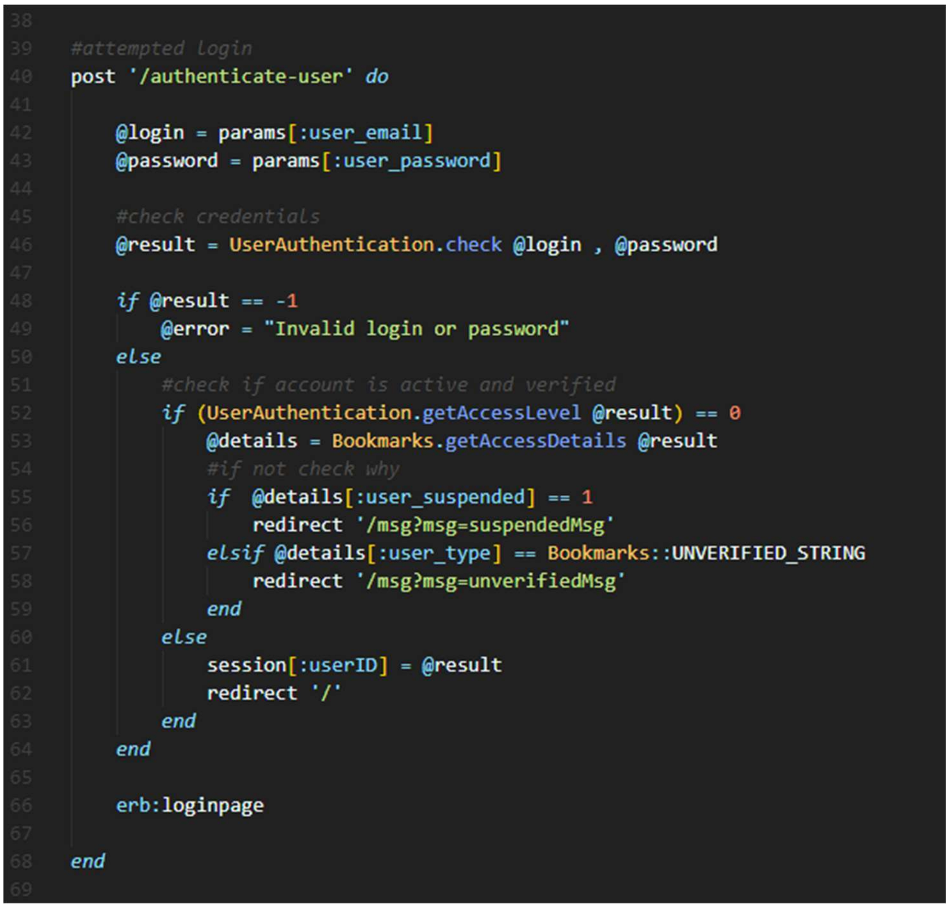
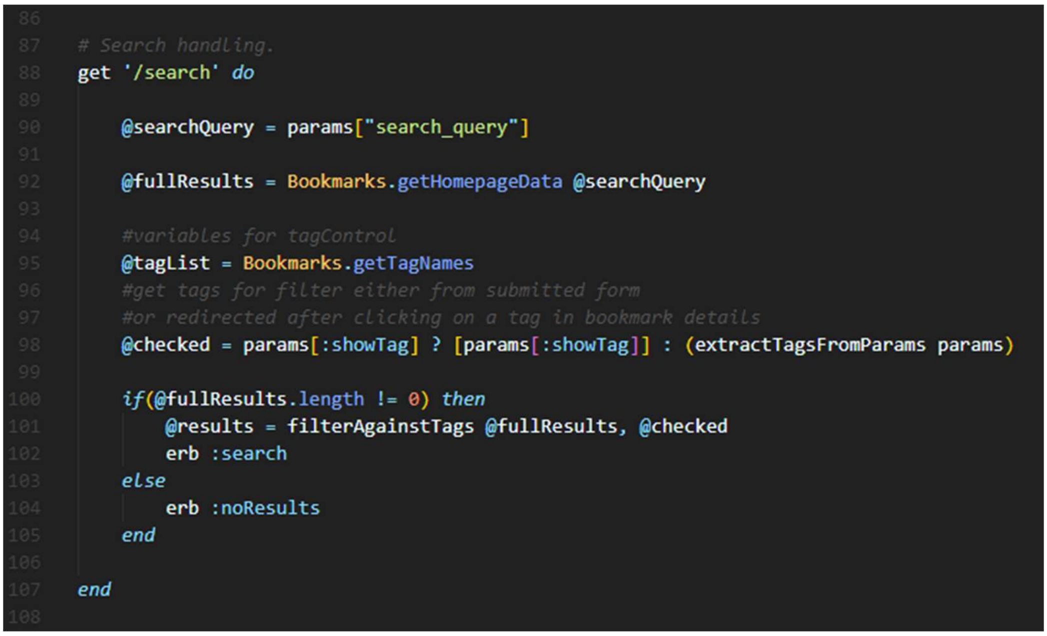


