

3EF-DI

Szymon Starzak

Jacek Cieślak

POLITECHNIKA RZESZOWSKA
im. Ignacego Łukasiewicza

Wydział Elektrotechniki i Informatyki

Nowoczesne technologie programistyczne



Sprawozdanie z projektu: „Aplikacja rezerwacji zajęć”

1. Opis projektu

- Celem projektu było stworzenie aplikacji webowej w Ruby on Rails, w której pracownicy mogą rezerwować zajęcia, a studenci mogą się do nich zapisywać.

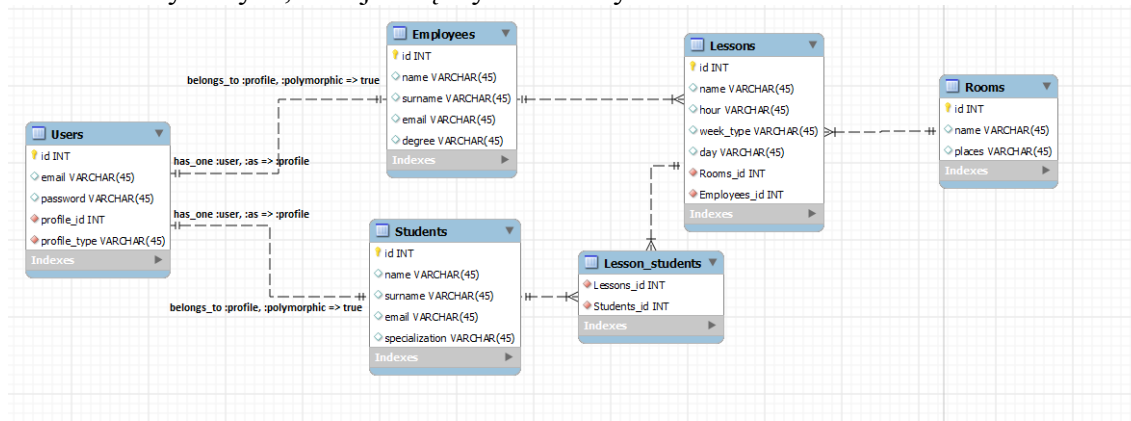
2. Rejestracja i logowanie

- Autentykacja użytkowników została stworzona za pomocą Devise. Zarówno przy rejestracji studentów jak i pracowników dane logowania (email i hasło) są przesyłane do jednej tabeli, a pozostałe informacje w zależności od tego czy rejestrująca się osoba jest studentem czy pracownikiem, trafiają do swoich tabel. Taka realizacja możliwa była dzięki wykorzystaniu mechanizmu polimorficznych asocjacji. Tak więc studenci i pracownicy posiadają 2 różne formularze rejestracji, ale dane potrzebne do logowania będą przechowywane w jednym modelu.
- Do logowania wykorzystujemy jeden formularz w którym podajemy swój email i hasło.

3. System kontroli wersji

- Przy tworzeniu wykorzystany został system kontroli wersji Git, dzięki któremu można było commitować postęp w tworzeniu projektu do zdalnego repozytorium.

4. Schemat bazy danych, relacje między stworzonymi modelami.



5. Wykorzystane Gemy

- Oprócz standardowych gemów dodawanych przez generowanie nowego projektu dodaliśmy takie gemy jak:
 - gem 'devise'- do autentykacji użytkowników
 - gem 'bootstrap-sass'- framework CSS, to tworzenia interfejsu graficznego stron
 - gem 'jquery-rails' -framework JS do upiększania stron
 - gem 'tzinfo-data', platforms: [:mingw, :mswin] – konieczne w przypadku korzystania z platformy windows.

