

```

51
52 #attempted login
53 post '/authenticate-user' do
54
55     @login = params[:user_email]
56     @password = params[:user_password]
57
58     #check credentials
59     @result = UserAuthentication.check @login , @password
60
61     if @result == -1
62         @error = "Invalid login or password"
63     else
64         #check if account is active and verified
65         if (UserAuthentication.getAccessLevel @result) == 0
66             @details = Bookmarks.getAccessDetails @result
67             #if not check why
68             if @details[:user_suspended] == 1
69                 redirect '/msg?msg=suspendedMsg'
70             elsif @details[:user_type] == Bookmarks::UNVERIFIED_STRING
71                 redirect '/msg?msg=unverifiedMsg'
72             end
73         elsif UserAuthentication.hasPasswordReset(@login)
74             redirect "/change-password?userID=#{@result}"
75         else
76             session[:userID] = @result
77             redirect '/'
78         end
79     end
80
81     erb:loginpage
82
83 end
84

```

Figure 1 Snippet A (bookmark-site.rb)

```

132
133 # Search handling.
134 get '/search' do
135
136     @searchQuery = params["search_query"]
137
138     @fullResults = Bookmarks.getHomepageData @searchQuery
139
140     #variables for tagControl
141     @tagList = Bookmarks.getTagNames
142     #get tags for filter either from submitted form
143     #or redirected after clicking on a tag in bookmark details
144     @checked = params[:showTag] ? [params[:showTag]] : (extractTagsFromParams params)
145
146     if(@fullResults.length != 0) then
147         @results = filterAgainstTags @fullResults, @checked
148         erb :search
149     else
150         erb :noResults
151     end
152
153 end
154

```

Figure 2 Snippet B (bookmark-site.rb)

```

223
224 #User submitted rating
225 post '/bookmark-spesifics' do
226
227   if Bookmarks.resourceExists? params[:bookmarkID], "bookmark"
228     @userID = session[:userID]
229     @bookmarkID = params[:bookmarkID]
230     @value = params[:selection] #submitted rating
231
232     if Bookmarks.isInteger @value then
233
234       #Either add or update the rating the user has already submitted
235       if Bookmarks.getRating(@bookmarkID, @userID) == nil then
236         success = addRating @bookmarkID, @userID, @value
237       else
238         success = changeRating @bookmarkID, @userID, @value
239       end
240
241       if success then
242         redirect '/msg?msg=ratingAddedMsg'
243       end
244     end
245
246     redirect '/msg?msg=actionErrorMsg'
247   else
248     redirect '/msg?msg=missingResourceMsg'
249   end
250 end
251
252

```

Figure 3 Code Snippet C (bookmark-site.rb)

```

488
489 get '/reported-bookmarks' do
490   if (UserAuthentication.getAccessLevel session[:userID]) == 2 #Is current user an admin
491     @reportList = Bookmarks.getUnresolvedReports
492     if @reportList.length() > 0 then
493       @reportTable = erb :reportTable, :locals => {:reportList => @reportList}
494       erb :reportedBookmarks
495     else
496       redirect '/msg?msg=noReportsMsg'
497     end
498   else
499     redirect '/'
500   end
501 end
502

```

Figure 4 Code Snippet D (bookmark-site.rb)

```

735
736 # Insert user's details into db when they register an account
737 def Bookmarks.addRegisterDetails (uEmail, uDisplay, uDepartment, password)
738   if !Bookmarks.isUniqueValue('users','user_email',uEmail) then
739     return false
740
741   elsif uEmail.nil? then
742     return false
743   else
744     query = "INSERT INTO users(user_email, user_displayName, user_department,
745       user_password, user_type, user_suspended)
746       VALUES (?, ?, ?, ?, ?, ?);"
747     @@db.execute query, uEmail, uDisplay, uDepartment, BCrypt::Password.create(password), UNVERIFIED_STRING, 0
748     return true
749   end
750 end
751

```

Figure 5 Snipped E (ruby-scripts/database-model.rb)

```

1054
1055     #Change bookmark details
1056     def Bookmarks.updateBookmark (bookmarkID, bookmarkTitle, bookmarkDesc, bookmarkLink)
1057         if bookmarkTitle.nil? then
1058             return false
1059         else
1060             query = "UPDATE bookmark
1061                     SET bookmark_title = ?, bookmark_description = ?, bookmark_link = ?
1062                     WHERE bookmark_ID = ?;"
1063             @@db.execute query, bookmarkTitle, bookmarkDesc, bookmarkLink, bookmarkID
1064             return true
1065         end
1066     end
1067

```

Figure 6 Snippet F (ruby-scripts/database-model.rb)

```

675
676     #Checks if value passed exists in a database
677     #Params tableName - name of the table in which to look for uniqueness
678     #      columnName - name of the column in which to check uniqueness
679     #      value - value to be looked for
680     #Returns: true - if value doesn't exist in a given column in given table
681     #      false - if it does
682     #      nil -if given column or table name are incorrect
683     def Bookmarks.isUniqueValue tableName , columnName , value
684         if (Bookmarks.getColumnNames tableName).include? columnName
685             query = "SELECT DISTINCT #{columnName}
686                     FROM #{tableName} ;"
687             result = @@db.execute query
688
689             (0..(result.length()-1)).each do |i|
690                 if result[i][columnName] == value
691                     return false
692                 end
693             end
694             return true
695         end
696         return nil
697     end
698
699

```

Figure 7 Snippet G (ruby-scripts/database-model.rb)

```

121
122     def filterAgainstTags bookmarks, tags
123         if !tags || tags.length == 0
124             return bookmarks
125         end
126         result = Array.new
127         bookmarks.each do |bookmark|
128             result<<bookmark if (intersection((Bookmarks.getBookmarkTagsNames bookmark[:ID].to_i), tags).length > 0)
129         end
130
131         return result
132     end
133

```

Figure 8 Snippet H (ruby-scripts/support-functions.rb)