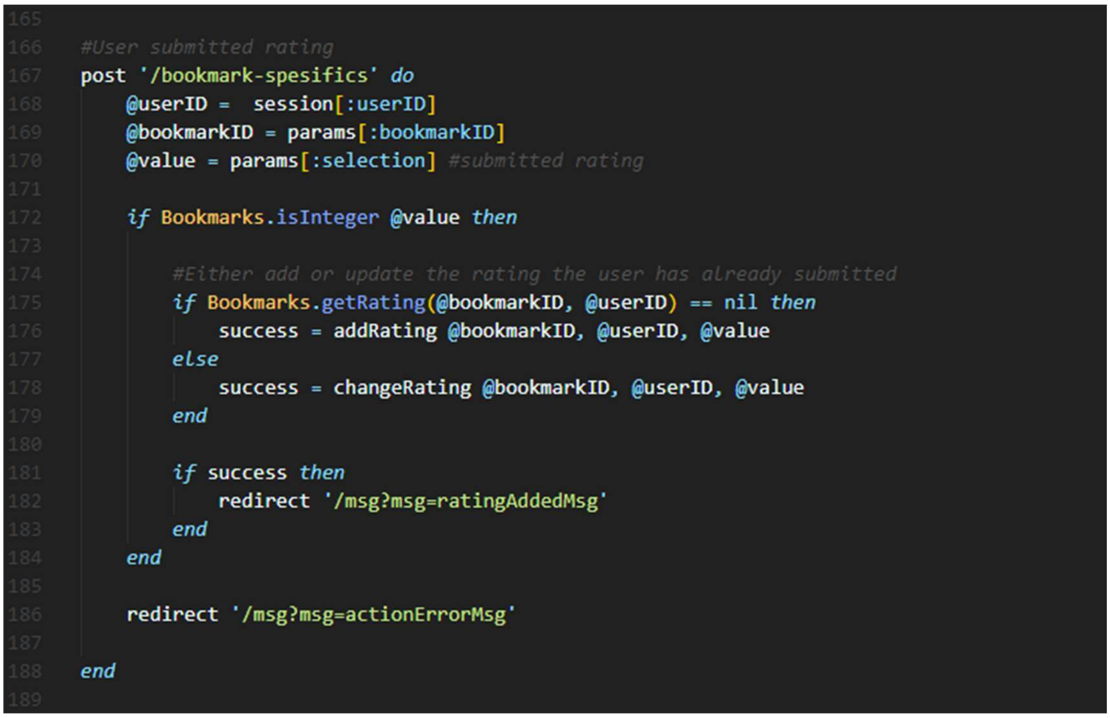
```ruby
#attempted login
post '/authenticate-user' do

    @login = params[:user_email]
    @password = params[:user_password]

    #check credentials
    @result = UserAuthentication.check @login , @password

    if @result == -1
        @error = "Invalid login or password"
    else
        #check if account is active and verified
        if (UserAuthentication.getAccessLevel @result) == 0
            @details = Bookmarks.getAccessDetails @result
            #if not check why
            if  @details[:user_suspended] == 1
                redirect '/msg?msg=suspendedMsg'
            elsif @details[:user_type] == Bookmarks::UNVERIFIED_STRING
                redirect '/msg?msg=unverifiedMsg'
            end
        else
            session[:userID] = @result
            redirect '/'
        end
    end

    erb:loginpage

end
```

Figure 1 Snippet A (bookmark-site.rb)



