| Gerb-BMSTU_01 | **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**

**ОТЧЕТ**

| **по зачетной работе №** | 1 |
| --- | --- |

**Дисциплина:** Языки Интернет-программирования

| Студент | ИУ6-33Б |  | 19.12.2021 | Хоанг Т.Ф. |
| --- | --- | --- | --- | --- |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

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

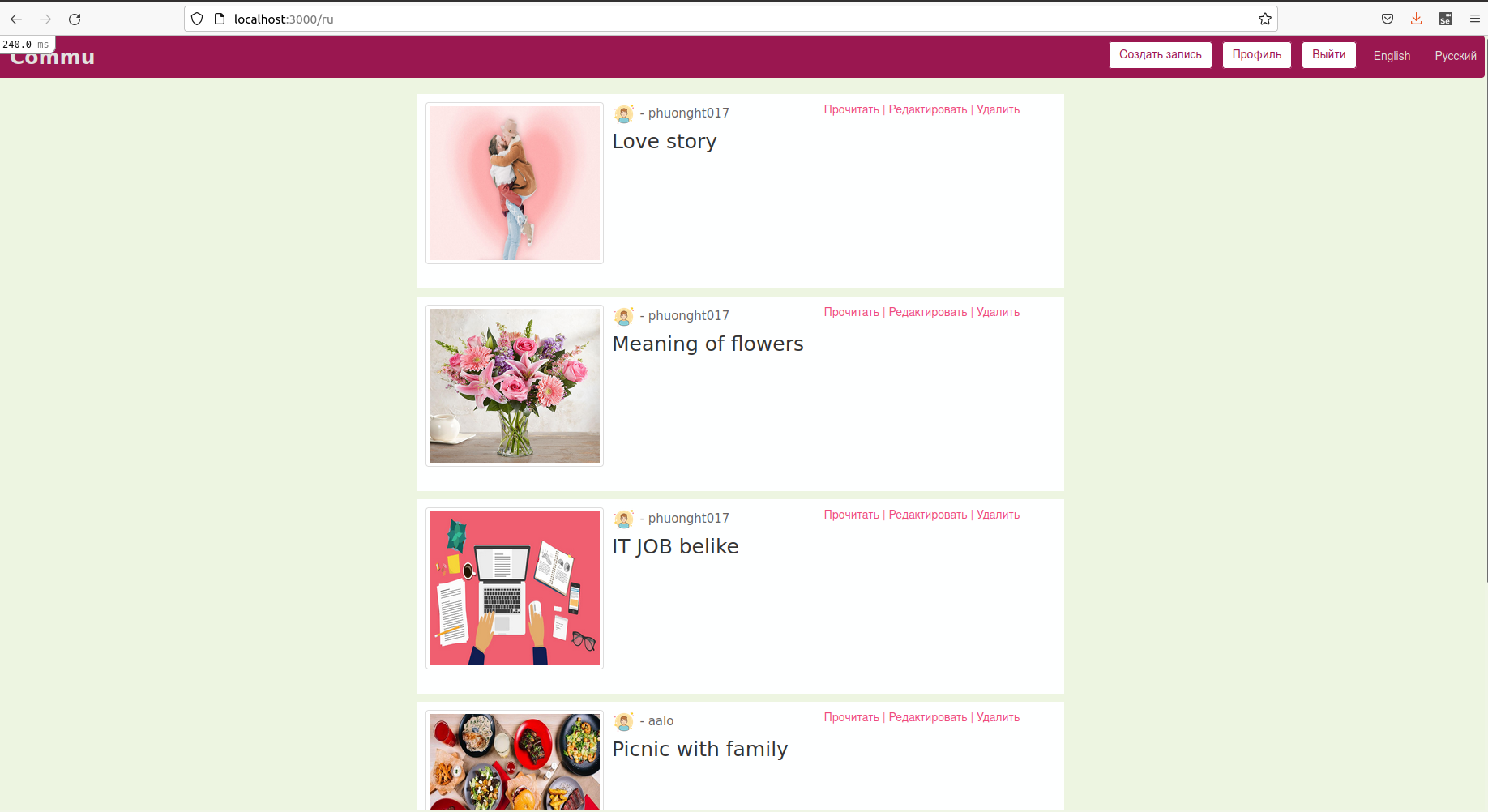
Обязательные требования к программе:

