Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

О Т Ч Е Т

по домашней работе № 1

Название: Зачётная работа

Дисциплина: Интернет-Программирование

Студент	ИУ6-32Б		К.В. Гринченко
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2021

Задание

Основная задача зачётной работы - продемонстрировать полученные знания в создании собственного веб-приложения. Тему зачётной работы допустимо выбрать самостоятельно, но обязательно согласовать с преподавателем.

Обязательные требования к программе:

- Для реализации использовать Ruby on Rails.
- Необходимо иметь контроллеры, обеспечивающие обработку запросов.
- Необходимо использовать модели для хранения данных в БД.
- Необходимо обеспечить аутентификацию пользователей.
- При реализации клиентской части необходимо применить код на языке Javascript и таблицы стилей CSS.
- Провести интернационализацию приложения и обеспечить вывод надписей на русском языке (см. пример в лекции 11).

Цель работы

Написать веб-приложение с регистрацией, возможностью создавать, просматривать, редактировать, удалять и комментировать посты с оцениванием игр и отзывами на них, учитывая все пункты задания.

Исходный код

Контроллеры

code app/controllers/application_controller.rb

```
# frozen_string_literal: true

# application_controller
class ApplicationController < ActionController::Base
  def self.default_url_options
    { locale: I18n.locale }
  end

  around_action :switch_locale

  def switch_locale(&action)
    locale = locale_from_url || session[:locale] || I18n.default_locale
    I18n.with_locale locale, &action
  end

  def locale_from_url
    locale = params[:locale]

    return locale if I18n.available_locales.map(&:to_s).include?(locale)

    nil
  end
end
```

code app/controllers/categories_controller.rb

```
# frozen_string_literal: true

# categories_controller
class CategoriesController < ApplicationController
  before_action :authenticate_user!
  before_action :set_category, only: %i[show edit update destroy]

  def index
    @categories = Category.all
  end

  def new
    @category = Category.new
  end

  def show
    @posts = Post.where(category_id: @category.id).order('created_at
DESC').paginate(page: params[:page], per_page: 10)
  end

  def edit
    render :edit
  end

  def create
    @category = Category.new(category_params)

    if @category.save
      redirect_to categories_path(locale: I18n.locale)
    else
      render :new
    end
  end

  def update
    if @category.update(category_params)
      redirect_to categories_path(locale: I18n.locale)
    else
      render :edit
    end
  end

  def destroy
    @category.destroy
    redirect_to categories_path(locale: I18n.locale)
  end

  private

  def set_category
    @category = Category.find(params[:id])
  end

  def categories_params
    params.require(:categories).permit(:name)
  end
```

```
  def category_params
    params.require(:category).permit(:name)
  end
end
```

code app/controllers/comments_controller.rb

```
# frozen_string_literal: true

# comments_controller
class CommentsController < ApplicationController
  before_action :authenticate_user!

  def create
    @post = Post.find(params[:post_id])
    puts comment_params.to_json
    @comment = @post.comments.create(comment_params)
    @comment.post_id = @post.id

    if @comment.save
      redirect_to post_path(@post)
    else
      render :new, status: :unprocessable_entity
    end
  end

  def destroy
    @post = Post.find(params[:post_id])
    @comment = @post.comments.find(params[:id])
    return unless @comment.user_id == current_user.id
    @comment.destroy
    redirect_to post_url(@post, locale: I18n.locale)
  end

  private

  def comments_params
    params.require(:comments).permit(:user, :body).merge(user_id: current_user.id)
  end

  def comment_params
    params.require(:comment).permit(:user, :body).merge(user_id: current_user.id)
  end
end
```

code app/controllers/posts_controller.rb

```
# frozen_string_literal: true

# posts_controller
class PostsController < ApplicationController
  before_action :authenticate_user!
  before_action :set_post, only: %i[show edit update destroy]

  def index
    @posts = Post.includes(:user, :comments).order('created_at DESC').paginate(page:
params[:page], per_page: 10)
  end
```

```
  def create_post
    @post = Post.new
  end

  def show; end

  def sponsors; end

  def edit
    render :edit
  end

  def create
    @post = Post.new(post_params)
    @post.user_id = current_user.id

    if @post.save
      redirect_to @post
    else
      render :create_post
    end
  end

  def update
    return unless @post.user_id == current_user.id
    if @post.update(post_params)
      flash[:success] = 'Post item successfully updated!'
      redirect_to post_url(@post, locale: I18n.locale)
    else
      flash.now[:error] = 'Post item update failed'
      render :edit
    end
  end

  def destroy
    return unless @post.user_id == current_user.id

    @post.destroy
    redirect_to root_path(locale: I18n.locale)
  end

  private

  def set_post
    @post = Post.find(params[:id])
  end

  def posts_params
    params.require(:posts).permit(:title, :body, :rating, :category_id).merge(user_id:
current_user.id)
  end

  def post_params
    params.require(:post).permit(:title, :body, :rating, :category_id).merge(user_id:
current_user.id)
  end
end
```

code app/controllers/users_controller.rb

```
# frozen_string_literal: true

# users_controller
class UsersController < ApplicationController
  before_action :authenticate_user!

  def index
    @users = User.all
  end

  def show
    @user = User.last
  end

  def show_user
    @user = User.find(params[:id])
  end

  def new
    @user = User.new
  end

  def create
    @user = User.new(user_params)

    respond_to do |format|
      if @user.save
        sign_in @user
        format.html { redirect_to @user, notice: 'Пользователь успешно создан' }
      else
        format.html { render :new }
      end
    end
  end

  def destroy
    @user = User.find(params[:id])
    return unless @user.id == current_user.id
    @user.destroy

    respond_to do |format|
      format.html { redirect_to users_url, notice: 'Пользователь был успешно удален' }
    end
  end

  private

  def set_user
    @user = User.find(params[:id])
  end

  def user_params
    params.require(:user).permit(:username, :password)
  end
end
```