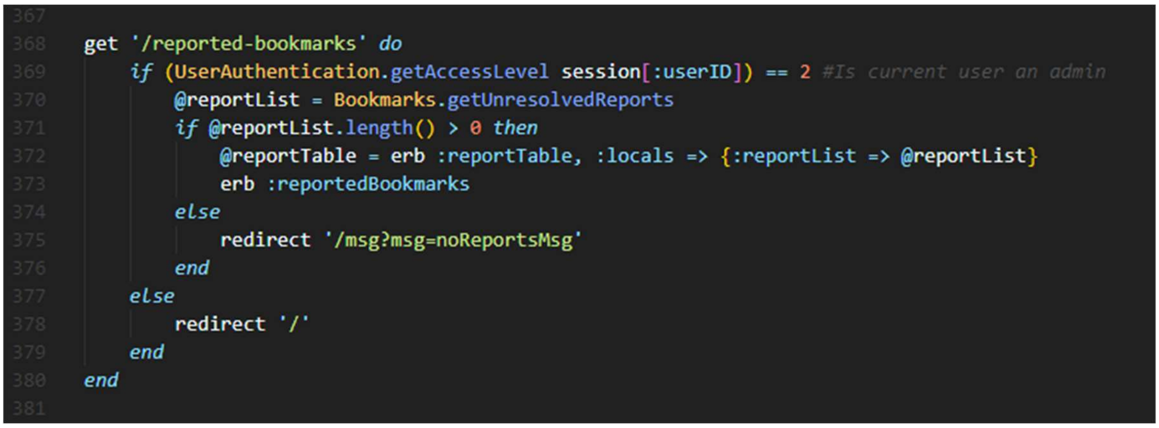
```ruby
# Search handling.
get '/search' do

    @searchQuery = params["search_query"]

    @fullResults = Bookmarks.getHomepageData @searchQuery

    #variables for tagControl
    @tagList = Bookmarks.getTagNames
    #get tags for filter either from submitted form
    #or redirected after clicking on a tag in bookmark details
    @checked = params[:showTag] ? [params[:showTag]] : (extractTagsFromParams params)

    if(@fullResults.length != 0) then
        @results = filterAgainstTags @fullResults, @checked
        erb :search
    else
        erb :noResults
    end

end
```

Figure 2 Snippet B (bookmark-site.rb)



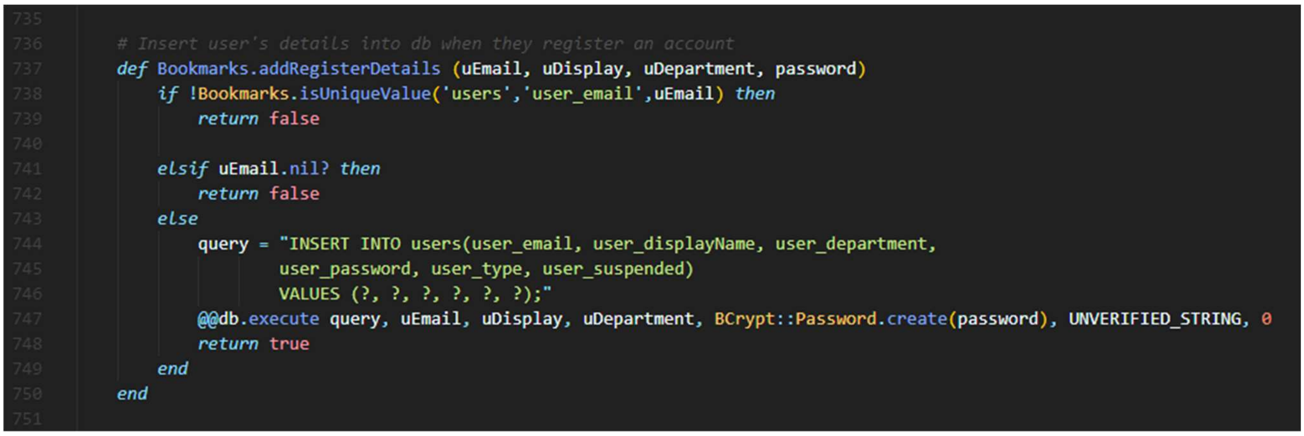
```ruby
#User submitted rating
post '/bookmark-spesifics' do
    @userID =  session[:userID]
    @bookmarkID = params[:bookmarkID]
    @value = params[:selection] #submitted rating

    if Bookmarks.isInteger @value then

        #Either add or update the rating the user has already submitted
        if Bookmarks.getRating(@bookmarkID, @userID) == nil then
            success = addRating @bookmarkID, @userID, @value
        else
            success = changeRating @bookmarkID, @userID, @value
        end

        if success then
            redirect '/msg?msg=ratingAddedMsg'
        end
    end

    redirect '/msg?msg=actionErrorMsg'

end
```

