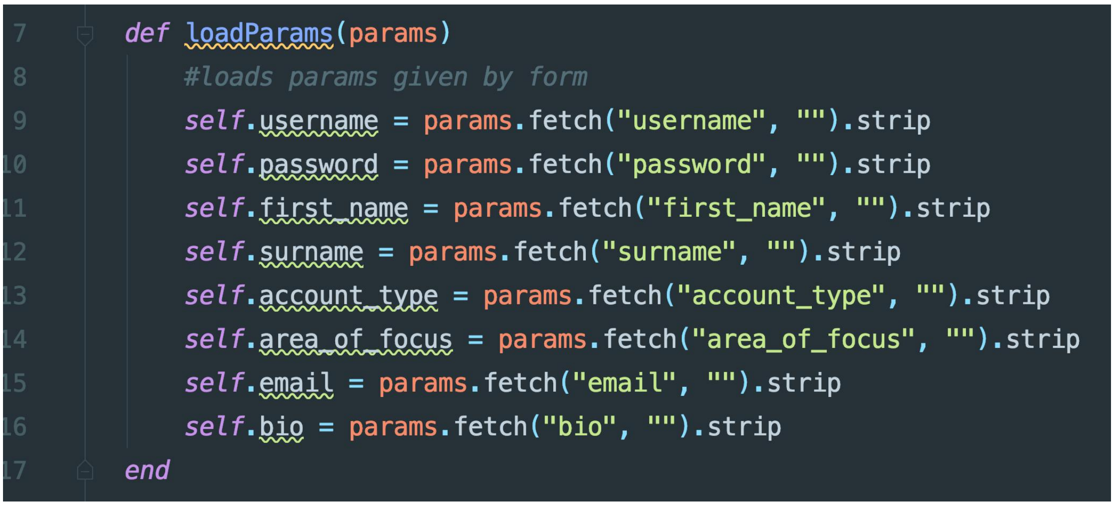
Exemplar Code



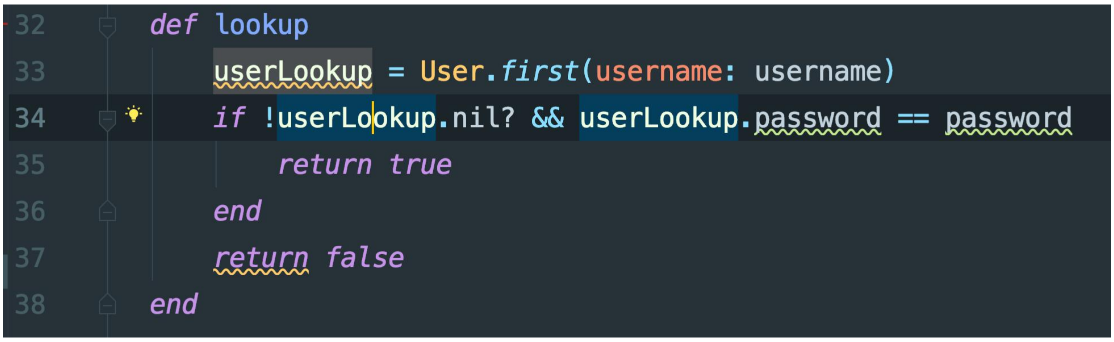
```
7   def loadParams(params)
8       #loads params given by form
9       self.username = params.fetch("username", "").strip
10      self.password = params.fetch("password", "").strip
11      self.first_name = params.fetch("first_name", "").strip
12      self.surname = params.fetch("surname", "").strip
13      self.account_type = params.fetch("account_type", "").strip
14      self.area_of_focus = params.fetch("area_of_focus", "").strip
15      self.email = params.fetch("email", "").strip
16      self.bio = params.fetch("bio", "").strip
17  end
```

a.9 project-master/models/user.rb

This program is mainly to obtain relevant user information from the form

When obtaining user information, the string function strip is used to remove the blank characters at the beginning and the end of the string to prevent the user information from failing due to multiple input of blank characters.