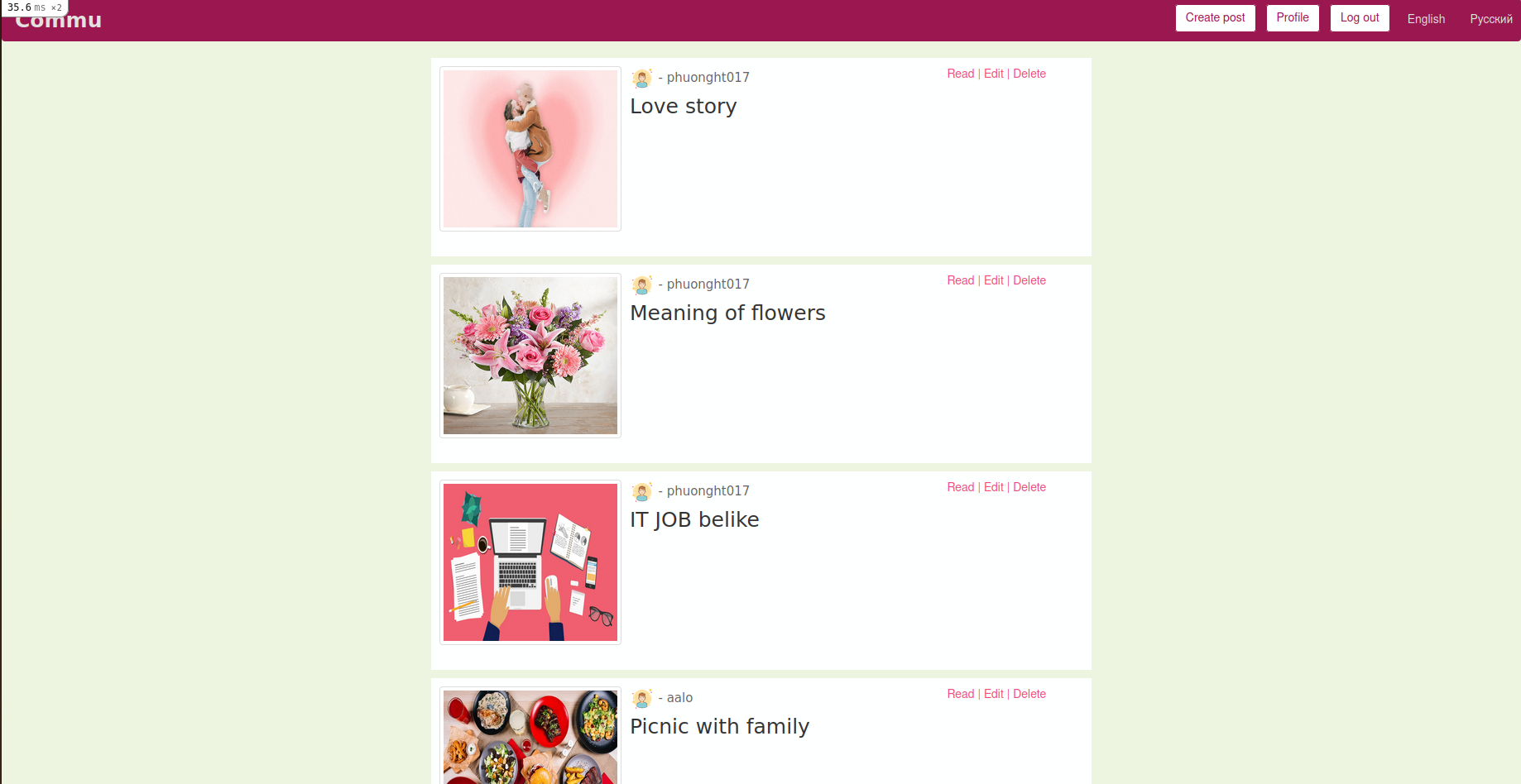
* Для реализации использовать Ruby on Rails.
* Необходимо иметь контроллеры, обеспечивающие обработку запросов.
* Необходимо использовать модели для хранения данных в БД.
* Необходимо обеспечить аутентификацию пользователей.
* При реализации клиентской части необходимо применить код на языке Javascript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

