**Міністерство освіти і науки України**

**Івано-Франківський національний технічний університет нафти і газу**

Кафедра програмного забезпечення автоматизованих систем

**ЛАБОРАТОРНА РОБОТА №3**

Тема:«Розроблення адміністративної частини сайту**»**

Виконав студент

групи ПІт-15-3 **Зьола І. П.**

Перевірив:

**Броновський І.В.**

Івано-Франківськ

2016

**Мета:** Розробити адміністративної частини сайту.

**Хід роботи**

Для розробки backend було обрано мову програмування Ruby та фреймворк Ruby on Rails.

Ruby — це інтерпретована, повністю об'єктно-орієнтована мова програмування з чіткою динамічною типізацією. Мова відрізняється високою ефективністю розробки програм і увібрала в себе найкращі риси Perl, Java, Python, Smalltalk, Eiffel, Ada і Lisp. Ruby поєднує в собі Perl-подібний синтаксис із об'єктно-орієнтованим підходом мови програмування Smalltalk. Також деякі риси запозичено із мов програмування Python, Lisp, Dylan та CLU.

Ruby on Rails — об'єктно-орієнтований фреймворк для створення веб-додатків, написаний на мові програмування Ruby. Ruby on Rails надає каркас модель-вид-контролер (Model-View-Controller) для веб-додатків, а також забезпечує їхню інтеграцію з веб-сервером і сервером бази даних.

Ruby on Rails був створений Девідом Гайнемаєр Генссоном на основі його роботи над засобом керування проектами Basecamp і був випущений в липні 2004 року. Ruby on Rails є відкритим програмним забезпеченням і розповсюджується за ліцензією MIT.

Для інсталяції Ruby та RoR на Ubuntu-подібних системах використовується rvm(Ruby Version Manager). Щоб встановити rvm та Ruby+RoR необхідно:

1) Встановити Curl

sudo apt-get install curl

2) Завантажити rvm та встановити Node.js, Ruby+RoR

gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3

\curl -L https://get.rvm.io | bash -s stable --ruby

rvm get stable --autolibs=enable

rvm install ruby-2.3.1

rvm --default use ruby-2.3.1

sudo apt-get install nodejs

gem install bundler

gem install rails

Бібліотеки в Ruby називаються gem-ами. Список всіх gem-ів, які використовуються в проекті знаходиться в корені проекту в файлі Gemfile. Щоб встановити всі gem-и, які використовуються в проекті неохідно в терміналі перейти в корінь проекту та виконати команду:

bundle install

Для виконання цієї лабораторної роботи було використано такі gem-и: devise (для автентифікації), pg (для роботи з PostgreSQL), jquery-rails (для підкючення JQuery).

**Код класу BooksСontroller:**

class BooksController < ApplicationController

before\_action :set\_book, only: [:show, :edit, :update, :destroy]

def index

@books = Book.includes(:genres)

.search(params[:keyword]).filter(params[:filter]).approved

@genres = Genre.all

end

def show

end

def create

@book = Book.new(book\_params)

respond\_to do |format|

if @book.save

format.html { redirect\_to @book, notice: 'Book was successfully created.' }

format.js

else

format.html { render :new }

end

end

end

private

def set\_book

@book = Book.find(params[:id])

end

def book\_params

params.require(:book).permit(:title, :author, :description, :amazon\_id,

{ genre\_ids: [] }, :rating)

end

end

**Код класу Book:**

class Book < ApplicationRecord

has\_many :book\_genres

has\_many :genres, through: :book\_genres

before\_create :set\_approved

scope :approved, -> {where(approved: true)}

scope :search, ->(keyword) { where('keywords LIKE ?', "%#{keyword.downcase}%") if keyword.present? }

scope :filter, ->(name) {

joins(:genres).where('genres.name = ?', name) if name.present?

}

validates\_inclusion\_of :rating, in: 1..5

validates :rating, numericality: { only\_integer: true }

protected

def set\_keywords

self.keywords = [title, author, description].map(&:downcase).join(' ')

end

def set\_approved

self.approved ||= false

end

end

**Код класу Admin/BooksСontroller:**

class Admin::BooksController < AdminController

before\_action :set\_book, only: [:edit, :update, :destroy, :approve]

before\_action :check\_admin

def index

@books = Book.all

end

def new

@book = Book.new

@genres = Genre.all

end

def edit

@genres = Genre.all

end

def create

@book = Book.new(book\_params)

if @book.save

redirect\_to admin\_books\_url, notice: 'Book was successfully created.'

else

render :new

end

end

def update

if @book.update(book\_params)

redirect\_to admin\_books\_url, notice: 'Book was successfully updated.'

else

render :edit

end

end

def destroy

@book.destroy

redirect\_to admin\_books\_url, notice: 'Book was successfully destroyed.'

end

def approve

@book.update(approved: true)

redirect\_to admin\_books\_url, notice: 'Book was successfully approved.'