for example：self.username = params.fetch("username", “”).strip

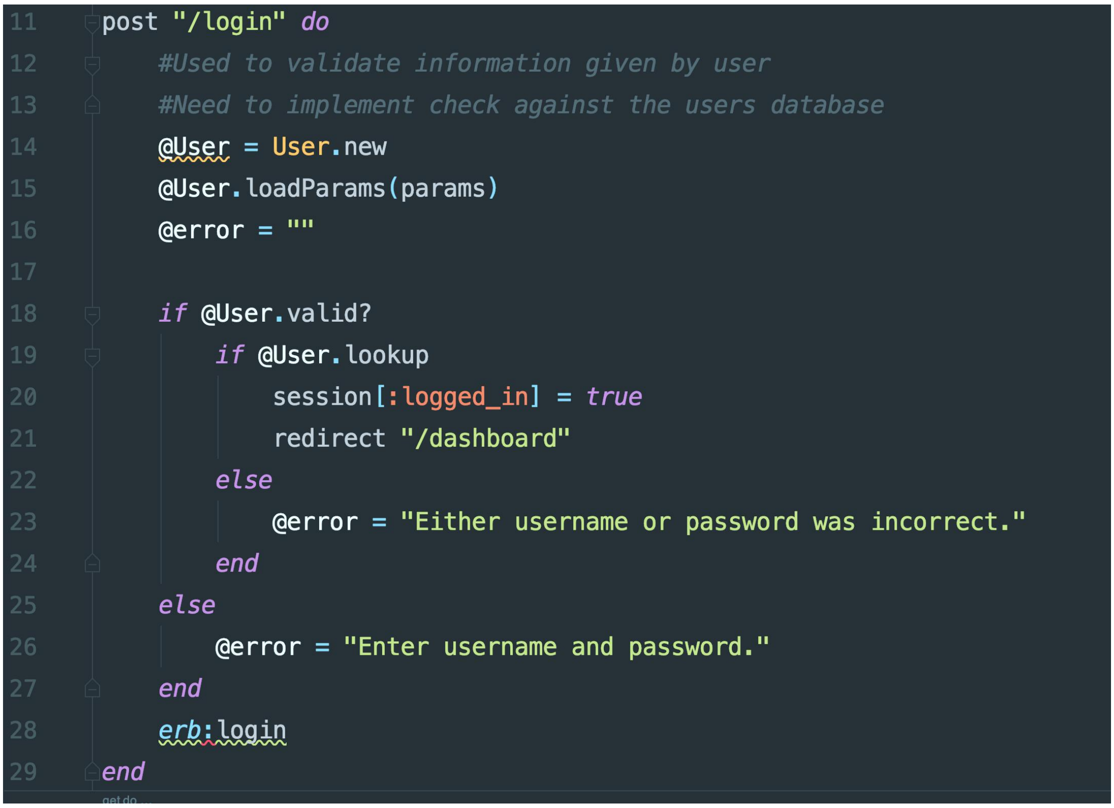


```
32  def lookup
33      userLookup = User.first(username: username)
34      if !userLookup.nil? && userLookup.password == password
35          return true
36      end
37      return false
38  end
```

a.34 project-master/models/user.rb

This program is mainly to check the user login function

if !userLookup.nil? && userLookup.password == password
Here, the function of verifying whether the user exists and whether the user name and password match the user name is written in the same line of code to improve the readability of the code.



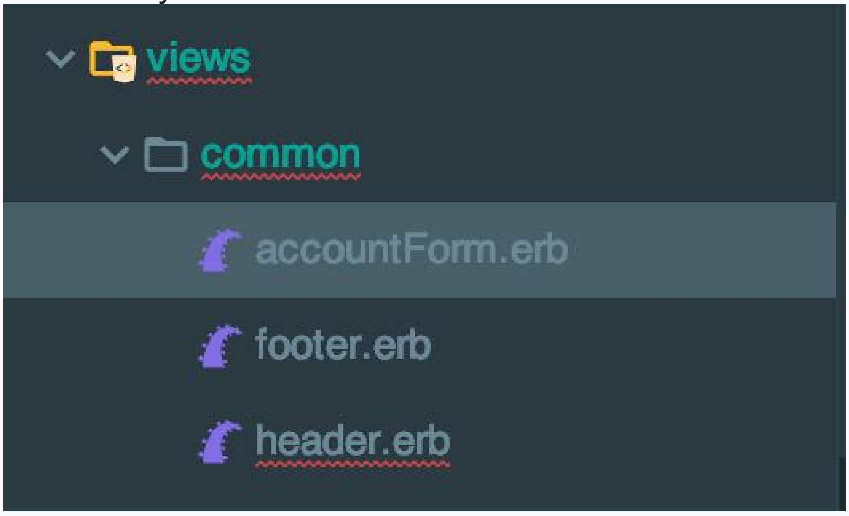
```ruby
post "/login" do
    #Used to validate information given by user
    #Need to implement check against the users database
    @User = User.new
    @User.loadParams(params)
    @error = ""

    if @User.valid?
        if @User.lookup
            session[:logged_in] = true
            redirect "/dashboard"
        else
            @error = "Either username or password was incorrect."
        end
    else
        @error = "Enter username and password."
    end
    erb:login
end
```

a.15 project-master/controllers/login.rb

This program is mainly to realize the user login function.