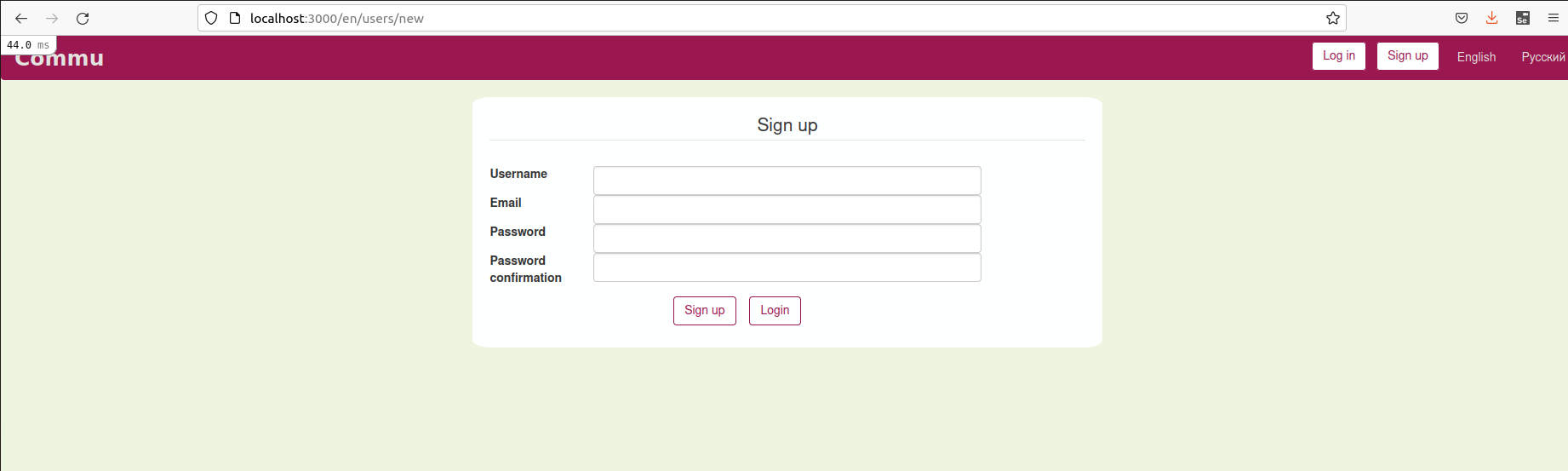
**Тема:**

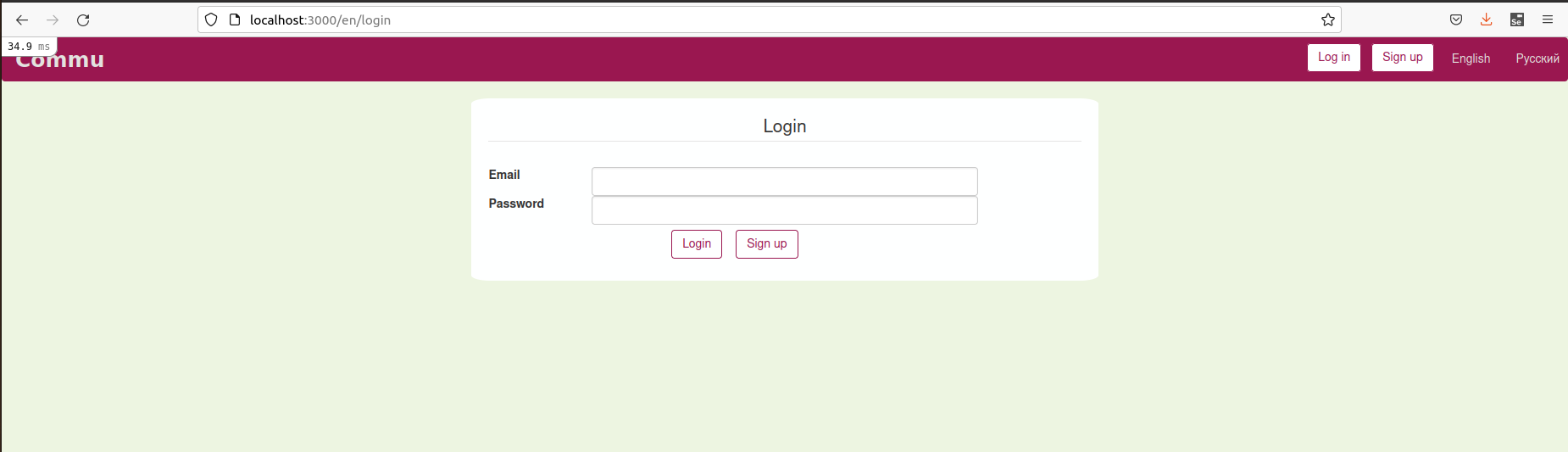
Cайт блога, который позволяет пользователям публиковать записи и оставлять комментарии

