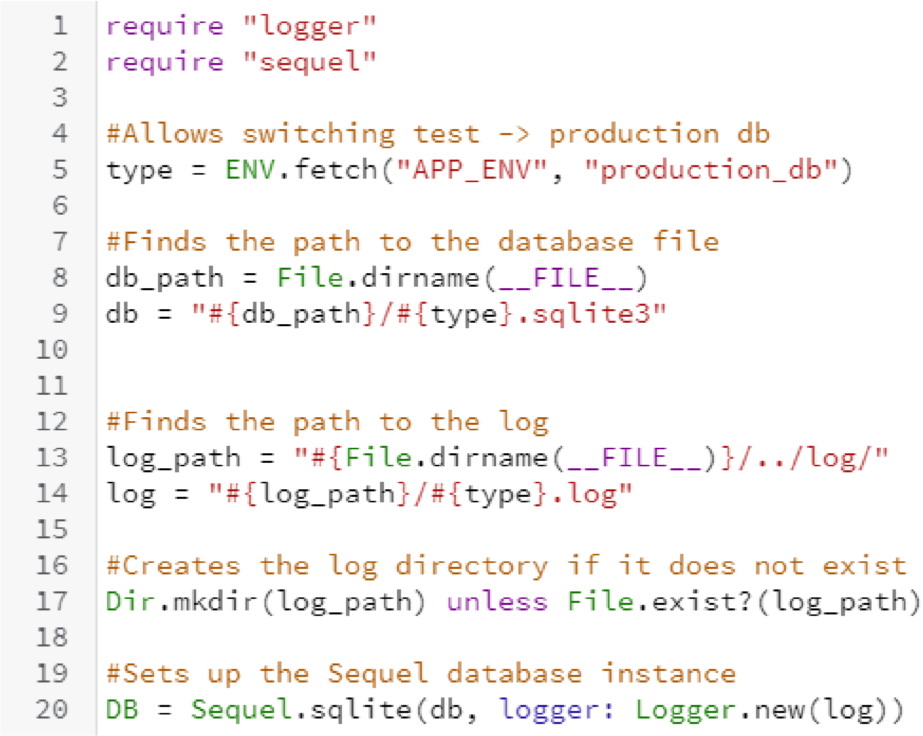
Team 33 Exemplar Code – Iteration 1

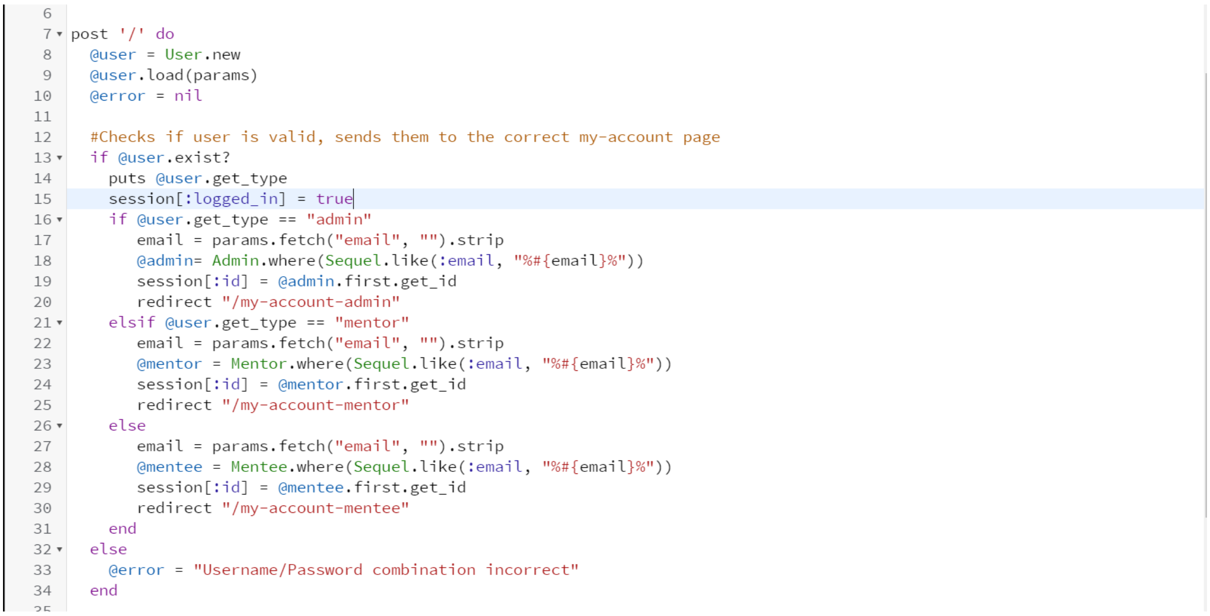
Databases



```ruby
require "logger"
require "sequel"

#Allows switching test -> production db
type = ENV.fetch("APP_ENV", "production_db")

#Finds the path to the database file
db_path = File.dirname(__FILE__)
db = "#{db_path}/#{type}.sqlite3"


#Finds the path to the log
log_path = "#{File.dirname(__FILE__)}/../log/"
log = "#{log_path}/#{type}.log"

#Creates the log directory if it does not exist
Dir.mkdir(log_path) unless File.exist?(log_path)

#Sets up the Sequel database instance
DB = Sequel.sqlite(db, logger: Logger.new(log))
```

