

Exemplar Code

Listing A (views/login.erb):

```
22 <!--defines form that posts to login.rb controller -->
23 <form method="post" action="login">
24   <label>Email:</label>
25   <input type="email" name="email" value="<%= h @user.email unless @user.nil? %>"/>
26
27   <!-- uses methods from the User model to print errors with email (eg: missing @)-->
28   <% if !@user.nil? && @user.errors.include?("email") %>
29     <% @user.errors["email"].each do |error| %>
30       <p class="error"><strong><%= error %></strong></p>
31     <% end %>
32   <% end %>
33   <br>
34   <label>Password:</label>
35   <input type="password" name="password" value="<%= h @user.password unless @user.nil? %>"/>
36
37   <!-- uses User model to print errors with password (eg: too short)-->
38   <% if !@user.nil? && @user.errors.include?("password") %>
39     <% @user.errors["password"].each do |error| %>
40       <p class="error"><strong><%= error %></strong></p>
41     <% end %>
42   <% end %>
```

Listing B (controllers/request.rb):

```
1  get "/request" do
2
3    mentee_id = session[:user_id]
4    puts "This is mentee's ID: #{session[:user_id]}"
5
6    if(session[:user_type].eq?("mentee"))
7      puts "User is a mentee."
8      @pair = Pair.new
9
10     puts "This is mentor's ID: #{params["id"]}"
11     @pair.mentor_id = params["id"]
12     @pair.mentee_id = mentee_id
13     @pair.status = 0 #every request is initially not accepted
14     #updating DB with new request
15     if @pair.valid?
16       puts "We have valid request."
17       @pair.save_changes
18       puts "Changes saved. Redirecting..."
19       response.set_cookie("request-popup", value: 'true')
20
21     else
22       puts "Request is invalid."
23       puts "#{@pair.errors}"
24     end
25   end
26   redirect "/browsementors"
27 end
```

Listing C (spec/feature/acceptance_spec.rb):

```
23 it "will not add a user without matching passwords" do
24   visit "/register"
25   fill_in "email", with: "test@email.com"
26   fill_in "password", with: "test1234"
27   fill_in "password_repeat", with: "test12345"
28   click_button "submit_register"
29   expect(page).to have_content "not the same"
30 end
31
32 it "will not add a user with short password (less than 8 characters)" do
33   visit "/register"
34   fill_in "email", with: "test@email.com"
35   fill_in "password", with: "test1"
36   fill_in "password_repeat", with: "test1"
37   click_button "submit_register"
38   expect(page).to have_content "must be at least"
39 end
40
41 it "will add a user when all details are entered" do
42   visit "/register"
43   add_user("mentee")
44   expect(page).to have_content "You have registered successfully"
45   clear_database
46 end
```

Uses capybara to simulate a user, and fill in forms