

Exemplar Code:

```
302 ▾ get "/adding_bookmarks" do
303   # List all the tags in the drop-down list
304   @list=Bookmark.find_tags($db)
305   erb :adding_bookmarks
306 end
307
308 ▾ post "/adding_bookmarks" do
309   @validation = true
310   @duplicate=false
311   @same_tag=false
312   time = Time.new
313
314   @title = params[:title]
315   @content = params[:content]
316   @description = params[:description]
317   @author = params[:author]
318   @author_id=session[:id_l]
319
320   @date = (time.year.to_s + "-" + time.month.to_s + "-" + time.day.to_s)
321   @rating = 0
322   @num_rating = 0
323   @reported = 0
324
325   # Add tags
326   if(params[:select_tag1]!="null")
327     @tag1=params[:select_tag1]
328   end
329
330   if(params[:select_tag2]!="null")
331     @tag2=params[:select_tag2]
332   end
333
334   if(params[:select_tag3]!="null")
335     @tag3=params[:select_tag3]
336   end
337   ---
338
339   # If all bookmark details have been entered
340   if (@title != '' && @title) &&
341     (@content != '' && @content) &&
342     (@description != '' && @description)
343
344     # If the title of the bookmark is unique
345     if(!Bookmark.duplicate(@title,$db))
346
347       # If there are duplicated tags
348       if(!Bookmark.same_tag(@tag1,@tag2,@tag3,$db))
349
350         # Create a new bookmark
351         Bookmark.new(@title, @content, @description, @author,@author_id, @date, @rating, @num_rating,
352                     @reported,@tag1,@tag2,@tag3, $db)
353         redirect "/"
354
355       else
356         @same_tag=true
357         @list=Bookmark.find_tags($db)
358         erb :adding_bookmarks
359       end
360
361     else
362       @duplicate=true
363       @list=Bookmark.find_tags($db)
364       erb :adding_bookmarks
365     end
366
367   else
368     @validation = false
369     @list=Bookmark.find_tags($db)
370     erb :adding_bookmarks
371   end
372 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

```

411 ▾ post "/view_bookmarks/filter" do
412     @choice=params[:filter]
413
414     # Filter due to the choice
415 ▾   if @choice=="rate"
416       session[:filter_r]=true
417       session[:filter_d]=false
418       session[:filter_re]=false
419 ▾   elsif @choice=="date"
420       session[:filter_r]=false
421       session[:filter_d]=true
422       session[:filter_re]=false
423 ▾   else
424       session[:filter_r]=false
425       session[:filter_d]=false
426       session[:filter_re]=true
427 ▾   end
428     redirect '/view_bookmarks'
429 end
430
431 ▾ post "/view_bookmarks/reset" do
432     # Reset the filter
433     session.delete(:filter_r)
434     session.delete(:filter_d)
435     session.delete(:filter_re)
436     redirect '/view_bookmarks'
437 end

```

Listing F: app.rb

```

374 ▾ get "/view_bookmarks" do
375     # No search, show all the bookmarks due to the filters
376 ▾   if session[:search_bm]==false||!session[:search_bm]
377       @list = Bookmark.find_all(session[:filter_r],session[:filter_d], session[:filter_re],$db)
378 ▾   else
379       # Show different items due to content of search due to the search type and the filters
380       @list = Bookmark.find_search(session[:filter_r],session[:filter_d], session[:filter_re],session[:result_bm],session[:search_by], $db)
381     end
382     erb :view_bookmarks
383 end
384
385 ▾ post "/view_bookmarks/back" do
386     # Reset when leave the page
387     session.delete(:search_bm)
388     session.delete(:search_by)
389     session.delete(:filter_r)
390     session.delete(:filter_d)
391     session.delete(:filter_re)
392     redirect '/view_bookmarks'
393 end
394
395 ▾ post "/view_bookmarks" do
396     session[:search_bm]=true
397     session[:search_by]=params[:search_by]
398     session[:result_bm] = params[:search]
399
400     @no_results=false
401     # Find all the bookmarks related to the search
402 ▾   @list = Bookmark.find_search( session[:filter_r],session[:filter_d], session[:filter_re],session[:result_bm],
403                                   session[:search_by],$db)
404     # Type nothing or something invalid
405 ▾   if (@list==[])
406       @no_results=true
407     end
408     erb :view_bookmarks
409 end

```

Listing G : app.rb