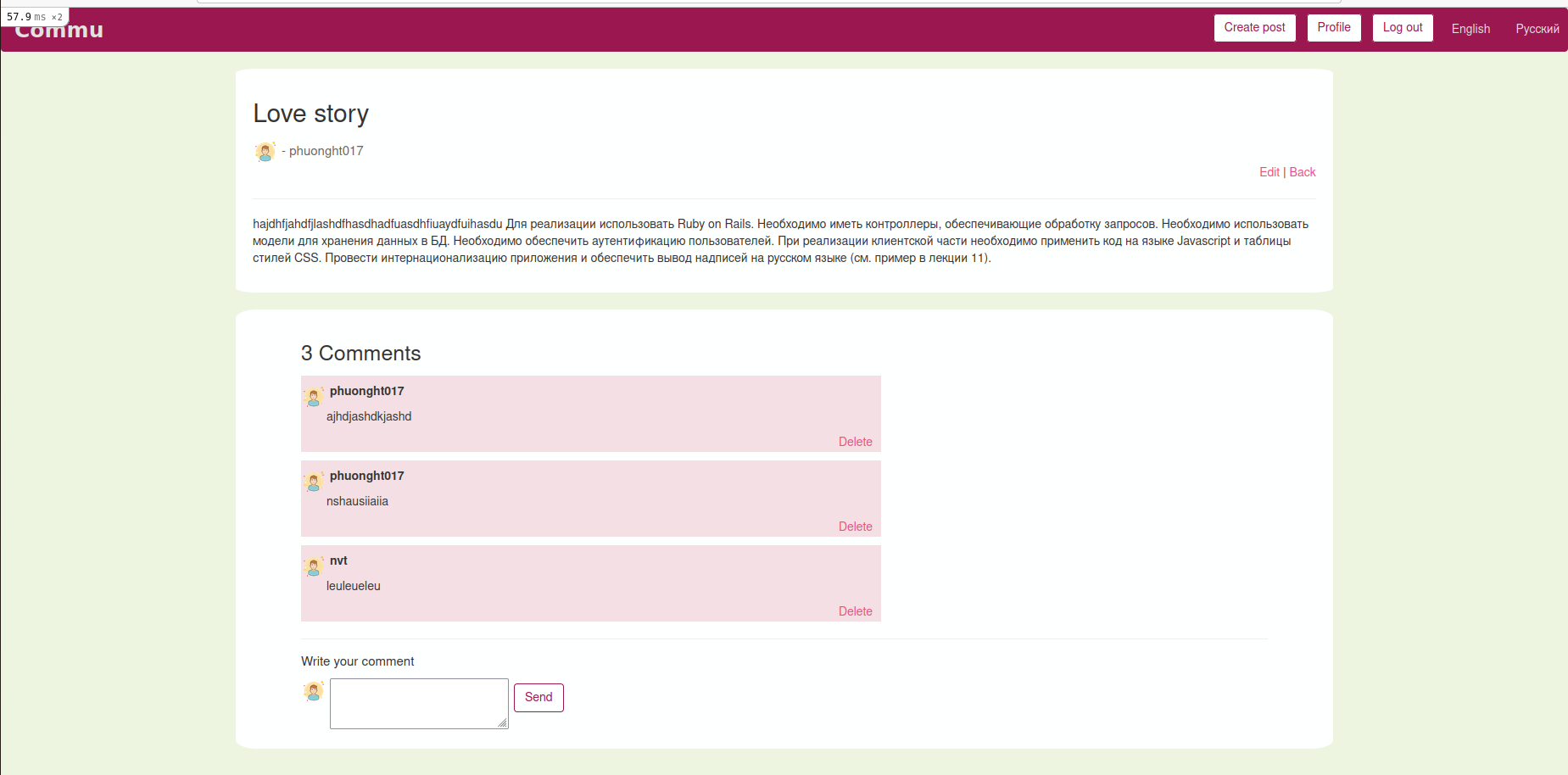
**Изображения форм интерфейса пользователя:**

