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 лабораторной работе № 1 1

Название: Добавление модели. ORM. Разработка БД, подключение, хранение и поиск данных.

Дисциплина: Языки интернет программирования

Студент	ИУ6-53Б		Т.Р. Сапарбаев
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			Р.С. Самарев
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2020

Задание

Модифицировать код ЛР 8 таким образом, чтобы запросы, которые были ранее выполнены, сохранялись в БД и при следующем запросе не требовали повтора вычислений.

- Сформировать модель в соответствии с потребностями хранения данных. Входные параметры являются ключами, по которым извлекается результат.
- Выполнить создание БД и миграцию соответствующими запросами rake.
- Написать тест на добавление и поиск данных с помощью модели. Проверить выполнение теста.
- Модифицировать код приложения таким образом, чтобы результат вычислений преобразовывался в строковый или бинарный формат (на выбор: json, xml, и пр.). Проверить через отладочную печать в консоль, что преобразование выполняется корректно.
- Вставить код для сохранения данных в БД и запрос на поиск предыдущего результата вычислений.
- Добавить действие в контроллер, позволяющее определить, что хранится в БД через сериализацию в XML.
- Проверить, что при выполнении запроса, данные добавляются в БД.
- При помощи консоли сообщений Webrick определить, производится ли поиск результата предыдущего запроса в БД и не повторяются ли одни и те же вычисления.
- Модифицировать модель таким образом, чтобы добавление записей с одинаковыми параметрами было невозможно.
- Реализовать тест модели, проверяющий невозможность повторного добавления одних и тех же результатов вычислений.
- Реализовать функциональный тест, проверяющий, что результаты вычислений различны при различных входных параметрах.
- Проверить маршруты приложения с помощью rake routes и убрать лишние. Обеспечить доступ при обращении по адресу /.

Результат приложить в виде двух файлов:

- архив, содержащий RoR-приложение;
- pdf-отчет, в котором должны присутствовать фрагменты добавленного кода.

Отчет должен содержать:

- ФИО, номер группы и текст задания;
- перечень и содержимое файлов, которые были изменены в процессе создания приложения.
- XML-распечатку содержимого БД (ограничить несколькими записями так, чтобы результат поместился на 1-2 страницах).
- Примры SQL-кода добавления и извлечения данных из БД из отладочной консоли сервера Webrick.

Маршруты (routes.rb):

```ruby
Rails.application.routes.draw do
  root 'home#input'
  get 'home/input' => 'home#input', :as => 'input'
  get 'home/output' => 'home#output', :as => 'output'
  get 'home/serialize' => 'home#serialize_db', :as => 'serialize'
  get 'home/last_record' => 'home#last_rec', :as => 'lastrec'
end
```

Контроллер (home_controller.rb):

```ruby
# frozen_string_literal: true

# Home Controller Class
class HomeController < ApplicationController
  before_action :check_params, only: %(output)

  def input; end

  def output
    @result = Result.create_or_get_by_request(@request)
  end

  def serialize_db
    render xml: Result.all.to_xml
  end

  def last_rec
    render xml: (res = Re-
sult.last).nil? ? ['База данных пуста'].to_xml : res.to_xml
  end

  protected

  def check_params
    @request = Integer(params[:txt], exception: false)
    @error = 'Введено не число' if @request.nil?
    render :output unless @error.nil?
  end
end
```

Контроллер для доступа к БД приложения (result.rb):

```ruby
# frozen_string_literal: true

# Class Result
class Result < ApplicationRecord
  validates :request, uniqueness: true

  class << self
    def compute(request)
      (request + 1).times.filter do |i|
        square = (i**2).to_s
        square == square.reverse
      end
    end

    def create_or_get_by_request(request)
      res = find_by(request: data)
      return res unless res.nil?

      create(request: request, response: Result.compute(request))
    end
  end

  def response
    ActiveSupport::JSON.decode(super)
  end

  def response=(value)
    super(ActiveSupport::JSON.encode(value))
  end
end
```

БД приложения в XML формате:

```
<results type="array">
  <result>
    <id type="integer">1</id>
    <request>200</request>
    <response type="array">
      <response type="integer">0</response>
      <response type="integer">1</response>
      <response type="integer">2</response>
      <response type="integer">3</response>
      <response type="integer">11</response>
      <response type="integer">22</response>
      <response type="integer">26</response>
      <response type="integer">101</response>
      <response type="integer">111</response>
      <response type="integer">121</response>
    </response>
    <created-at type="dateTime">2021-01-25T23:37:03Z</created-at>
    <updated-at type="dateTime">2021-01-25T23:37:03Z</updated-at>
  </result>
  <result>
    <id type="integer">2</id>
    <request>100</request>
    <response type="array">
      <response type="integer">0</response>
      <response type="integer">1</response>
      <response type="integer">2</response>
      <response type="integer">3</response>
      <response type="integer">11</response>
      <response type="integer">22</response>
      <response type="integer">26</response>
    </response>
    <created-at type="dateTime">2021-01-25T23:58:36Z</created-at>
    <updated-at type="dateTime">2021-01-25T23:58:36Z</updated-at>
  </result>
  <result>
    <id type="integer">3</id>
    <request>10</request>
    <response type="array">
      <response type="integer">0</response>
      <response type="integer">1</response>
      <response type="integer">2</response>
      <response type="integer">3</response>
    </response>
    <created-at type="dateTime">2021-01-27T11:54:33Z</created-at>
    <updated-at type="dateTime">2021-01-27T11:54:33Z</updated-at>
  </result>
</results>
```