Модели

code app/models/category.rb

```
class Category < ApplicationRecord
  validates :name, presence: true
  has_many :posts
end
```

code app/models/comment.rb

```
class Comment < ApplicationRecord
  belongs_to :user
  belongs_to :post

  def created_at
    attributes['created_at'].strftime("%d.%m.%Y %H:%m")
  end
end
```

code app/models/post.rb

```
class Post < ApplicationRecord
  belongs_to :user
  belongs_to :category
  has_many :comments, dependent: :destroy

  validates :title, :body, :category_id, :rating, presence: true

  def created_at
    attributes['created_at'].strftime("%d.%m.%Y %H:%m")
  end
end
```

code app/models/user.rb

```
class User < ApplicationRecord
  has_many :posts, dependent: :destroy
  has_many :comments, dependent: :destroy

  def created_at
    attributes['created_at'].strftime("%d.%m.%Y %H:%m")
  end

  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :validatable
end
```

Маршруты

code config/routes.rb

```
Rails.application.routes.draw do
  scope "(:locale)", locale: /#{I18n.available_locales.join("|")}/ do
    devise_for :users
    root "posts#index"

    get "/users" => "users#index", as: "users"
    get "/create" => "posts#create_post", as: "create"
    get "/posts/:id" => "posts#show"
    get "/edit/:id" => "posts#edit", as: "edit"
    get '/category/create' => "categories#new", as: "create_category"

    patch "posts.:id" => "posts#update", as: :update_post
```

```
    patch "categories.:id" => "categories#update", as: :update_category
    get '/:locale.:id' => 'posts#destroy', as: :destroy_post
    get 'users/:id' => 'users#show_user', as: :show_user
    get 'users/:id' => 'users#destroy', as: :destroy_user
    get 'categories/:id' => 'categories#show', as: :show_category
    get 'categories/:id' => 'categories#destroy', as: :destroy_category

    get 'posts/:post_id/comments(.:format)' => 'comment#create', as: :create_comment

    resources :users
    resources :posts do
      resources :comments
    end
    resources :categories
  end
end
```

Тесты

test test/integration/session_flows_test.rb

```
require 'test_helper'

class SessionFlowsTest < ActionDispatch::IntegrationTest
  include Devise::Test::IntegrationHelpers

  setup do
    @post = posts(:post_one)
    @category = categories(:one)
    @comment = comments(:comment_one)
  end

  test 'unauthorized user will be redirected to login page' do
    get root_url
    assert :redirect
  end

  test 'user with incorrect credentials will be redirected to login page' do
    post user_session_url, params: { name: Faker::Lorem.word, password:
Faker::Lorem.word }
    assert controller: :session, action: :login
  end

  test 'user with correct credentials will see the root' do
    password = Faker::Lorem.word

    user = User.create(name: Faker::Lorem.word, password: password,
password_confirmation: password)

    post user_session_url, params: { name: user.name, password: password }

    assert controller: :posts, action: :index
  end

  test 'user will see the root after signing up' do
    username = Faker::Lorem.word
    password = Faker::Lorem.word

    post users_url, params: { user: { name: username, password: password,
password_confirmation: password } }
```

```
    assert :success
  end

  test 'user can logout' do
    password = Faker::Lorem.word

    user = User.create(name: Faker::Lorem.word, password: password,
password_confirmation: password)

    post user_session_url, params: { name: user.name, password: password }

    get destroy_user_session_url

    assert controller: :session, action: :login
  end

  test 'user can create post' do
    username = Faker::Lorem.word
    password = Faker::Lorem.word

    post users_url, params: { user: { name: username, password: password,
password_confirmation: password } }
    get create_url, params: {post: { title: @post.title, body: @post.body } }

    assert :success
  end

  test 'user can create category' do
    username = Faker::Lorem.word
    password = Faker::Lorem.word

    post users_url, params: { user: { name: username, password: password,
password_confirmation: password } }
    get create_category_url, params: {category: { name: @category.name } }

    assert :success
  end
end
```

test test/models/category_test.rb

```
require 'test_helper'

class CategoryTest < ActiveSupport::TestCase
  test 'should not save category' do
    category = Category.new
    assert_not category.save
  end

  test 'should save post' do
    name = Faker::Lorem.word

    category = Category.new(name: name)
    assert category.save!
  end

  test 'should find post' do
    assert Category.find_by(name: 'MyString')
  end
end
```

test test/models/comment_test.rb

```
require 'test_helper'

class CommentTest < ActiveSupport::TestCase
  setup do
    @user = users(:example)
    @post = posts(:post_one)
  end

  test 'should not save comment' do
    comment = Comment.new
    assert_not comment.save
  end

  test 'should save comment' do
    body = Faker::Lorem.word
    summary = Faker::Lorem.word
    image = Faker::Lorem.word

    comment = Comment.create(body: body, :user_id => @user.id, :post_id => @post.id)
    assert comment.save!
  end

  test 'should find comment' do
    assert Comment.find_by(body: 'MyText')
  end
end
```

test test/models/post_test.rb

```
require 'test_helper'

class PostTest < ActiveSupport::TestCase
  setup do
    @user = users(:example)
    @category = categories(:one)
  end

  test 'should not save post' do
    post = Post.new
    assert_not post.save
  end

  test 'should save post' do
    title = Faker::Lorem.word
    body = Faker::Lorem.word
    summary = Faker::Lorem.word

    post = Post.new(title: title, summary: summary, body: body, rating: 4, :user_id =>
@user.id, :category_id => @category.id)
    assert post.save!
  end

  test 'should find post' do
    assert Post.find_by(title: 'MyString')
  end
end
```

test test/integration/user_test.rb

```
require 'test_helper'

class UserTest < ActiveSupport::TestCase
  test 'should not save user' do
    user = User.new
    assert_not user.save
  end

  test 'should save user' do
    user = User.new(email: 'example@mail.ru', password: 'password')
    assert user.save!
  end

  test 'should find user' do
    assert User.find_by(name: 'example@mail.ru')
  end
end
```