Listing A project/db/db.rb

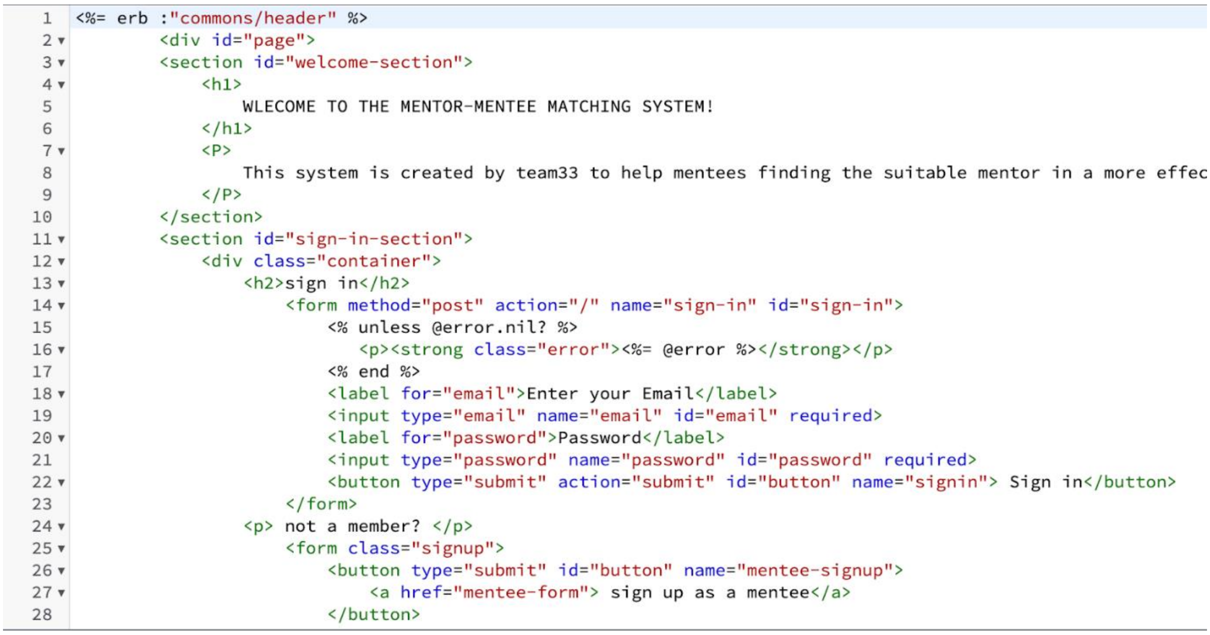
Controllers



```ruby

post '/' do
  @user = User.new
  @user.load(params)
  @error = nil

  #Checks if user is valid, sends them to the correct my-account page
  if @user.exist?
    puts @user.get_type
    session[:logged_in] = true
    if @user.get_type == "admin"
       email = params.fetch("email", "").strip
       @admin= Admin.where(Sequel.like(:email, "%#{email}%"))
       session[:id] = @admin.first.get_id
       redirect "/my-account-admin"
    elsif @user.get_type == "mentor"
       email = params.fetch("email", "").strip
       @mentor = Mentor.where(Sequel.like(:email, "%#{email}%"))
       session[:id] = @mentor.first.get_id
       redirect "/my-account-mentor"
    else
       email = params.fetch("email", "").strip
       @mentee = Mentee.where(Sequel.like(:email, "%#{email}%"))
       session[:id] = @mentee.first.get_id
       redirect "/my-account-mentee"
    end
  else
    @error = "Username/Password combination incorrect"
  end
```