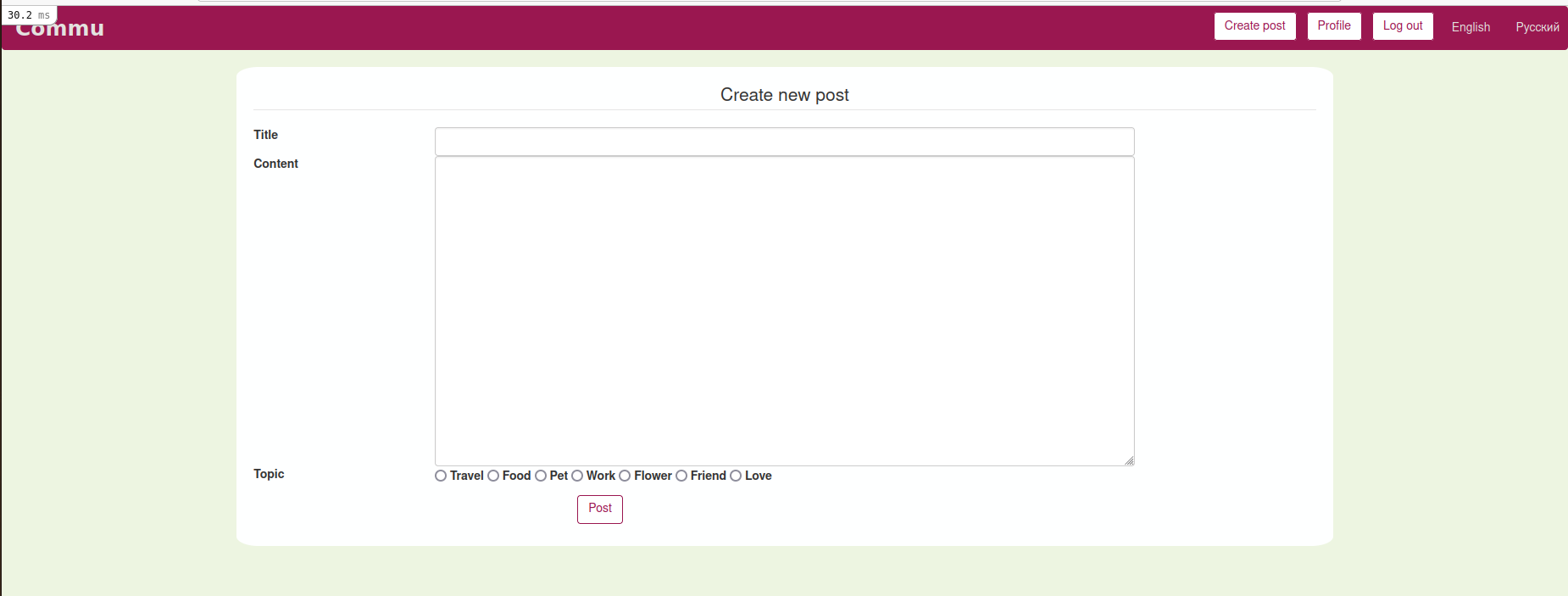
****

****

****

****

****

****

**Код:**

**routes:**

Rails.application.routes.draw do

scope ':locale', locale: /#{I18n.available\_locales.join("|")}/ do

resources :users

resources :posts do

resources :comments

end

resources :sessions, only: [:new, :create, :destroy]

get "signup", to: "users#new", as: "signup"

get "login", to: "sessions#new", as: "login"

post "login", to: "sessions#create"

get "logout", to: "sessions#destroy", as: "logout"

root 'posts#index'

delete '/post/:id', to: "posts#destroy", as: 'delete\_post'

end

post "login", to: "sessions#create"

get '', to: redirect("/#{I18n.default\_locale}")

end

**Controller:**

**application\_controller.rb**

class ApplicationController < ActionController::Base

protect\_from\_forgery

before\_action :set\_locale

def set\_locale

I18n.locale = params[:locale] if params[:locale].present?

end

def default\_url\_options(options = {})

options.merge(locale: I18n.locale)

end

helper\_method :current\_user

def current\_user

if session[:user\_id]

@current\_user ||= User.find(session[:user\_id])

else

@current\_user = nil

end

end

end

**users\_controller.rb**

class UsersController < ApplicationController

def index

if current\_user

@users = User.all

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def new

@user = User.new

end

def create

@user = User.new(user\_params)

if @user.save

redirect\_to :root #sau sua thanh root

else

redirect\_to :new\_user, notice: "Invalid information. Cannot register"

end

end

def show

if current\_user

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

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def update

if current\_user

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

if @user.update(user\_params)

redirect\_to :root #sau sua thanh root

else

redirect\_to :edit, notice: "Invalid information. Cannot save changes"

end

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def edit

if current\_user

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

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def destroy

if current\_user

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

@user.destroy

redirect\_to :root #sau se sua

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

private

def user\_params

params.require(:user).permit(:username, :email, :password)

end

end

**sessions.controller.rb**

class SessionsController < ApplicationController

def new

session[:user\_id] = nil;

end

def create

user = User.find\_by(email: params[:email])

if user&.authenticate(params[:password])

session[:user\_id] = user.id

session[:user\_name] = user.username

redirect\_to :root, notice: 'User logged in'

else

redirect\_to login\_path, notice: 'Invalid Email or Password'

end

end

def destroy

session[:users\_id] = nil

redirect\_to :login, notice: 'User logged out'

end

end

**posts\_controller.rb**

class PostsController < ApplicationController

def index

@posts = Post.all.order("created\_at DESC")

end

def new

if current\_user

@post = Post.new

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def create

if current\_user

@post = Post.new(user\_id: session[:user\_id], title: post\_params[:title], content: post\_params[:content], topic: post\_params[:topic])

if @post.save

redirect\_to posts\_path

else

render 'new'

end

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def show

if current\_user

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

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def update

if current\_user

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

if @post.update(post\_params)

redirect\_to @post

else

render 'edit'

end

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def edit

if current\_user

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

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def destroy

if current\_user

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

@post.destroy

puts "DESTROYED"

redirect\_to posts\_path

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

private

def post\_params

params.require(:post).permit(:title, :content, :topic)

end

end

**comments\_controller.rb**

class CommentsController < ApplicationController

def create

if current\_user

@post = Post.find(params[:post\_id])