Стили CSS

code app/assets/stylesheets/application.css

```
.post-menu__nav {
  padding-top: 80px;
  position: fixed;
  z-index: 20;
  display: flex;
  height: 100%;
  overflow-y: auto;
  left: -100%;
  transition: all 0.5s;
}

.post-menu__active .post-menu__nav {
  left: 0;
  transition: all 0.5s;
}

.post-menu__link {
  padding: 20px;
  font-size: 32px;
  text-decoration: none;
  text-transform: uppercase;
  letter-spacing: 5px;
  font-weight: bold;
  color: #fff;
  transition: all 0.4s;
}

.post-menu__link:hover {
  color: grey;
}

.post-additional {
  font-weight: bold;
}

html {
    height: 100%;
```

```
}

body {
    margin: 0;
    /* Растягиваем body по высоте html */
    min-height: 100%;
    display: grid;
    grid-template-rows: auto 1fr auto;
}

footer {
    background: #1d1d71;
}

h2, p {
    color: black;
}
```

Javascript

code app/assets/javascripts/main/main.js

```
function post_menu() {
  let menu = document.getElementById("menu-nav");
  menu.style.width = '0px';
  menu.style.display = menu.style.display == "none" ? "block" : "none";
  let cur_menu_width = 0, menu_width = 450;

  var interval = window.setInterval(function() {
      cur_menu_width += 100;
      menu.style.width = cur_menu_width + 'px';
      if (cur_menu_width >= menu_width) {
          window.clearInterval(interval);
      }
  });
}
```

Локализация

code config/locales/en.yml

```
en:
  hi: 'Welcome, '
  header:
    users: "Our users"
    log_out: "Log out"
    log_in: "Log in"
    registration: "Registration"
    create: "Create post"
    summary: Summary
    categories: All categories

  show_posts:
    none: No reviews yet

  edit_post: Edit post
  show:
    new: New post
    last_update: 'Last update at: '
    created_at: 'Created at: '
    created_by: 'Created by: '

  new_post:
    title: Title
```

```
  post: Note
  categories: Categories
  rating: Rating
  enter_title: Enter title
  enter_text: Enter text
  enter_categories: Enter tags, split by comma

users_table:
  users: Users
  username: User
  reg_date: Registration date

show_user:
  email: 'Email address:'
  reg_date: 'Registration date:'

comment:
  new_comm: New comment
  username: Username
  comm_text: Comment
  add: Add comment

edit: Edit
delete: Delete
cancel: Cancel
create: Create
sign_up: Sign up
enter_email: Enter email
enter_password: Enter password
password: Password
password_confirmation: Password confirmation
email: Email
remember: Remember me
body: Body
title: title
write: ' writes:'
sponsors: 'There are no sponsors('
categories: 'Categories: '
rating: 'Rating: '
new_category:
  new: New category
  name: Name
  enter_name: Enter name
  categories: Categories

add: 'Add'
no_rights: No rights to delete user
submit: Submit
final: Final Work
info_name: Made by Kirill Grinchenko IU6-32B
priv: "All rights reserved"
category: 'Category: '
show_category:
  new: New category
  name: Name
  categories: Categories

none: 'None'
not_saved: 'prohibited this post from being saved:'
```

```
  users_list:
    show: "Show"

  en: English
  ru: Russian
```

code config/locales/ru.yml

```
ru:
  hi: 'Добро пожаловать, '
  header:
    users: "Пользователи"
    log_out: "Выйти"
    log_in: "Войти"
    registration: "Регистрация"
    create: "Создать пост"
    categories: Все категории

  show_posts:
    none: Пока нет рецензий

  edit_note: Изменить запись
  show:
    new: Новая запись
    last_update: 'Обновлено: '
    created_at: 'Создано в: '
    created_by: 'Создано: '

  new_post:
    title: Название
    post: Текст
    categories: Категории
    rating: Рейтинг
    enter_title: Введите название
    enter_text: Введите текст
    enter_categories: Введите теги через запятую

  users_table:
    users: Пользователи
    username: Пользователь
    reg_date: Дата регистрации

  show_user:
    email: 'Email адрес:'
    reg_date: 'Дата регистрации:'

  comment:
    new_comm: Новый комментарий
    username: Имя пользователя
    comm_text: Комментарий
    add: Добавить комментарий

  edit: Изменить
  delete: Удалить
  cancel: Отмена
  create: Создать
  sign_up: Регистрация
  enter_email: Введите email
  enter_password: Введите пароль
  password: Пароль
```

```
  password_confirmation: Подтверждение пароля
  email: Email
  remember: Запомнить меня
  write: ' пишет:'
  categories: 'Категории: '
  rating: 'Рейтинг: '
  new_category:
    new: Новая категория
    name: Название
    enter_name: Введите название
    categories: Категории

  add: 'Добавить'
  no_rights: Нет прав на удаление пользователя
  final: Зачетная работа
  info_name: Сделал Гринченко Кирилл Группа ИУ6-32Б
  priv: "Все права защищены"
  category: 'Категория: '
  show_category:
    new: Новая категория
    name: Название
    categories: Категории

  none: 'Нет'
  not_saved: 'не позволяют сохранить пост:'
  users_list:
    show: "Показать"

  en: Английский
  ru: Русский
```

Формы

Layouts:

code app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html>
  <head>
    <title>Final work</title>
    <%= csrf_meta_tags %>
    <%= csp_meta_tag %>

    <%= stylesheet_link_tag
'https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css' %>
    <%= stylesheet_link_tag    'application', media: 'all', 'data-turbolinks-track': 'reload' %>
    <%= javascript_include_tag 'application', 'data-turbolinks-track': 'reload' %>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/
tWtIaxVXM" crossorigin="anonymous"></script>
  </head>

  <body>
    <header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom bg-dark
text-white">
      <div class="pos-f-t post-menu post-menu__active" id="menu" style="padding-left:
20px;">
        <nav class="navbar navbar-light bg-dark">
```