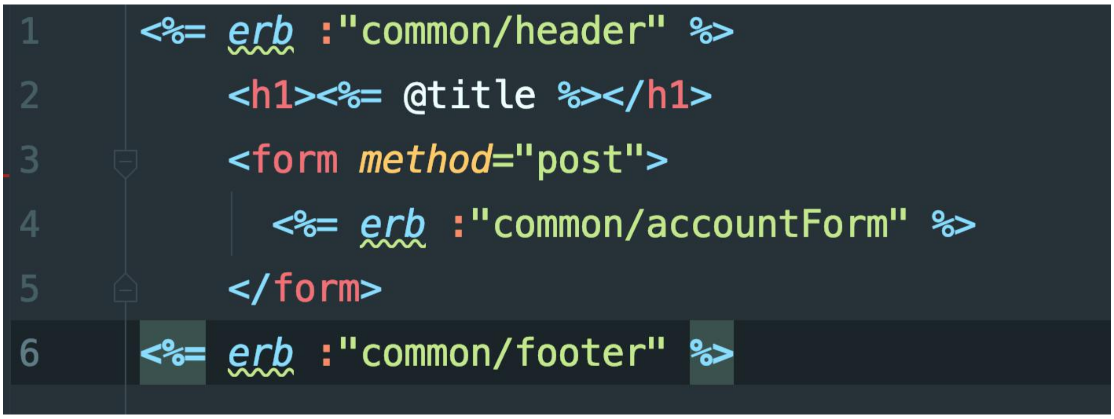
The loadParams method can be trimmed from the code to be implemented in the user model.

The advantage of this is that on the one hand, it is easy to call directly, making the login program look less cumbersome, and on the other hand, it makes the code easy to read and understand.



Judging from the file directory, this code is mainly to extract the common code. The purpose of this is to prevent repetitive code from being written, so that the code can be reused as much as possible to facilitate future modifications.

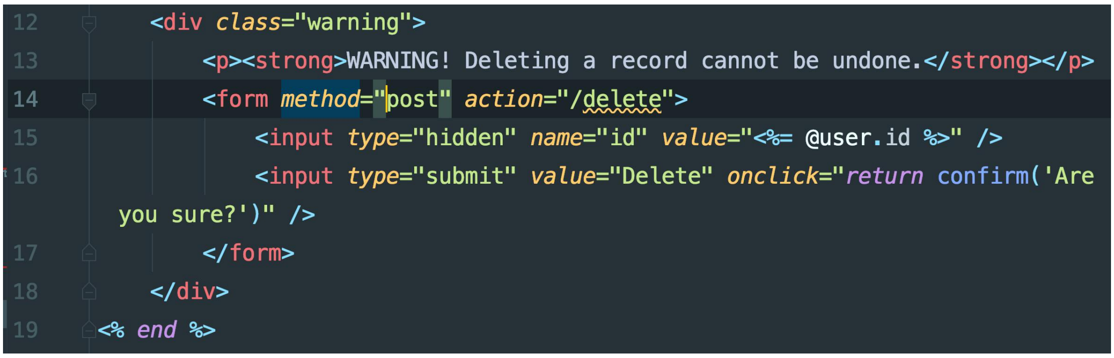
The create.rb in the illustration below uses these common codes to make the program look more concise.



```erb
<%= erb :"common/header" %>
    <h1><%= @title %></h1>
    <form method="post">
      <%= erb :"common/accountForm" %>
    </form>
<%= erb :"common/footer" %>
```

a.1.4.6 project-master/views/create.erb

Friendly error notifications can improve the user experience.



```erb
    <div class="warning">
        <p><strong>WARNING! Deleting a record cannot be undone.</strong></p>
        <form method="post" action="/delete">
            <input type="hidden" name="id" value="<%= @user.id %>" />
            <input type="submit" value="Delete" onclick="return confirm('Are you sure?')" />
        </form>
    </div>
<% end %>
```

a.16 project-master/views/edit.erb

Here onclick can prevent accidental deletion, let the user double-check whether really confirm delete, so as not to cause data loss.