@comment = @post.comments.create(user\_id: session[:user\_id], content: comment\_params[:content])

redirect\_to post\_path(@post)

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

def destroy

if current\_user

@post = Post.find(params[:post\_id])

@comment = @post.comments.find(params[:id])

@comment.destroy

redirect\_to post\_path(@post)

else

redirect\_to :login, notice: "You need to log in to access more pages"

end

end

private

def comment\_params

params.required(:comment).permit(:content)

end

end

**model:**

**user.rb**

class User < ApplicationRecord

has\_secure\_password

validates :email, presence: true

has\_many :posts, dependent: :destroy

has\_many :comments, dependent: :destroy

end

**post.rb**

class Post < ApplicationRecord

belongs\_to :user

has\_many :comments, dependent: :destroy

end

**comment.rb**

class Comment < ApplicationRecord

belongs\_to :user

belongs\_to :post

end

**schema**

ActiveRecord::Schema.define(version: 2021\_12\_17\_083601) do

create\_table "comments", force: :cascade do |t|

t.text "content"

t.datetime "created\_at", precision: 6, null: false

t.datetime "updated\_at", precision: 6, null: false

t.integer "post\_id"

t.string "user\_id"

end

create\_table "posts", force: :cascade do |t|

t.string "content"

t.datetime "created\_at", precision: 6, null: false

t.datetime "updated\_at", precision: 6, null: false

t.string "user\_id"

t.string "topic"

t.string "title"

end

create\_table "users", force: :cascade do |t|

t.string "username"

t.string "email"

t.string "password\_digest"

t.datetime "created\_at", precision: 6, null: false

t.datetime "updated\_at", precision: 6, null: false

t.index ["email"], name: "index\_users\_on\_email", unique: true

end

end

**views:**

**application.html.erb**

<!DOCTYPE html>

<html>

<head>

<title>Commu</title>

<meta name="viewport" content="width=device-width,initial-scale=1">

<%= csrf\_meta\_tags %>

<%= csp\_meta\_tag %>

<%= stylesheet\_link\_tag 'https://stackpath.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css'%>

<%= stylesheet\_link\_tag 'application', media: 'all', 'data-turbolinks-track': 'reload' %>

<%= javascript\_pack\_tag 'application', 'data-turbolinks-track': 'reload' %>

</head>

<body>

<!--Navbar-->

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-md">

<%= link\_to 'Commu', root\_path, class: "navbar-brand" %>

</div>

<div class="nav navbar-right">

<ul class="nav navbar-nav navbar-right">

<%if !current\_user %>

<li><button type="button" class="btn"><%= link\_to t('.log-in'), login\_path %></button></li>

<li><button type="button" class="btn"><%= link\_to t('.sign-up'), new\_user\_path %></button></li>

<%else%>

<li><button type="button" class="btn"><%= link\_to t('.create-post'), new\_post\_path %></button></li>

<li><button type="button" class="btn"><%= link\_to t('.profile'), user\_path(current\_user) %></button></li>

<li><button type="button" class="btn"><%= link\_to t('.log-out'), logout\_path %></button></li>

<%end%>

<li><%= link\_to 'English', locale: "en" %></li> |

<li><%= link\_to 'Русский', locale: "ru" %></li>

</ul>

</div>

</nav>

<!--End of Navbar-->

<%= yield %>

</body>

</html>

**users:**

**new.html.erb**

<div id="login-card" class="container align-self-center">

<%= form\_with(model: @user) do |form| %>

<fieldset>

<legend>Sign up</legend>

<p id="notice"><%= notice %></p>

<div class="field">

<div class="mb-3 row">

<%= form.label :username, class: "col-sm-2 col-form-label" %>

<div class="col-sm-8">

<%= form.text\_field :username, class: "form-control" %>

</div>

</div>

<div class="mb-3 row">

<%= form.label :email, class: "col-sm-2 col-form-label" %>

<div class="col-sm-8">

<%= form.text\_field :email, class: "form-control" %>

</div>

</div>

<div class="mb-3 row">

<%= form.label :password, class: "col-sm-2 col-form-label" %>

<div class="col-sm-8">

<%= form.password\_field :password, class: "form-control" %>

</div>

</div>

<div class="mb-3 row">

<%= form.label :password\_confirmation, class: "col-sm-2 col-form-label" %>

<div class="col-sm-8">

<%= form.password\_field :password\_confirmation, class: "form-control" %>

</div>

</div>

<div class="group-btn actions">

<%= submit\_tag "Sign up", class: "btn btn-form" %>

<button type="button" class="btn btn-form"><%= link\_to 'Login', login\_path %></button>