```
        <button class="navbar-toggler" id="button_menu" type="button" aria-
controls="navbarToggleExternalContent" aria-expanded="false" aria-label="Toggle
navigation" onclick="post_menu()">
          <span class="navbar-toggler-icon"></span>
        </button>
      </nav>
      <nav class="post-menu__nav bg-dark" id="menu-nav" style="display: none;">
        <ul>
          <% Category.all.each do |category| %>
            <li class="text-decoration-none text-white" style="list-style-type: none;">
              <%= link_to category.name, show_category_path(category, locale: I18n.locale),
class: "post-menu__link" %>
            </li>
          <% end %>
        </ul>
      </nav>

      <div class="post-menu__overlay">
      </div>
    </div>

    <a href="/<%= I18n.locale %>" class="d-flex align-items-center mb-3 mb-md-0 me-
md-auto text-white text-decoration-none">
      <svg class="bi me-2" width="40" height="32"></svg>
      <span class="fs-3 web-page-name"><strong>My Games</strong></span>
    </a>

    <ul class="nav nav-pills col-12 col-lg-auto mb-2 mb-md-0">
      <li class="dropdown">
        <%= link_to '', class: 'nav-link px-2 text-white dropdown-toggle',
          data: {"bs-toggle": 'dropdown'} do %>
          <%= t I18n.locale %>
        <% end %>

        <ul class="dropdown-menu">
          <% I18n.available_locales.each do |locale| %>
            <li>
              <% if I18n.locale != locale %>
                <%= link_to t(locale), url_for(locale: locale),
                  class: 'dropdown-item' %>
              <% end %>
            </li>
          <% end %>
        </ul>
      </li>
    </ul>

    <ul class="nav nav-pills">
      <% if user_signed_in? %>
        <li class="nav-item">
          <%= link_to t('header.create'), create_path, class: "btn btn-outline-light" %>
        </li>
        <li class="nav-item">
          <%= link_to t('header.users'), users_path, class: "nav-link px-2 text-white" %>
        </li>
          <li class="nav-item">
            <%= link_to t('header.categories'), categories_path, class: "nav-link px-2 text-
white" %>
          </li>
```

```
            <li class="nav-item" style="padding: .5rem;"><%= t('hi') + current_user.email +
'!' %></li>
            <li class="nav-item">
              <%= link_to t('header.log_out'), destroy_user_session_path(locale: I18n.locale),
method: :delete, class: "nav-link px-2 text-white" %>
            </li>
          <% else %>
            <li class="nav-item">
              <%= link_to t('header.log_in'), new_user_session_path(locale: I18n.locale), class:
"nav-link px-2 text-white" %>
            </li>
            <li class="nav-item">
              <%= link_to t('header.registration'), new_user_registration_path(locale:
I18n.locale), class: "nav-link px-2 text-white" %>
            </li>
          <% end %>
        </ul>
      </header>

      <div class="container" id="content">
        <%= yield %>
      </div>
      <footer class="footer-07">
        <div class="container">
          <div class="row justify-content-center">
            <div class="col-md-12 text-center">
              <h2><%=t('info_name')%></h2>
            </div>
          </div>
          <div class="row justify-content-center">
            <div class="col-md-12 text-center">
              <p class="copyright">
                Copyright &copy; <%=t('priv')%>
              </p>
            </div>
          </div>
        </div>
      </footer>
    </body>
  </html>
```

Categories:

code app/views/categories/_form.html.erb

```
<%= form_with(model: category, local: true) do |form| %>
  <% if category.errors.any? %>
    <div class="alert alert-danger">
      <h2><%= pluralize(category.errors.count, "error") + ' ' + t('not_saved') %></h2>

      <ul>
        <% category.errors.full_messages.each do |message| %>
          <li><%= message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>

  <div class="field">
    <label><%= t('new_category.name') %></label>
```

```
    <%= form.text_field :name, class: "form-control", id: "exampleFormControlInput1",
placeholder: t('new_category.enter_name') %>
  </div><br />

  <%= form.submit t('add'), :class => 'btn btn-outline-dark me-2' %>
  <%= link_to t('cancel'), categories_path(locale: I18n.locale), class: 'btn btn-outline-dark
me-2' %>
<% end %>
```

code app/views/categories/edit.html.erb

```
<div class="container">
  <h3><%= t('edit_category') %></h3>

  <%= render 'form', category: @category %>
</div>
```

code app/views/categories/index.html.erb

```
<div class="container">
  <div style="float: left;">
    <h3><%= t('show_category.categories')%></h3>
  </div>
  <div style="float: right;">
    <%= link_to t('show_category.new'), create_category_path(locale: locale), class: 'btn
btn-outline-dark me-2' %>
  </div>


  <table class="table table-striped">
    <thead>
      <tr>
        <th>№</th>
        <th><%= t('show_category.name')%></th>
        <th colspan="2"></th>
      </tr>
    </thead>

    <tbody>
      <% i = 1 %>
      <% @categories.each do |category| %>
        <tr>
          <td><%= i %></td>
          <td><%= category.name %></td>
          <td><%= link_to t('edit'), edit_category_path(:id => category.id, locale: locale)
%></td>
          <td><%= link_to t('delete'), destroy_category_path(:id => category.id, locale:
locale), data: { confirm: 'You sure?' }, method: :delete %></td>
        </tr>
        <% i += 1 %>
      <% end %>
    </tbody>
  </table>
</div>
```

code app/views/categories/new.html.erb

