist?","punchline":"Tooth hurty.","id":11},
138     {"type":"programming","setup":"Why did the programmer quit his job?","punchline":"Because he didn't get arrays.","id":18},
139     {"type":"general","setup":"Did you hear about the two silk worms in a race?","punchline":"It ended in a tie.","id":19},
140     {"type":"general","setup":"What do you call a laughing motorcycle?","punchline":"A Yamahahahaha.","id":20},
141     {"type":"programming","setup":"How many programmers does it take to change a lightbulb?","punchline":"None that's a hardware problem","id":25},
142     {"type":"programming","setup":"['hip', 'hip']","punchline":"(hip hip array)","id":27},
143     {"type":"general","setup":"What happens to a frog's car when it breaks down?","punchline":"It gets toad away","id":31},
144     {"type":"general","setup":"Did you know the first French fries weren't cooked in France?","punchline":"It's true, they were cooked in Greece.","id":32},
145     {"type":"general","setup":"What do you call a factory that sells passable products?","punchline":"A satisfactory","id":50},
146 ]
147
148 //
149 // HumorRetrievalService.swift
150 // JokeFinder
151 //
152
153 struct HumorRetrievalService {
154
155     static func fetchNewJoke() -> Joke {
156
157         let fileName = "JokeSource"
158         let fileExtension = ".json"
159
160         guard let urlForFileThatContainsJokes = Bundle.main.url(forResource: fileName, withExtension: fileExtension) else {
161             fatalError("Failed to locate jokes source file in application bundle.")
162         }
163
164         do {
165
166             let data = try Data(contentsOf: urlForFileThatContainsJokes)
167
168             let decoder = JSONDecoder()
169
170             let jokes = try decoder.decode([Joke].self, from: data)
171
172             return jokes.randomElement()!
173
174         } catch {
175
176             print(error)
177             fatalError("Error loading data from jokes source file in application bundle.")

```

```
178
179     }
180
181 }
182
183 }
184
185
186 //
187 // BrowsingView.swift
188 // JokeFinder
189 //
190
191 import SwiftUI
192
193 struct BrowsingView: View {
194
195     @State var currentJoke: Joke = HumorRetrievalService.fetchNewJoke()
196
197     @State var punchlineShowing = false
198
199     @State var searchText = ""
200
201     @State var providedComments = ""
202
203     @State var history: [SavedJoke] = []
204
205     var body: some View {
206         HStack {
207             VStack(alignment: .leading) {
208
209                 HStack {
210                     Text(currentJoke.setup)
211                         .fontWeight(.heavy)
212                         .padding(.bottom, 40)
213                         .font(.largeTitle)
214
215                     Spacer()
216                 }
217
218                 Text(currentJoke.punchline)
219                     .padding(.bottom, 40)
220                     .font(.largeTitle)
221                     .opacity(punchlineShowing == true ? 1.0 : 0.0)
222
223                 Spacer()
224
225                 HStack {
226
227                     Button(action: {
228                         resetAndGetNewJoke()
229                     }, label: {
230                         Text("New Joke")
231                     })
232
233                     Button(action: {
234                         punchlineShowing = true
235                     }, label: {
236                         Text("Show Punchline")
```

```

237         })
238         .disabled(punchlineShowing == true)
239     }
240     .padding(.bottom, 10)
241
242     Divider()
243
244     TextField("Add notes: when and where might you use this joke?",
245             text: $providedComments,
246             axis: .vertical)
247         .lineLimit(4...8)
248         .disabled(punchlineShowing == false)
249
250     HStack {
251         Spacer()
252
253         Button(action: {
254             saveJoke()
255         }, label: {
256             Text("Save")
257         })
258         .disabled(providedComments.isEmpty || punchlineShowing == false)
259     }
260
261 }
262 .padding(.horizontal, 30)
263 .padding(.vertical, 40)
264
265 VStack {
266     List(
267         filtering(
268             listOfSavedJokes: history,
269             on: searchText
270         )
271     ) { savedJoke in
272         ListItemView(toShow: savedJoke)
273     }
274     .searchable(text: $searchText)
275 }
276
277 }
278
279 }
280
281
282
283
284
285 func resetAndGetNewJoke() {
286     punchlineShowing = false
287
288     currentJoke = HumorRetrievalService.fetchNewJoke()
289
290 }
291
292
293 func saveJoke() {
294     let jokeToSave = SavedJoke(
295

```

```

296         joke: currentJoke,
297         comments: providedComments
298     )
299
300     history.append(jokeToSave)
301
302     providedComments = ""
303
304 }
305 }
306
307 #Preview {
308     BrowsingView()
309         .frame(width: 800, height: 600)
310 }
311
312
313 //
314 // ListItemView.swift
315 // JokeFinder
316 //
317
318 import SwiftUI
319
320 struct ListItemView: View {
321     let toShow: SavedJoke
322
323
324     var body: some View {
325         VStack(alignment: .leading, spacing: 5) {
326             HStack {
327                 Text(toShow.joke.setup)
328                     .font(.title2)
329                     .bold()
330                 Spacer()
331             }
332
333             Text(toShow.joke.punchline)
334                 .font(.title2)
335
336             HStack(alignment: .top, spacing: 5) {
337                 Text("Notes:")
338                     .font(Font.subheadline.smallCaps())
339
340                 Text(toShow.comments)
341                     .font(.subheadline)
342             }
343                 .padding(.top, 10)
344         }
345         .padding()
346     }
347 }
348 }
349
350 #Preview {
351     ListItemView(toShow: firstSavedJoke)
352         .frame(width: 400, height: 300)
353 }
354

```