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



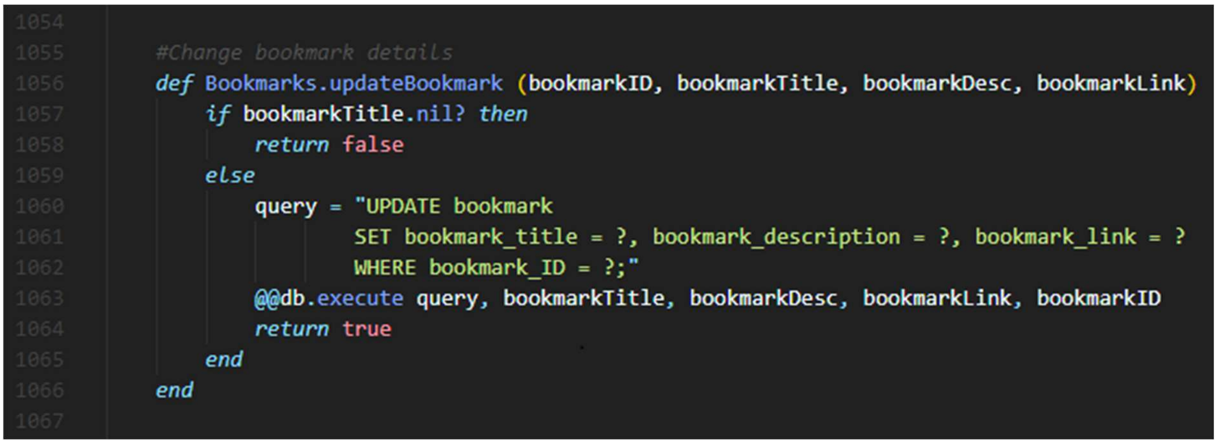
```ruby
get '/reported-bookmarks' do
    if (UserAuthentication.getAccessLevel session[:userID]) == 2 #Is current user an admin
        @reportList = Bookmarks.getUnresolvedReports
        if @reportList.length() > 0 then
            @reportTable = erb :reportTable, :locals => {:reportList => @reportList}
            erb :reportedBookmarks
        else
            redirect '/msg?msg=noReportsMsg'
        end
    else
        redirect '/'
    end
end
```

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



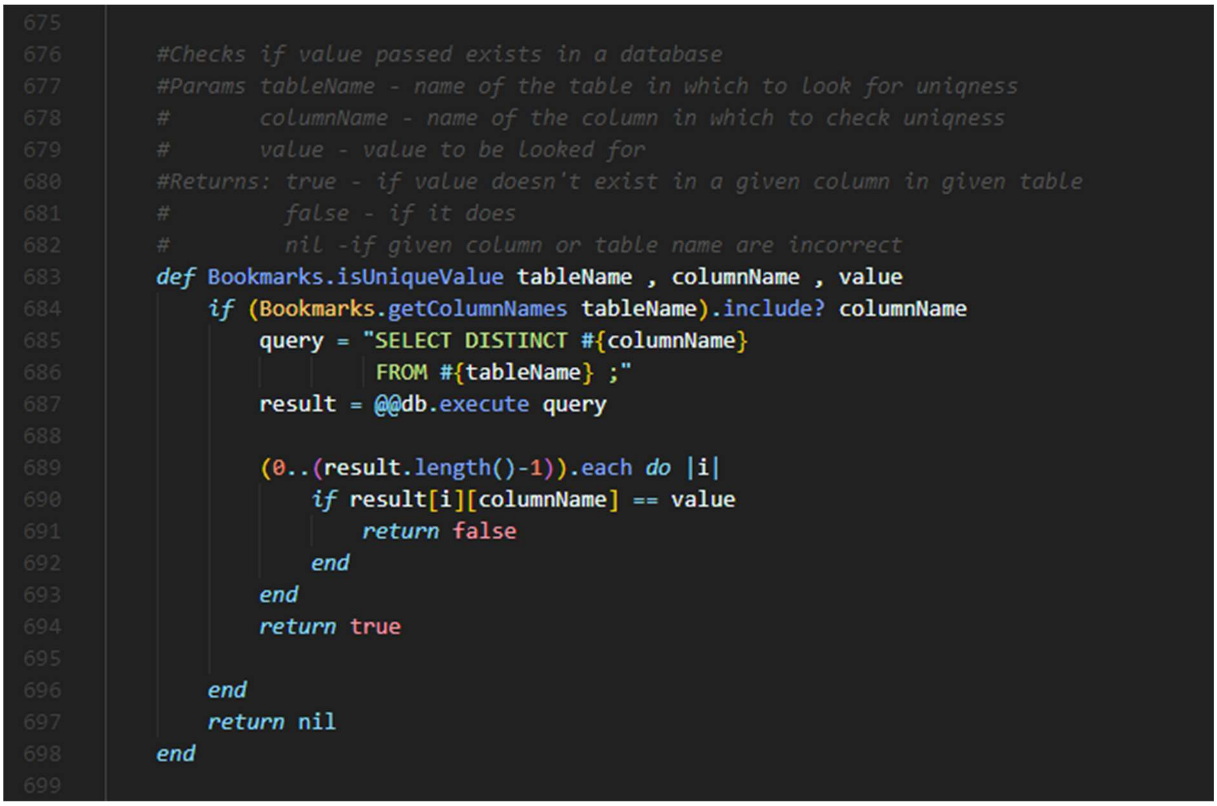
```ruby
# Insert user's details into db when they register an account
def Bookmarks.addRegisterDetails (uEmail, uDisplay, uDepartment, password)
    if !Bookmarks.isUniqueValue('users','user_email',uEmail) then
        return false

    elsif uEmail.nil? then
        return false
    else
        query = "INSERT INTO users(user_email, user_displayName, user_department,
                user_password, user_type, user_suspended)
                VALUES (?, ?, ?, ?, ?, ?);"
        @@db.execute query, uEmail, uDisplay, uDepartment, BCrypt::Password.create(password), UNVERIFIED_STRING, 0
        return true
    end
end
```