```
<div class="container">
  <h3><%= t('new_category.new')%></h3><br />

  <%= render 'form', category: @category %>
```

```
</div>
```

code app/views/categories/show.html.erb

```
<h2><%= t('category') + @category.name %></h2>

<% @posts.each do |post| %>
  <article class="post">
    <section class="post-head">
      <h2><%= link_to post.title, post_path(post) %></h2>
    </section>
    <section class="post-additional">
      <%= t('category') %>
      <% if post.category.present? %>
        <%= link_to post.category.name, post.category %><br />
      <% else %>
        <%= 'None' %><br />
      <% end %>
      <%= t('rating') + post.rating.to_s + '/5.0' %>
    </section><br />
    <section class="post-summary">
      <%= post.body[0..250] + (post.body.to_s.length > 250 ? "..." : "") %>
    </section><br />
    <section class="services" style="color: grey;">
      <% id = post.user_id.to_i %>
      <% user = User.find(id).email %>
      <%= t('show.created_by') + user %>
      |
      <small><%= t('show.created_at') + post.created_at %></small>
      <% if post.user_id == current_user.id %>
        |
        <%= link_to t('edit'), edit_path(post, locale: I18n.locale) %>
        |
        <%= link_to t('delete'), post_path(post,  locale: I18n.locale), data: {method: :delete,
confirm: 'You sure?'} %>
      <% end %>
    </section>
  </article>
  <hr />
<% end %>

<% if @posts.length == 0 %>
  <p id="no-notes" style="color: grey; align: center; text-align: center; vertical-align:
middle; line-height: 300px;"><%= t('show_posts.none') %></p>
<% end %>

<%= will_paginate @posts, renderer: WillPaginate::ActionView::BootstrapLinkRenderer %>
```

Posts:

code app/views/posts/_form.html.erb

```
<%= form_with(model: post, local: true) do |form| %>
  <% if post.errors.any? %>
      <div class="alert alert-danger">
        <h2><%= pluralize(post.errors.count, "error") + ' ' + t('not_saved') %></h2>

        <ul>
          <% post.errors.full_messages.each do |message| %>
            <li><%= message %></li>
          <% end %>
```

```
      </ul>
    </div>
  <% end %>

  <div class="field form-group">
    <label for="exampleFormControlTextarea1"><%= t('new_post.title')%></label>
    <%= form.text_field :title, class: "form-control", id: "exampleFormControlInput1",
placeholder: t('new_post.enter_title') %>
  </div><br />

  <div class="form-group">
    <label for="exampleFormControlSelect1"><%= t('new_post.rating') %></label>
    <%= form.select :rating, options_for_select([["1", 1], ["2", 2], ["3", 3], ["4", 4], ["5", 5]]),
{},  class: "form-select", aria_label: "Default select example" %>
  </div><br />

  <div class="field form-group">
    <label for="exampleFormControlTextarea1"><%= t('new_post.categories')%></label>
    <%= form.collection_select :category_id, Category.order(:name), :id, :name,
{include_blank: true}, class: "form-select" %>
  </div><br />

  <div class="field form-group">
    <label for="exampleFormControlTextarea1"><%= t('new_post.post') %></label>
    <%= form.text_area :body, class: "form-control", id: "exampleFormControlTextarea1",
rows: "3", placeholder: t('new_post.enter_text') %>
  </div><br />

  <div class="actions">
    <%= form.submit t('add'), class: "btn btn-outline-dark me-2" %>
    <%= link_to t('cancel'), root_path(locale: I18n.locale), class: 'btn btn-outline-dark me-2'
%>
  </div>
<% end %>
```

code app/views/posts/create_post.html.erb

```
<div class="container">
  <h3><%= t('show.new')%></h3><br />

  <%= render 'form', post: @post %>
</div>
```

code app/views/posts/edit.html.erb

```
<div class="container">
  <h3><%= t('edit_post') %></h3>

  <%= render 'form', post: @post %>
</div>
```

code app/views/posts/index.html.erb

```
<% @posts.each do |post| %>
  <article class="post">
    <section class="post-head">
      <h2><%= link_to post.title, post_path(post) %></h2>
    </section>
    <section class="post-additional text-muted">
      <%= t('category') %>
      <% if post.category.present? %>
```

```
      <%= link_to post.category.name, post.category %><br />
    <% else %>
      <%= t('none') %><br />
    <% end %>
    <%= t('rating') + post.rating.to_s + '/5.0' %>
  </section><br />
  <section class="post-summary">
    <%= post.body[0..250] + (post.body.to_s.length > 250 ? "..." : "") %>
  </section><br />
  <section class="services" style="color: grey;">
    <% id = post.user_id.to_i %>
    <% user = User.find(id).email %>
    <%= t('show.created_by') + user %>
    |
    <small><%= t('show.created_at') + post.created_at %></small>
    <% if post.user_id == current_user.id %>
      |
      <%= link_to t('edit'), edit_path(post, locale: I18n.locale) %>
      |
      <%= link_to t('delete'), post_path(post,  locale: I18n.locale), data: {method: :delete,
confirm: 'You sure?'} %>
    <% end %>
  </section>
</article>
<hr />
<% end %>

<% if @posts.length == 0 %>
  <p id="no-notes" style="color: grey; align: center; text-align: center; vertical-align:
middle; line-height: 300px;"><%= t('show_posts.none') %></p>
<% end %>

<%= will_paginate @posts, renderer: WillPaginate::ActionView::BootstrapLinkRenderer,
style: "backgroundColor: #FFFBF0;" %>
```

code app/views/posts/show.html.erb

```
<article class="post">
  <section class="post-head">
    <h2><%= @post.title %></h2>
  </section>
  <section class="service">
    <small><%= t('show.created_at') + @post.created_at %></small>
    <% if @post.user_id == current_user.id %>
      |
      <%= link_to t('edit'), edit_path(@post, locale: I18n.locale) %>
      |
      <%= link_to t('delete'), post_path(@post,  locale: I18n.locale), data: {method: :delete,
confirm: 'You sure?'} %>
    <% end %>
  </section><br />
  <section class="post-additional">
    <%= t('category') %>
    <% if @post.category.present? %>
      <%= link_to @post.category.name, @post.category, class: "can-underline" %><br />
    <% else %>
      <%= t('none') %><br />
    <% end %>
    <%= t('rating') + @post.rating.to_s + '/5.0' %>
  </section><br />
```