end

private

def set\_book

@book = Book.find(params[:id])

end

def book\_params

params.require(:book).permit(:title, :author, :description, :amazon\_id,

:approved, { genre\_ids: [] }, :rating)

end

end

**Файл routes.rb:**

Rails.application.routes.draw do

devise\_for :users

root to: 'books#index'

resources :genres, only: [:index]

resources :books, only: [:index, :create, :show]

namespace :admin do

resources :genres, except:[:show]

resources :books, except:[:show]

post 'books/:id/approve', to: 'books#approve', as: 'approve\_book'

end

get 'static\_pages/api', as: 'api'

end

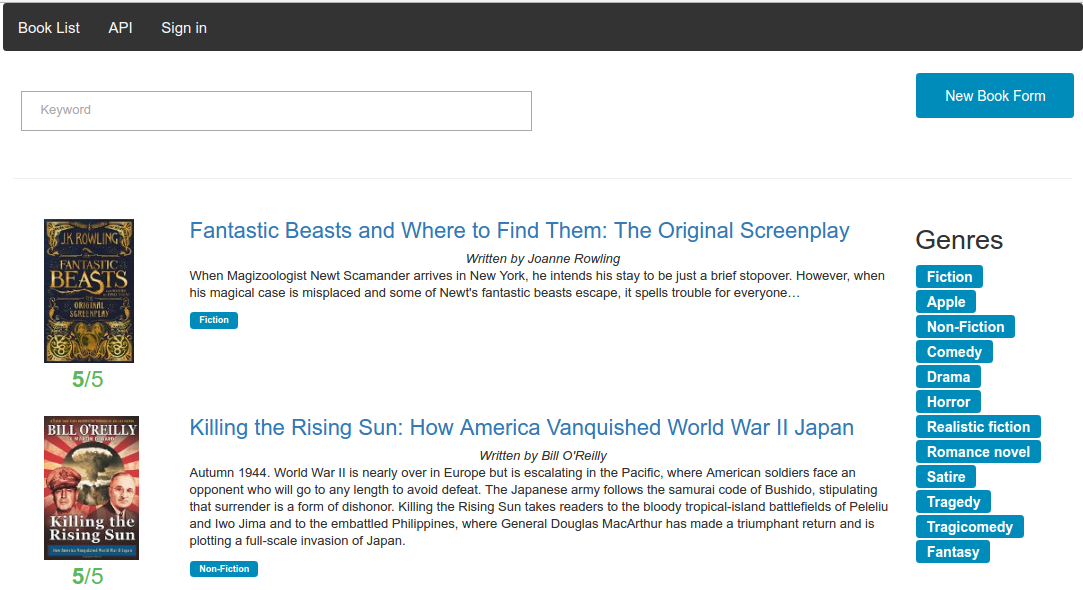


Рис. 1 - Вигляд основної частини сайту.

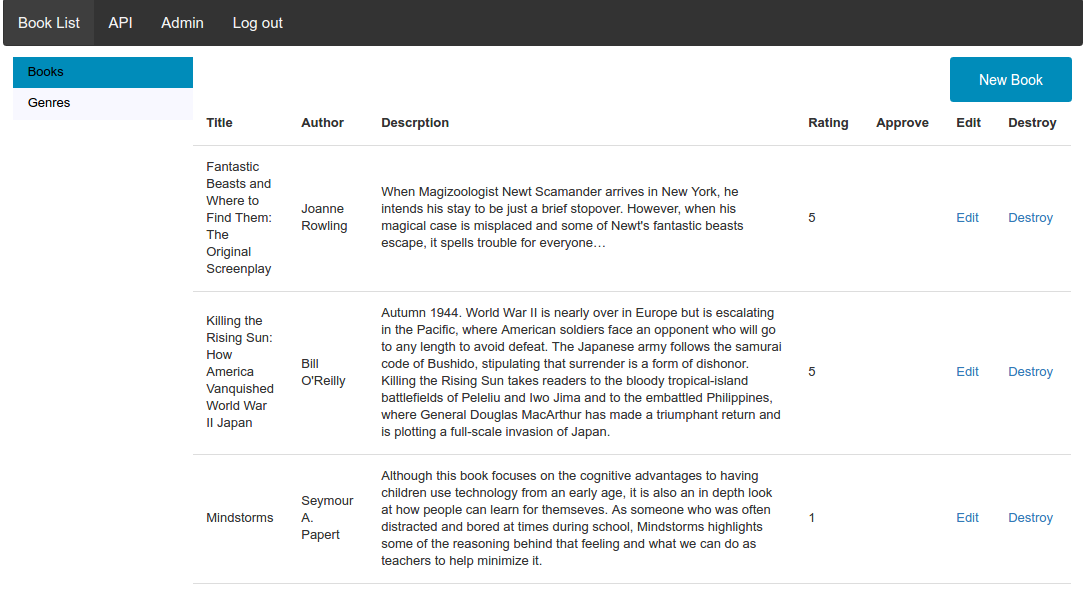


Рис. 2 - Вигляд адмін панелі.

**Висновок**: На даній лабораторній роботі було розроблено адміністративну частину сайту.