

The Exemplar code:

```
1 # frozen_string_literal: true
2
3 require "rspec"
4 require "../db/dbqueries"
5
6 RSpec.describe User do
7   describe "#self.exist?" do
8     context "when given a name test that exist" do
9       it { expect(described_class.exist?("test")).to eq(true) }
10    end
11
12     context "when given a name test3 that doesnt exist" do
13       it { expect(described_class.exist?("test3")).to eq(false) }
14    end
15  end
16
17   describe "#self.login?" do
18     context "when given correct login details" do
19       it "returns true" do
20         expect(described_class.login?("test2", "test")).to eq(true)
21       end
22     end
23
24     context "when given incorrect login details" do
25       it "returns false" do
26         expect(described_class.login?("Tony", "Kc5Wd4b6")).to eq(false)
27       end
28     end
29   end
30 end
```

Listing A project/spec/dbqueries_spec.rb

This part shows how we handled testing, the User.login?, which was previously tested in order to ensure the user was properly added in the database.

```

3 require "sinatra"
4 require "rspec"
5
6 get "/form" do
7   @user_name = params["Username"]
8   @password = params["Password"]
9
10  @form_was_submitted = !@user_name.nil? || !@password.nil?
11
12  @user_name_error = nil
13  @password_error = nil
14  @correct_details = false
15
16  if @form_was_submitted
17    @user_name.strip!
18    @password.strip!
19    # If an user with this username exists
20    if User.exist?(@user_name)
21      # If the login details are correct
22      User.login?(@user_name, @password) ? @correct_details = true : @password_error = false
23    else
24      @user_name_error = true
25    end
26  else
27    # If user's inputs are empty, assign true to error for the empty field
28    @user_name.nil? ? @user_name_error = true : @password_error = true
29  end
30 end

```

Listing B project/controllers/form.rb

This section is designed for testing the user's input. Checking the login information and feedback if the input are incorrect.