</div>

</fieldset>

<% end %>

</div>

**sessions:**

**new.html.erb**

<div id="login-card" class="container align-self-center">

<%= form\_tag '/login' do %>

<fieldset>

<legend>Login</legend>

<p id="notice"><%= notice %></p>

<div class="mb-3 row">

<%= label\_tag :email, 'Email', class: "col-sm-2 col-form-label" %>

<div class="col-sm-8">

<%= text\_field\_tag :email, params[:email], class: "form-control" %>

</div>

</div>

<div class="mb-3 row">

<%= label\_tag :password, 'Password', class: "col-sm-2 col-form-label" %>

<div class="col-sm-8">

<%= password\_field\_tag :password, params[:password], class: "form-control" %>

</div>

</div>

<div class="group-btn">

<%= submit\_tag "Login", class: "btn btn-form" %>

<button type="button" class="btn btn-form"><%= link\_to 'Sign up', new\_user\_path %></button>

</div>

</fieldset>

<% end %>

</div>

**posts:**

**index.html.erb**

<% content\_for :page\_title, "Index" %>

<div class="section">

<div class="container">

<% @posts.each do |post| %>

<div class="card-post">

<table>

<tr>

<td colspan="1" class="thumb">

<div>

<%=image\_tag "#{post.topic}.jpg", class: "thumbnail"%>

</div>

</td>

<td colspan="5">

<div class="post-content">

<div>

<%=image\_tag "user.png", class:"avatar for-user"%>

<p class="post-user for-user"><%= "- #{post.user.username}"%> </p>

</div>

<p class="post-title"><%= "#{post.title}"%></p>

</div>

</td>

<td>

<div class="side-btn">

<div class="side-btn">

<%= link\_to t('.read'), post\_path(locale, post), class: "side-btn-link" %> |

<%= link\_to t('.edit'), edit\_post\_path(locale, post), class: "side-btn-link" %> |

<%= link\_to t('.delete'), post, method: :delete, class: "side-btn-link", data: { confirm: 'Are you sure?' } %>