Listing B project/controllers/index.rb

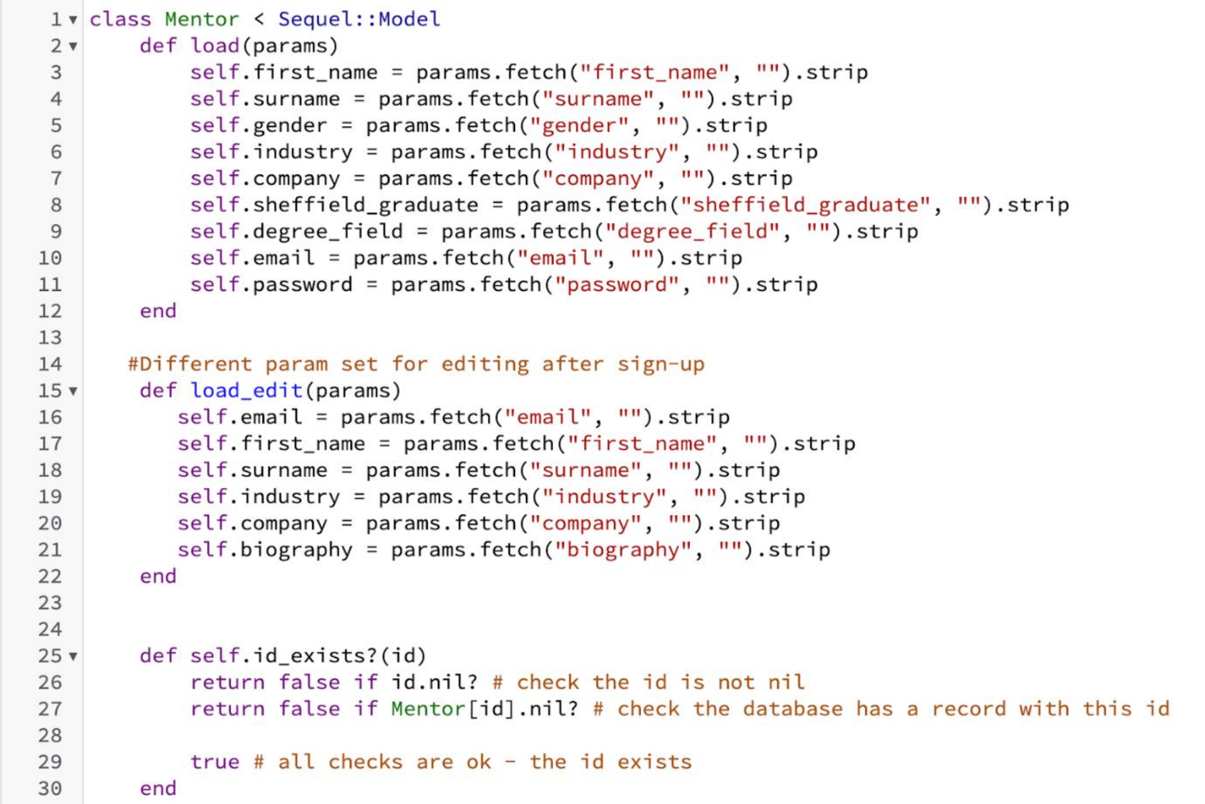
Views



```
<%= erb :"commons/header" %>
        <div id="page">
        <section id="welcome-section">
            <h1>
                WLECOME TO THE MENTOR-MENTEE MATCHING SYSTEM!
            </h1>
            <P>
                This system is created by team33 to help mentees finding the suitable mentor in a more effec
            </P>
        </section>
        <section id="sign-in-section">
            <div class="container">
                <h2>sign in</h2>
                    <form method="post" action="/" name="sign-in" id="sign-in">
                        <% unless @error.nil? %>
                           <p><strong class="error"><%= @error %></strong></p>
                        <% end %>
                        <label for="email">Enter your Email</label>
                        <input type="email" name="email" id="email" required>
                        <label for="password">Password</label>
                        <input type="password" name="password" id="password" required>
                        <button type="submit" action="submit" id="button" name="signin"> Sign in</button>
                    </form>
                <p> not a member? </p>
                    <form class="signup">
                        <button type="submit" id="button" name="mentee-signup">
                            <a href="mentee-form"> sign up as a mentee</a>
                        </button>
```