6. Zrzuty ekranu

- Kody źródłowe do obsługi pracowników.

```
class Employee < ActiveRecord::Base
  attr_accessor :email, :password

  has_one :user, :as => :profile
  has_many :lessons

  validates :name, :surname, :degree, :email, :password, presence: true
  validates :name, :surname, length: { minimum: 3 }, :presence => { :message => " cannot be blank" }
end
```

1. Ilustracja: Model dla pracownika

```
class EmployeesController < ApplicationController
  def new
    @employee = Employee.new
  end

  def create
    @employee = Employee.new(employee_params)
    @first_save = @employee.save

    @employee.create_user
    @employee.user.email = @employee.email
    @employee.user.password = @employee.password
    @second_save = @employee.user.save

    if @first_save && @second_save
      redirect_to user_session_path
    else
      render 'new'
    end
  end

  def index
  end

  def employee_params
    params.require(:employee).permit(:name, :surname, :degree, :email, :password)
  end
end
```

2. Ilustracja: Kontroler Pracownika

```
<%= render 'employees/error_messages' %>
<div class="container">
  <div class="login-container">
    <div id="output" ></div>
    <div class="form-box">
      <h3> Rejestracja Pracownika </h3>
      <%= form_for :employee, url: employees_path, html: { method: :post, :class => "form-horizontal" } do |f| %>
        <%= f.text_field :name, placeholder: "Imię", autofocus: true %>
        <%= f.text_field :surname, placeholder: "Nazwisko" %>
        <%= f.text_field :degree, placeholder: "Stopień naukowy" %>
        <%= f.text_field :email, placeholder: "Email" %>
        <%= f.password_field :password, placeholder: "Hasło" %>
      <hr>
      <div class="form-group text-center">
        <%= f.submit "Zapisz", class: "btn btn-primary active btn-lg" %>
      </div>
    <% end %>
  </div>
</div>
<div class="col-xs-2 col-sm-4" >
</div>
</div>
```

3. Ilustracja: Widok dla akcji new kontrolera Pracownika

- Wygląd aplikacji w przeglądarce.

4. Ilustracja: Rejestracja pracownika

| Id | Nazwa | Godzina | Dzień | Tydzień | Sala | Miejsce | Edycja | Delete |
|----|-------|---------|--------|---------|------|---------|--------|--------|
| 2 | Java | 16:00 | Środa | AB | A1 | 15/15 | Edycja | Delete |
| 3 | Java | 12:00 | Piątek | AB | F102 | 15/15 | Edycja | Delete |
| 4 | Java | 8:00 | Wtorek | A | B101 | 15/15 | Edycja | Delete |

5. Ilustracja: Dodawanie zajęć przez pracownika

| Id | Nazwa | Godzina | Dzień | Tydzień | Sala | Miejsce | Zapisz |
|----|----------|---------|----------|---------|------|---------|-----------------|
| 1 | Historia | 8:00 | Czwartek | B | B101 | 14/15 | Jesteś zapisany |
| 2 | Java | 16:00 | Środa | AB | A1 | 15/15 | Zapisz |
| 3 | Java | 12:00 | Piątek | AB | F102 | 14/15 | Jesteś zapisany |
| 4 | Java | 8:00 | Wtorek | A | B101 | 15/15 | Zapisz |

6. Ilustracja: Rezerwacja zajęć przez studenta

7. Wnioski:

- Ruby on rails jest bardzo przyjaznym narzędziem do tworzenia aplikacji webowych, dzięki jego głównym założeniom:
 - DRY
 - Convention over Configuration
 - szybkość, łatwość i przyjemność pisania kodu,

- możliwość użycia wtyczek, które w sposób błyskawiczny rozszerzają aplikacje o rozmaite funkcje, jak logowanie, przesyłanie i skalowanie obrazków, czy tagowanie.
- Aplikacje pisane w RoR są o wiele bezpieczniejsze ponieważ sami musimy definiować przekierowania co uniemożliwia dostęp do wrażliwych części skryptu.
- W sieci internet można znaleźć mnóstwo przykładów i pomocnych tematów pomagających praktycznie w rozwiązaniu każdego problemu. Jednak wymaga to dobrej znajomości języka angielskiego.