</div>

</div>

</td>

<tr>

</table>

</div>

<% end %>

</div>

</div>

**new.html.erb**

<!--new-->

<div class="show-post-card">

<div class="section">

<%= form\_with(model: @post) do |form| %>

<fieldset>

<legend>Create new post</legend>

<div class="mb-3 row">

<%= form.label :title, class: "col-sm-2 col-form-label" %>

<div class="col-sm-8">

<%= form.text\_field :title, class: "form-control" %>

</div>

</div>

<div class="mb-3 row">

<%= form.label :content, class: "col-sm-2 col-form-label" %>

<div class="col-sm-8">

<%= form.text\_area :content, rows: 17, class: "form-control text-area-post" %>

</div>

</div>

<div class="mb-3 row">

<div class="col-sm-2 col-form-label"><strong>Topic</strong></div>

<div class="col-sm-8">

<%= form.radio\_button :topic, 'travel' %>

<%= form.label :topic, 'Travel', :value => 'travel' %>

<%= form.radio\_button :topic, 'food' %>

<%= form.label :topic, 'Food', :value => 'food' %>

<%= form.radio\_button :topic, 'pet' %>

<%= form.label :topic, 'Pet', :value => 'pet' %>

<%= form.radio\_button :topic, 'work' %>

<%= form.label :topic, 'Work', :value => 'work' %>

<%= form.radio\_button :topic, 'flower' %>

<%= form.label :topic, 'Flower', :value => 'flower' %>

<%= form.radio\_button :topic, 'friend' %>

<%= form.label :topic, 'Friend', :value => 'friend' %>

<%= form.radio\_button :topic, 'love' %>

<%= form.label :topic, 'Love', :value => 'love' %>

</div>

</div>

<div class="actions group-btn">

<%= form.submit "Post", class: "btn btn-form" %>

</div>

</fieldset>

<% end %>

</div>

</div>

**show.html.erb**

<div class="show-post-card">

<h2><%= @post.title %></h2>

<%=image\_tag "user.png", class:"avatar for-user"%>

<h5 class="for-user"><%= " - #{@post.user.username}"%></h5>

<div class="side-btn">

<%= link\_to t('.edit'), edit\_post\_path(@post), class: "side-btn-link" %> |

<%= link\_to t('.back'), root\_path, class: "side-btn-link"%>

</div>

<hr class="solid">

<p><%= @post.content %></p>

</div>

<div class="show-post-card">

<section class="section comments">

<div class="container">

<h2 class="count-cmt"><%= @post.comments.count %> <%=t('.comments') %></h2>

<%= render @post.comments%>

<div class="comment-form">

<hr />

<h3 class="cmt-label"><%=t('.write-your-cmt')%></h3>

<%= render 'comments/new' %>

</div>

</div>

</section>

</div>

**comments:**

**\_comment.html.erb**

<div class="box">

<article class="media">

<div class="media-content">

<div class="content">

<p>

<div>

<%=image\_tag "user.png", class:"avatar-cmt avatar for-user"%>

<strong><%= comment.user.username %></strong>

</div>

<div class="cmt-content"><%= comment.content %></div>

</p>

</div>

</div>

<div class="side-btn" style="margin-right:10px">

<%= link\_to t('.delete'), [comment.post, comment],

method: :delete, class: "side-btn-link", data: { confirm: 'Are you sure?' } %>

</div>

</article>

</div>

**\_new.html.erb**

<div class="section">

<%= form\_for [@post, Comment.new] do |form| %>

<table>

<td style="vertical-align:middle">

<div class="field cmt-items">

<div class="control">

<%=image\_tag "user.png", class:"avatar-cmt avatar for-user"%>

<%= form.text\_area :content%>

</div>

</div>

</td>

<td>

<div class="actions cmt-items">

<%= form.submit t('.send'), class: "btn btn-form" %>

</div>

</td>

</table>

<% end %>

</div>

**test:**

**users\_controller\_test.rb**

require "test\_helper"

class UsersControllerTest < ActionDispatch::IntegrationTest

Rails.application.configure do

routes.default\_url\_options[:locale] = :en

end

setup do

@user = users(:one)

end

test "should get index" do

get users\_url

assert\_redirected\_to login\_url

end

test "should get new" do

get new\_user\_url

assert\_response :success

end

test "should create user" do

assert\_difference('User.count') do

post users\_url, params: { user: { email: 'email', password: 'secret', password\_confirmation: 'secret', username: @user.username } }

end

assert\_redirected\_to root\_url

end

test "should show user" do

get user\_url(@user)

assert\_redirected\_to login\_url

end

test "should get edit" do

get edit\_user\_url(@user)

assert\_redirected\_to login\_url

end

test "should update user" do

patch user\_url(@user), params: { user: { email: @user.email, password: 'secret', password\_confirmation: 'secret', username: @user.username } }

assert\_redirected\_to login\_url

end

test "should destroy user" do

delete user\_url(@user)

assert\_redirected\_to login\_url

end

end

**posts\_controller\_test.rb**

require "test\_helper"

class PostsControllerTest < ActionDispatch::IntegrationTest

Rails.application.configure do

routes.default\_url\_options[:locale] = :en

end

setup do

@user = users(:one)

@post = posts(:two)

end

test "should get new" do

get new\_post\_url

assert\_redirected\_to login\_url

end

test "should create post" do

post posts\_url, params: { post: { title: 'title', content: 'content', topic: 'family', user\_id: 'MyString'} }

assert\_redirected\_to login\_url

end

test "should show post" do

get post\_url(@post.id)

assert\_redirected\_to login\_url

end

test "should get edit" do

get edit\_post\_path(@post.id)

assert\_redirected\_to login\_url

end

test "should update user" do

patch post\_url(@post.id), params: { post: { title: 'title', content: 'content', topic: 'family', user\_id: 'MyString'} }

assert\_redirected\_to login\_url

end

test "should destroy post" do

delete post\_url(@post.id)

assert\_redirected\_to login\_url

end

end

**sessions\_controller\_test.rb**

require "test\_helper"

class SessionsControllerTest < ActionDispatch::IntegrationTest

Rails.application.configure do

routes.default\_url\_options[:locale] = :en

end

test "should get new" do

get new\_session\_url

assert\_response :success

end

test "should get create" do

post sessions\_path

assert\_redirected\_to login\_url

end

test "should get destroy" do

get logout\_url

assert\_redirected\_to login\_url

end

end

**user\_test.rb**

require "test\_helper"

class UserTest < ActiveSupport::TestCase

setup do

@user = User.create(username: 'anhtu', email: 'tu@mail.ru', password: 'anhtu', password\_confirmation: 'anhtu')

end

test 'valid user' do

assert @user.valid?

end

test 'invalid without email' do

@user.email = nil

refute @user.valid?

assert\_not\_nil @user.errors[:email]

end

test 'invalid without password' do

@user.password = nil

refute @user.valid?

assert\_not\_nil @user.errors[:password]

end

end