Listing C project/views/index.erb

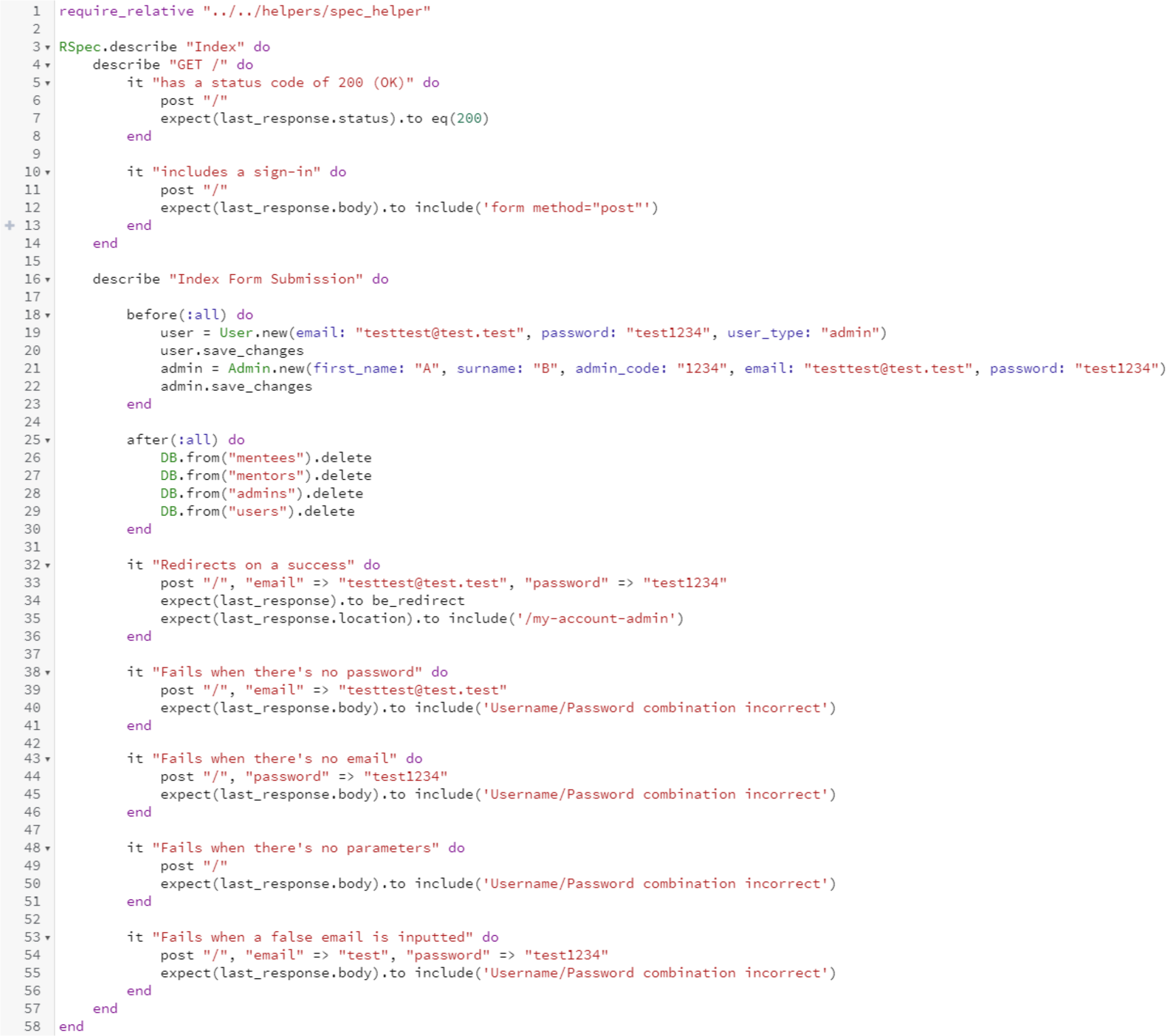
Models



```
class Mentor < Sequel::Model
    def load(params)
        self.first_name = params.fetch("first_name", "").strip
        self.surname = params.fetch("surname", "").strip
        self.gender = params.fetch("gender", "").strip
        self.industry = params.fetch("industry", "").strip
        self.company = params.fetch("company", "").strip
        self.sheffield_graduate = params.fetch("sheffield_graduate", "").strip
        self.degree_field = params.fetch("degree_field", "").strip
        self.email = params.fetch("email", "").strip
        self.password = params.fetch("password", "").strip
    end

   #Different param set for editing after sign-up
    def load_edit(params)
       self.email = params.fetch("email", "").strip
       self.first_name = params.fetch("first_name", "").strip
       self.surname = params.fetch("surname", "").strip
       self.industry = params.fetch("industry", "").strip
       self.company = params.fetch("company", "").strip
       self.biography = params.fetch("biography", "").strip
    end


    def self.id_exists?(id)
        return false if id.nil? # check the id is not nil
        return false if Mentor[id].nil? # check the database has a record with this id

        true # all checks are ok - the id exists
    end
```