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



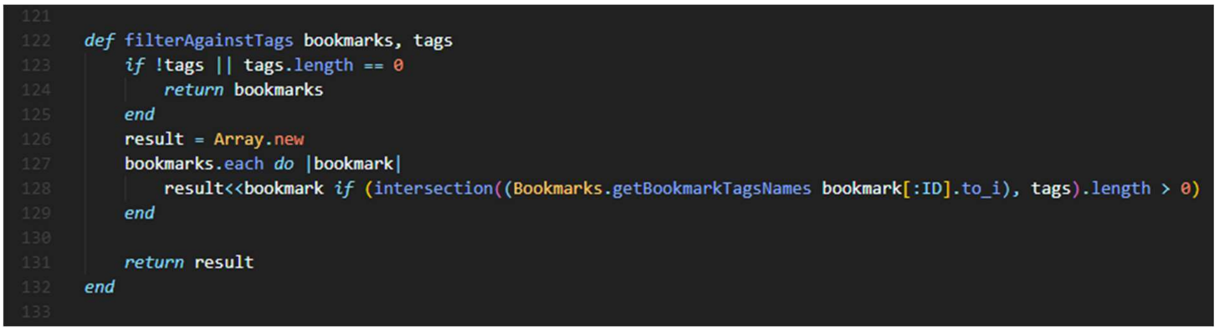
```ruby

    #Change bookmark details
    def Bookmarks.updateBookmark (bookmarkID, bookmarkTitle, bookmarkDesc, bookmarkLink)
        if bookmarkTitle.nil? then
            return false
        else
            query = "UPDATE bookmark
                    SET bookmark_title = ?, bookmark_description = ?, bookmark_link = ?
                    WHERE bookmark_ID = ?;"
            @@db.execute query, bookmarkTitle, bookmarkDesc, bookmarkLink, bookmarkID
            return true
        end
    end

```

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



```ruby

    #Checks if value passed exists in a database
    #Params tableName - name of the table in which to look for uniqness
    #       columnName - name of the column in which to check uniqness
    #       value - value to be looked for
    #Returns: true - if value doesn't exist in a given column in given table
    #         false - if it does
    #         nil -if given column or table name are incorrect
    def Bookmarks.isUniqueValue tableName , columnName , value
        if (Bookmarks.getColumnNames tableName).include? columnName
            query = "SELECT DISTINCT #{columnName}
                    FROM #{tableName} ;"
            result = @@db.execute query

            (0..(result.length()-1)).each do |i|
                if result[i][columnName] == value
                    return false
                end
            end
            return true

        end
        return nil
    end

```

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



```ruby

def filterAgainstTags bookmarks, tags
    if !tags || tags.length == 0
        return bookmarks
    end
    result = Array.new
    bookmarks.each do |bookmark|
        result<<bookmark if (intersection((Bookmarks.getBookmarkTagsNames bookmark[:ID].to_i), tags).length > 0)
    end

    return result
end

```

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