

Exemplar Code:

```
247 ▾ get "/adding_bookmarks" do
248     erb :adding_bookmarks
249 end
250
251 ▾ post "/adding_bookmarks" do
252     @validation = true
253     @duplicate=false
254     time = Time.new
255
256     @title = params[:title]
257     @content = params[:content]
258     @description = params[:description]
259     @author = params[:author]
260     @author_id=session[:id]
261
262     @date = (time.day.to_s + "/" + time.month.to_s + "/" + time.year.to_s)
263     @rating = 0
264     @num_rating = 0
265     @reported = 0
266
267     #If bookmark details have been entered
268 ▾ if (@title != '' && @title) &&
269     (@content != '' && @content) &&
270     (@description != '' && @description)
271     #If the title of the bookmark is unique
272 ▾ if(!Bookmark.duplicate(@title,$db))
273     Bookmark.new(@title, @content, @description, @author,@author_id, @date, @rating, @num_rating, @reported, $db)
274     redirect "/"
275 ▾ else
276     @duplicate=true
277     erb :adding_bookmarks
278     end
279 ▾ else
280     @validation = false
281     erb :adding_bookmarks
282     end
283 end
```

Listing A: app.rb

```
4 ▾ module Bookmark
5
6     # Create new bookmark
7 ▾ def Bookmark.new(title, content, description, author,author_id, date, rating, num_rating, reported, db)
8
9 ▾     query= "INSERT INTO bookmark(title,content,description,author,author_id,date_created,rating,num_of_ratings,reported)
10         VALUES(?,?,?,?,?,?,?,?)"
11
12     result = db.execute query, title, content, description, author,author_id, date, rating, num_rating, reported
13
14 end
15
16 # Return all bookmarks
17 ▾ def Bookmark.find_all(db)
18     result = []
19     query = "SELECT title,author,date_created,rating,num_of_ratings,reported,bookmark_id FROM bookmark;"
20     rows = db.execute query
21 ▾     rows.each do |row|
22         result.push({title: row[0], author: row[1], date: row[2], rating: row[3],num_of_rate: row[4],reported: row[5],id: row[6]})
23     end
24
25     return result
26 end
```

Listing B: models/bookmark.rb

```

11 --bookmark table
12 CREATE TABLE bookmark (
13     --primary key for table
14     bookmark_id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
15
16     title TEXT NOT NULL,
17     content TEXT NOT NULL,
18     description TEXT,
19     author TEXT NOT NULL,
20     author_id INTEGER NOT NULL,
21     date_created DATE NOT NULL,
22     rating INTEGER,
23     num_of_ratings INTEGER,
24     reported BOOLEAN NOT NULL
25 );

```

Listing C: create_databases.sql

```

12 class TestStringComparison < Minitest::Test
13
14     $db = SQLite3::Database.new 'database/test_class_database.sqlite';
15
16     # Compares methods returned hash values by outputting methods to console and checking this output
17     # (assert_equal currently not working for hash values)
18     def test_find_search
19         assert_output(/, '') do
20             setup_test_find_search("");
21         end
22         assert_output(/{:id=>1, :firstname=>"Logan", :surname=>"Miller", :access_level=>"admin", :suspended=>0}\n/, '') do
23             setup_test_find_search("LOGAN");
24         end
25         assert_output(/{:id=>1, :firstname=>"Logan", :surname=>"Miller", :access_level=>"admin", :suspended=>0}\n/, '') do
26             setup_test_find_search("logan");
27         end
28         assert_output(/{:id=>1, :firstname=>"Logan", :surname=>"Miller", :access_level=>"admin", :suspended=>0}\n/, '') do
29             setup_test_find_search("oga");
30         end
31     end
32
33     # Outputs result of Users.find_all so that the returned hash value can be checked
34     # (assert_equal currently not working for hash values)
35     def setup_test_find_search(search)
36         puts Users.find_search(search, $db);
37     end

```

Listing D: user_test_class.rb

```
3 Scenario: Go to adding bookmark page
4   Given I am on the login page
5   When I fill in "email" with "jamesa@gmail.com"
6   When I fill in "password" with "pWORD1"
7   When I press "Login" within ".contentSmallForm"
8   When I press "Add Bookmarks" within ".contentSmallForm"
9   Then I should be on the adding bookmark page
```

Listing E: features/add_bookmark.feature