```
  <section class="post-body">
    <p><%= @post.body %></p>
  </section>
</article><br /><hr>

<h4><%= t('comment.add') %></h4><br />

<%= form_for([@post, @post.comments.build]) do |form| %>
  <div class="field form-group">
    <label for="exampleFormControlTextarea1"><%= t('comment.comm_text')
%></label>
    <%= form.text_area :body, class: "form-control", id: "exampleFormControlTextarea1",
rows: "3" %>
  </div><br />

  <div class="actions">
    <%= form.submit t('comment.add'), class: "btn btn-outline-dark me-2" %>
  </div>
<% end %><br />

<div id='comments'>
  <% @post.comments.reverse().each do |comm| %>
    <div class="comment alert">
      <% id = comm.user_id.to_i %>
      <% if id != 0 %>
        <h5><%= user = User.find(id).email + t('write') %></h5>

        <p>
          <%= comm.body %>
        </p>
        <small><%= t('show.created_at') + comm.created_at %></small>
        <% if id == current_user.id %>
          |
          <%= link_to t('delete'), post_comment_path(comm.post, comm, locale: I18n.locale),
data: {method: :delete, confirm: 'You sure?'} %>
        <% end %>
      <% end %>
    </div>
  <% end %>
</div>

<script src="main.js"></script>
```

Users:

code app/views/users/index.html.erb

```
<div class="container">
  <h3><%= t('users_table.users')%></h3>

  <table class="table table-striped">
    <thead>
      <tr>
        <th><%= t('users_table.username')%></th>
        <th><%= t('users_table.reg_date')%></th>
        <th colspan="2"></th>
      </tr>
    </thead>

    <tbody>
```

```
    <% @users.each do |user| %>
      <tr>
        <td><%= user.email %></td>
        <td><%= user.created_at %></td>
        <td><%= link_to t('users_list.show'), show_user_path(:id => user.id, locale: locale),
class: "can-underline" %></td>
        <% if user.id == current_user.id %>
          <td><%= link_to t('delete'), destroy_user_path(:id => user.id, locale: locale), data:
{ confirm: 'You sure?' }, method: :delete, class: "can-underline" %></td>
        <% else %>
        <td><%= t('no_rights') %></td>
        <% end %>
      </tr>
    <% end %>
  </tbody>
 </table>
</div>
```

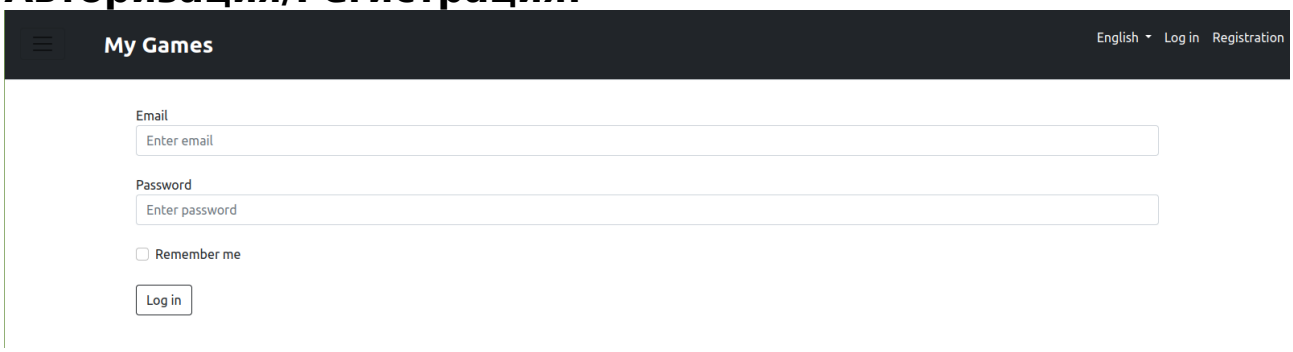
code app/views/users/show_user.html.erb

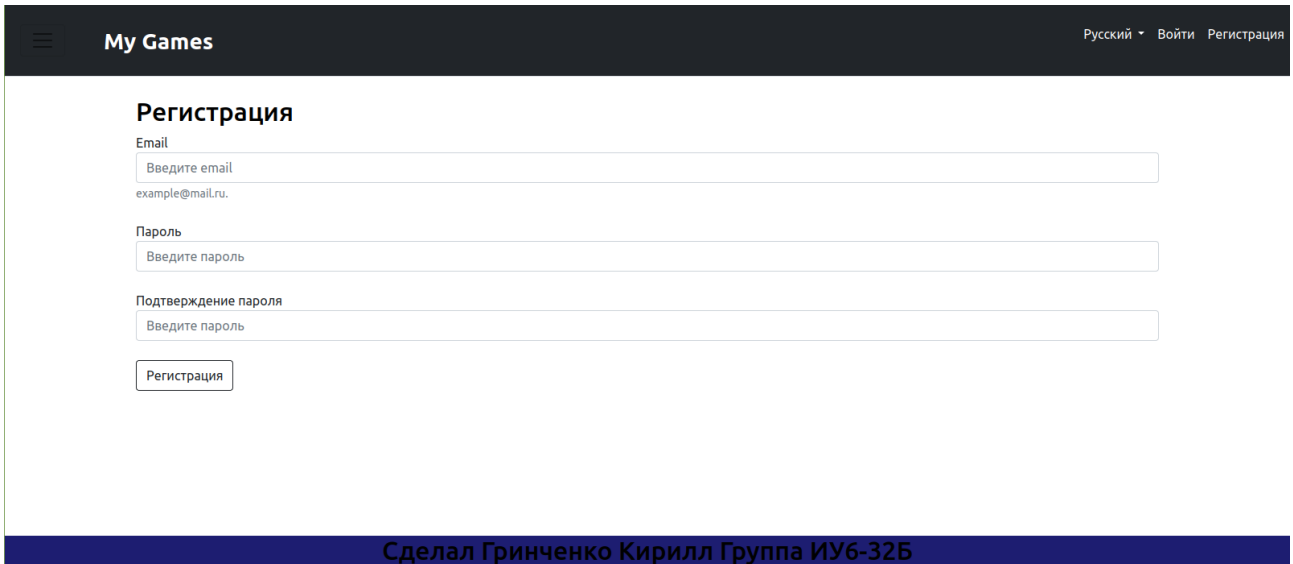
```
<div class="container">
 <p>
  <strong><%= t('show_user.email') %></strong>
  <%= @user.email %>
 </p>

