**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**

**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]

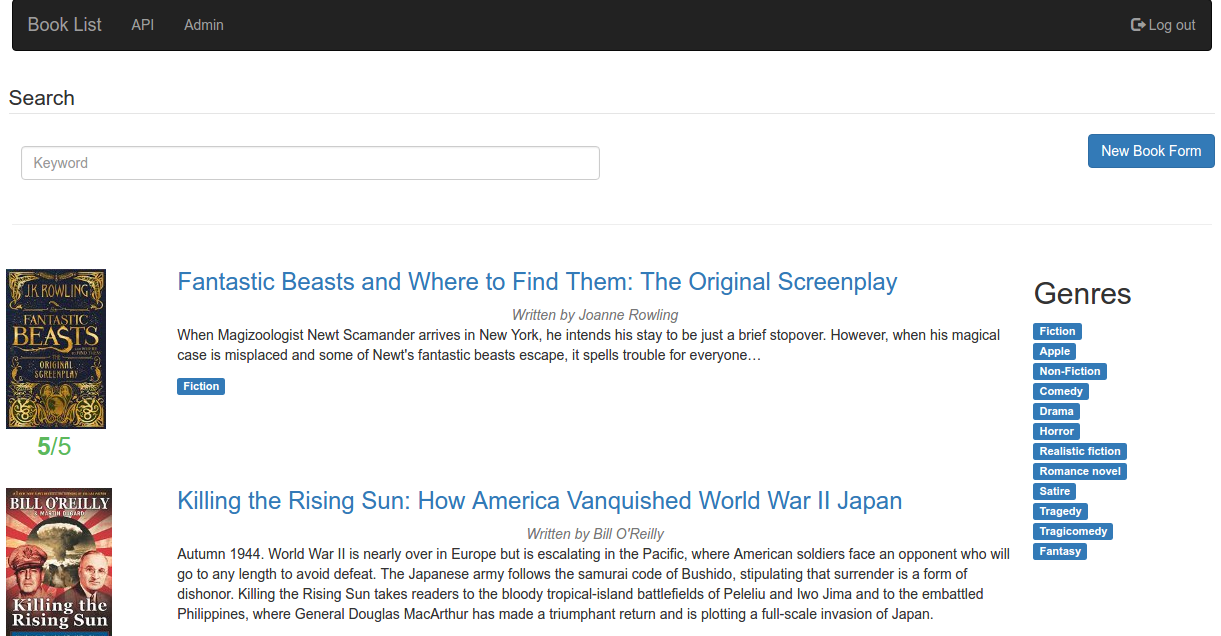
post 'books/:id/approve', to: 'books#approve', as: 'approve\_book'

end

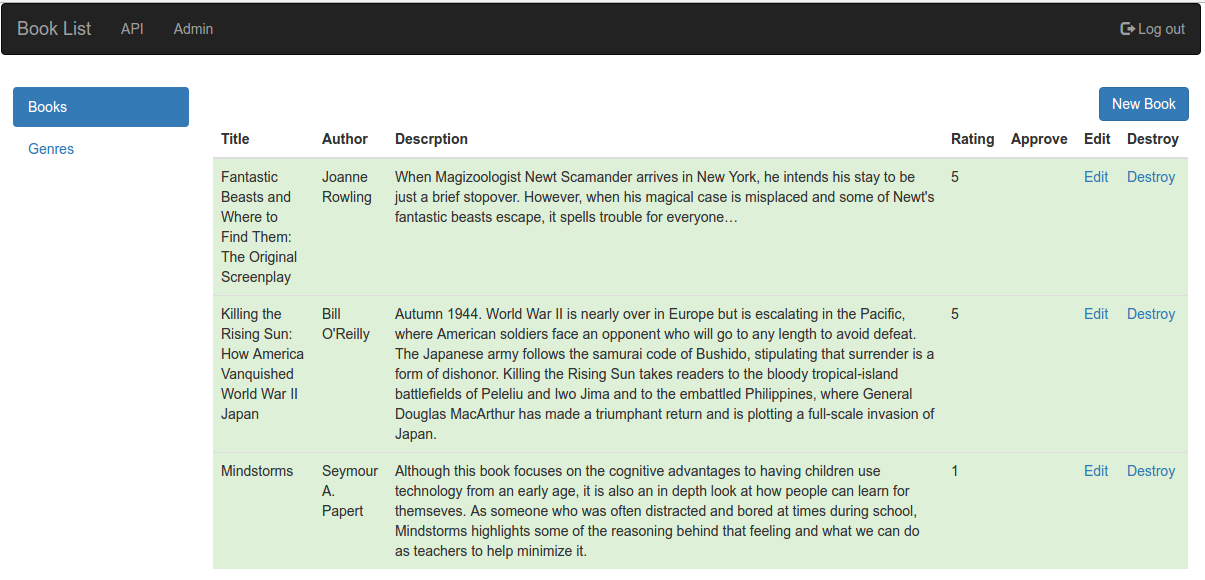
get 'static\_pages/api', as: 'api'

end

**Вигляд основної частини сайту:**



**Вигляд адмін панелі:**



Весь проект знаходиться за адрессою: https://github.com/ivanziola1/reading\_list

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