Listing D project/models/mentor.rb

Testing Unit



```ruby
require_relative "../../helpers/spec_helper"

RSpec.describe "Index" do
    describe "GET /" do
        it "has a status code of 200 (OK)" do
            post "/"
            expect(last_response.status).to eq(200)
        end

        it "includes a sign-in" do
            post "/"
            expect(last_response.body).to include('form method="post"')
        end
    end

    describe "Index Form Submission" do

        before(:all) do
            user = User.new(email: "testtest@test.test", password: "test1234", user_type: "admin")
            user.save_changes
            admin = Admin.new(first_name: "A", surname: "B", admin_code: "1234", email: "testtest@test.test", password: "test1234")
            admin.save_changes
        end

        after(:all) do
            DB.from("mentees").delete
            DB.from("mentors").delete
            DB.from("admins").delete
            DB.from("users").delete
        end

        it "Redirects on a success" do
            post "/", "email" => "testtest@test.test", "password" => "test1234"
            expect(last_response).to be_redirect
            expect(last_response.location).to include('/my-account-admin')
        end

        it "Fails when there's no password" do
            post "/", "email" => "testtest@test.test"
            expect(last_response.body).to include('Username/Password combination incorrect')
        end

        it "Fails when there's no email" do
            post "/", "password" => "test1234"
            expect(last_response.body).to include('Username/Password combination incorrect')
        end

        it "Fails when there's no parameters" do
            post "/"
            expect(last_response.body).to include('Username/Password combination incorrect')
        end

        it "Fails when a false email is inputted" do
            post "/", "email" => "test", "password" => "test1234"
            expect(last_response.body).to include('Username/Password combination incorrect')
        end
    end
end
```

Listing E project/spec/unit/index_spec.rb