 <p>
  <strong><%= t('show_user.reg_date') %></strong>
  <%= @user.created_at %>
 </p>
</div>
```

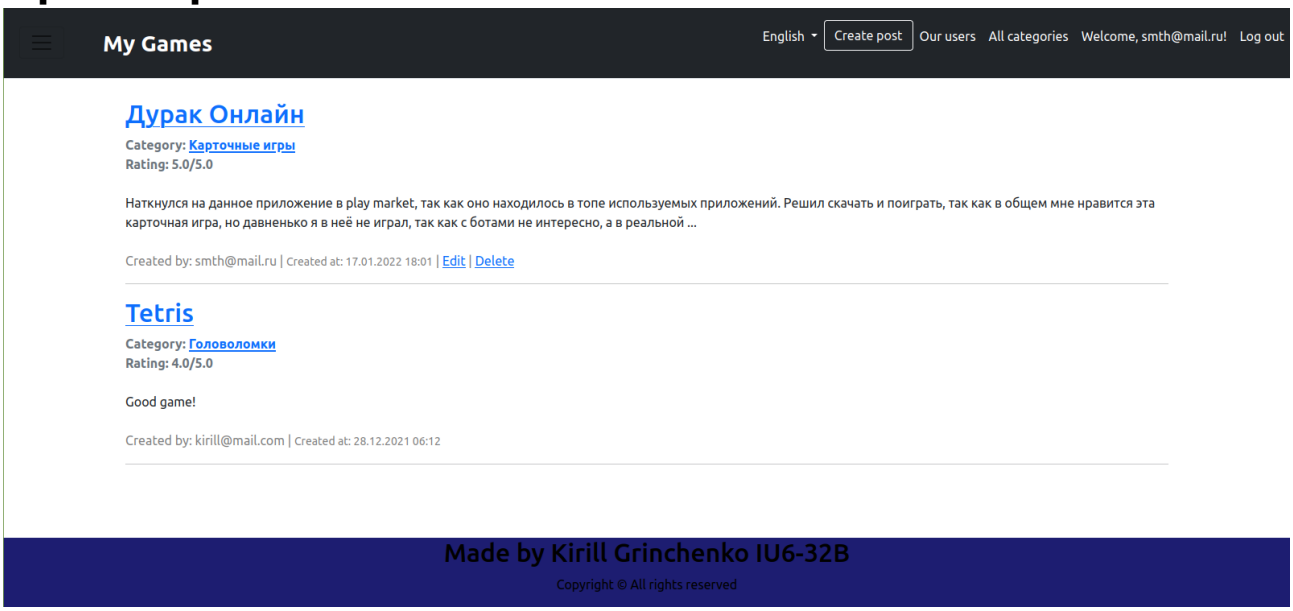
Результаты

Авторизация/Регистрация:

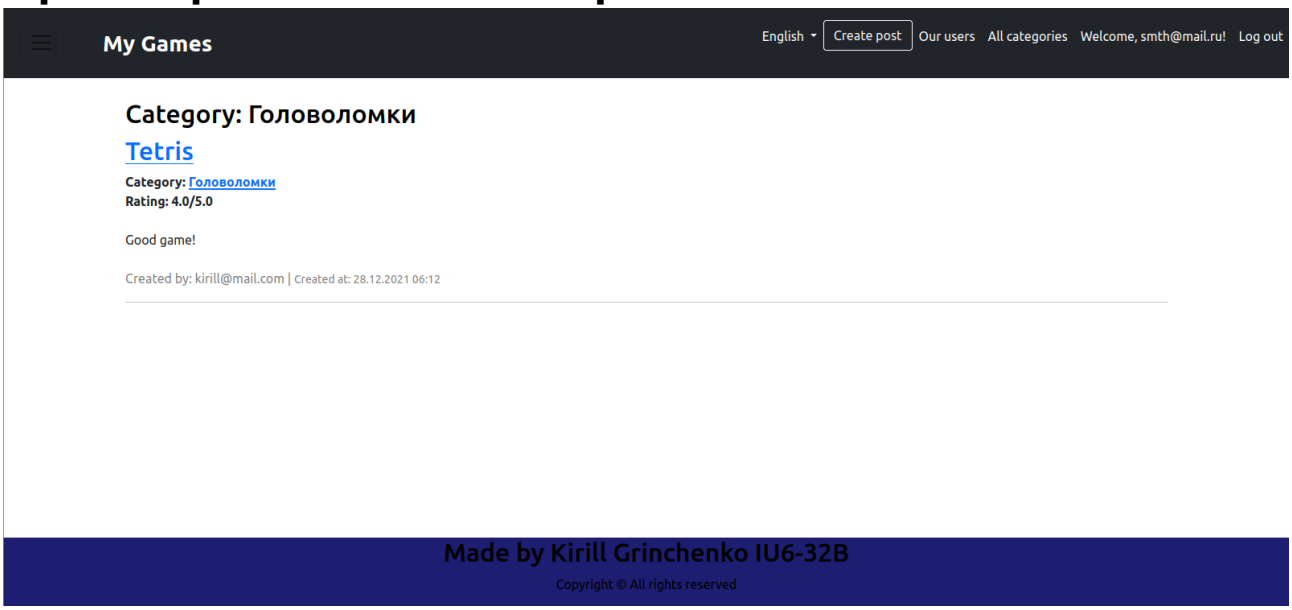




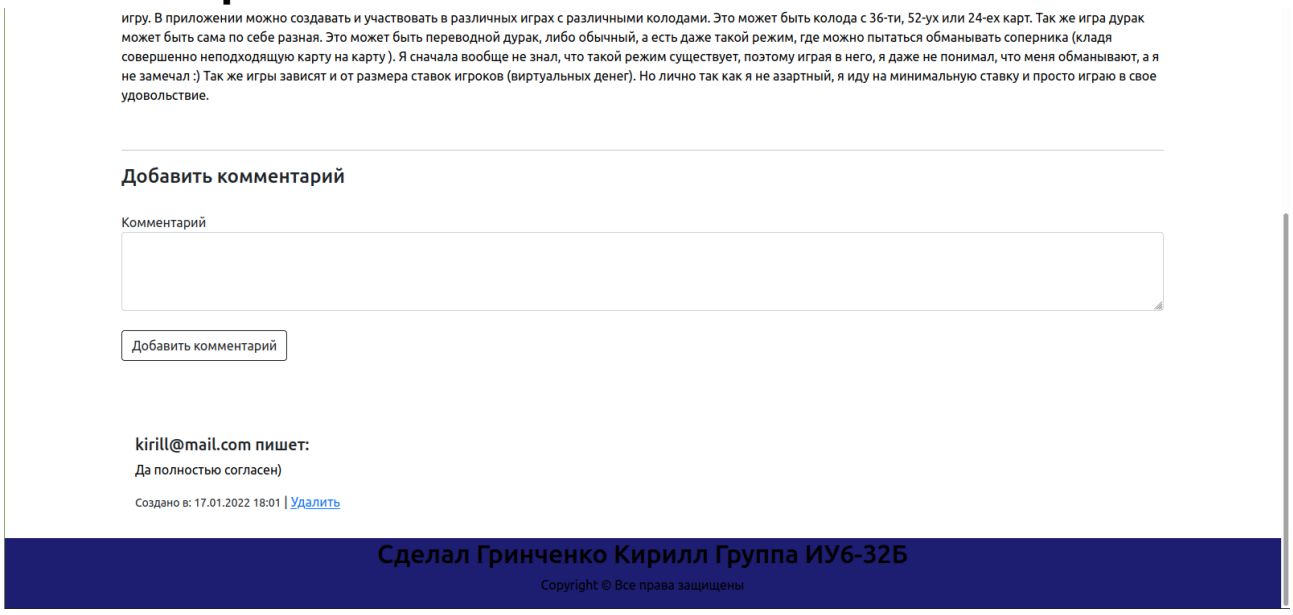
Просмотр постов:



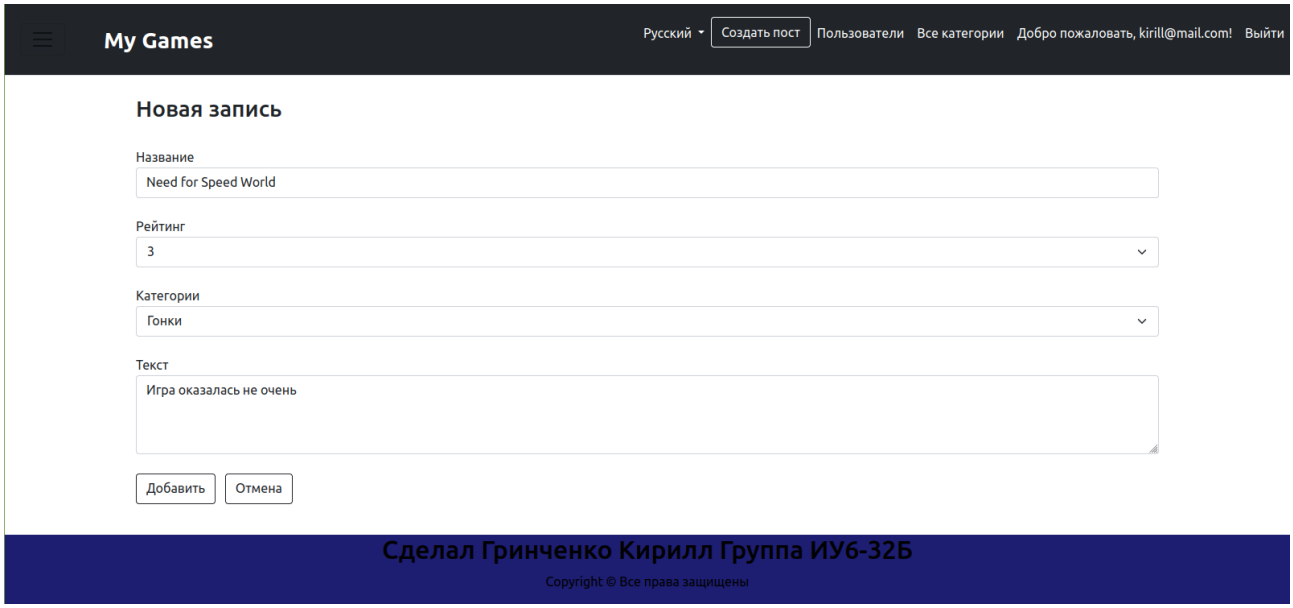
Просмотр постов по категориям:



Комментирование поста:



Создание поста:



Тестирование

kirill@kirill-W9x0LU:~/Рабочий стол/ИП/FinalWork/FinalWork$ rails test
Running via Spring preloader in process 57918
Run options: --seed 59246

Running:

....

Finished in 8.595116s, 2.2106 runs/s, 2.2106 assertions/s.
19runs, 19 assertions, 0 failures, 0 errors, 0 skips

Rubocop

kirill@kirill-W9x0LU:~/Рабочий стол/ИП/FinalWork/FinalWork/app/controllers$ rubocop *
Inspecting 5 files
.....

5 files inspected, no offenses detected

Вывод

В ходе данной работы были совмещены и продемонстрированы все знания, полученные мной за данный курс, для создания собственного веб-приложения на Ruby on Rails.