Тест модели (result_test.rb):

```ruby
require 'test_helper'

class ResultTest < ActiveSupport::TestCase
  test 'add new record to db' do
    req = '1'
    res = [0,1].to_json
    unless exs = Result.find_by(request: req).nil?
      Result.find_by(request: req).destroy
    end
    Result.create(request: req, response: res).save
    assert_not_nil Result.find_by(request: req).nil?
    Result.find_by(request: req).destroy if exs
  end
```

```
  test 'duplicate addition' do
    req = '1'
    res = [0,1].to_json
    unless exs = Result.find_by(request: req).nil?
      Result.find_by(request: req).destroy
    end
    Result.create(request: req, response: res).save
    assert_not Result.create(request: req, response: '{}').valid?
    Result.find_by(request: req).destroy if exs
  end
end
```

Пример работы приложения:

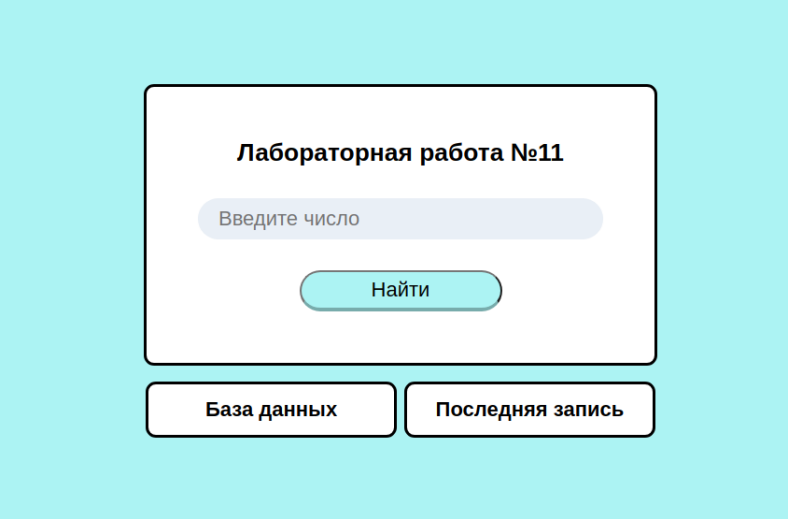

Рисунок 1 – Главная страница приложения

Содержимое БД:

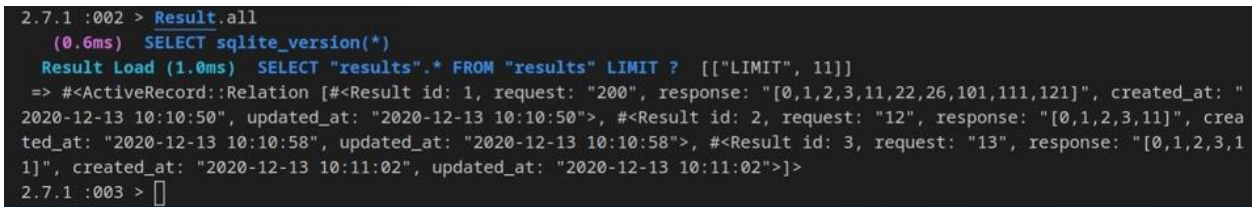

Рисунок 2 – Просмотр содержимого БД через команду «rails c»

Тесты приложения:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR11$ rails test
Running via Spring preloader in process 29099
Run options: --seed 51212

# Running:

..

Finished in 0.202650s, 9.8692 runs/s, 9.8692 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
```

Рисунок 3 – Тесты приложения

Отчеты Rubocop:

```
saparbi@saparbi-TM1703:~/Desktop/YAIP/LR11/app/models$ rubocop
warning: parser/current is loading parser/ruby27, which recognizes
warning: 2.7.2-compliant syntax, but you are running 2.7.1.
warning: please see https://github.com/whitequark/parser#compatibility-with-ruby-mri.
Inspecting 2 files
..

2 files inspected, no offenses detected

saparbi@saparbi-TM1703:~/Desktop/YAIP/LR11/app/controllers$ rubocop
warning: parser/current is loading parser/ruby27, which recognizes
warning: 2.7.2-compliant syntax, but you are running 2.7.1.
warning: please see https://github.com/whitequark/parser#compatibility-with-ruby-mri.
Inspecting 2 files
..

2 files inspected, no offenses detected
```

Рисунок 4 – Отчёт Rubocop

Вывод: в ходе выполнения лабораторной работы был доработан код ЛР8 таким образом, чтобы все запросы сохранялись в БД. Приложение было протестировано и проверено на соответствие стилю программой Rubocop.