|  |  |
| --- | --- |
| **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 Лабораторной работе № 8** |

**Название: Ruby on rails**

**Дисциплина:** Языки Интернет-программирования



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-32Б |  |  | Плютто А. П. |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Гаврилова Е. Ю. |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Задание**

Разработать веб-приложение, имеющее HTML-страницу с формой ввода данных и HTML-страницу для представления результатов. Результат

расчёта должен быть представлен в форме таблицы, оформленной с помощью элемента table или отдельными ячейками div и имеющей не менее

двух колонок. Если по условию задания результат может быть представлен только в виде одной строки таблицы, необходимо реализовать вывод

промежуточных результатов расчёта в качестве дополнительных строк.

В этом случае первой колонкой таблицы будет порядковый номер итерации.

Под вводом с клавиатуры в тексте заданий следует понимать ввод в

поле ввода данных формы на HTML-странице.

Текст задания:

Два числа называются взаимно дружественными, если каждое из них

равно сумме всех делителей другого, кроме самого этого числа. Например: 220 и 284, так как

Делители 220 ⇒ 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110 сумма = 284

Делители 284 ⇒ 1, 2, 4, 71, 142 сумма = 220

Написать программу, определяющую дружественные числа, лежащие в

диапазоне от 1 до n (n вводится с клавиатуры). Вывести на печать все

найденные числа. При программировании использовать функции.

**Решение**

Для начала создадим каркас приложения, для этого запустим такую команду:

1. rails new '/Users/pluttan/Desktop/\_Sem3/BMSTU-3/IP/labs/lab8/trysrever'

После этого поменяем виды на свои, добавим свои контроллеры, пропишем зависимости в routes и добавим несколько тестов.

Index.html.erb

1. <!DOCTYPE html>

2. <html **lang**="en">

3.

4. <head>

5. <meta **charset**="UTF-8">

6. <meta **name**="viewport" **content**="width=device-width, initial-scale=1.0">

7. <title>форма</title>

8. <style>

9. /\* Стилизация формы и элементов \*/

10. body {

11. font-family: Arial, sans-serif;

12. background-color: #f0f0f0;

13. }

14.

15. .container {

16. max-width: 400px;

17. margin: 0 auto;

18. padding: 20px;

19. background-color: #fff;

20. border-radius: 8px;

21. box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

22. }

23.

24. h2 {

25. text-align: center;

26. color: #333;

27. }

28.

29. .form-group {

30. margin-bottom: 20px;

31. }

32.

33. label {

34. display: block;

35. font-weight: bold;

36. margin-bottom: 5px;

37. }

38.

39. input[type="text"] {

40. width: calc(100% - 20px);

41. padding: 10px;

42. border: 1px solid #ccc;

43. border-radius: 4px;

44. font-size: 16px;

45. }

46.

47. button {

48. display: block;

49. width: 100%;

50. padding: 10px;

51. background-color: #007BFF;

52. color: #fff;

53. border: none;

54. border-radius: 4px;

55. font-size: 16px;

56. cursor: pointer;

57. }

58.

59. button:hover {

60. background-color: #0056b3;

61. }

62. </style>

63. </head>

64.

65. <body>

66. <div **class**="container">

67. <h2>Дружественные числа</h2>

68. <form **action**="http://localhost:3000/result" **method**="get">

69. <div **class**="form-group">

70. <label **for**="inputText">Введите n:</label>

71. <input **type**="text" **id**="inputText" **name**="inputText" **placeholder**="n сюда)" **required**>

72. </div>

73. <div **class**="form-group">

74. <button **type**="submit">Произвести поиск дружественных чисел от 1 до n</button>

75. </div>

76. </form>

77. </div>

78. </body>

79.

80. </html>

81.

Result.html.erb

1. <!DOCTYPE html>

2. <html **lang**="en">

3.

4. <head>

5. <meta **charset**="UTF-8">

6. <meta **name**="viewport" **content**="width=device-width, initial-scale=1.0">

7. <title>Таблица</title>

8. <style>

9. /\* Стилизация таблицы \*/

10. body {

11. font-family: Arial, sans-serif;

12. background-color: #f0f0f0;

13. }

14.

15. .container {

16. max-width: 800px;

17. margin: 0 auto;

18. padding: 20px;

19. background-color: #fff;

20. border-radius: 8px;

21. box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

22. }

23.

24. h2 {

25. text-align: center;

26. color: #333;

27. }

28.

29. table {

30. width: 100%;

31. border-collapse: collapse;

32. margin-top: 20px;

33. }

34.

35. th,

36. td {

37. padding: 10px;

38. text-align: center;

39. }

40.

41. th {

42. background-color: #007BFF;

43. color: #fff;

44. }

45. </style>

46. </head>

47.

48. <body>

49. <div **class**="container">

50. <h2>Полученные значения:</h2>

51. <table>

52. <thead>

53. <tr>

54. <th **style**="border-radius: 10px 0 0 10px;">ID</th>

55. <th>Дружественное число 1</th>

56. <th **style**="border-radius: 0 10px 10px 0;">Дружественное число 2</th>

57. </tr>

58. </thead>

59. <tbody **id**="tbody">

60. <%= @message %>

61. </tbody>

62. </table>

63. </div>

64. </body>

65.

66. </html>

67.

Page2\_controller.rb

1. class Page2Controller < ApplicationController

2. def result

3. @n = 0

4. params.each do |key, value|

5. Rails.logger.warn "Param #{key}: #{value}"

6. if key == "inputText" then @n = value.to\_i end

7. end

8. @message = find\_and\_check\_for\_range(@n)

9. @result = @message[1]

10. p @result

11. @message = @message[0]

12. @result

13. end

14.

15. def find\_divisors\_and\_check(number, n)

16. divisors = (1..(number / 2 + 1)).select { |divisor| number % divisor == 0 }

17. sum\_of\_divisors = divisors.sum

18. second\_divisors = (1..(sum\_of\_divisors / 2 + 1)).select { |divisor| sum\_of\_divisors % divisor == 0 }

19. if second\_divisors.sum == number && number < sum\_of\_divisors && n >= sum\_of\_divisors

20. return [number, sum\_of\_divisors]

21. else

22. return false

23. end

24. end

25.

26. def find\_and\_check\_for\_range(n)

27. id = 0

28. @message = ""

29. @result = []

30. (1..n).each do |number|

31. result = find\_divisors\_and\_check(number, n)

32. id += 1 if result

33. @result << result if result

34. @message += "<tr><td>#{id}</td> <td>#{result[0]}</td> <td>#{result[1]}</td></tr>" if result

35. end

36. [@message.html\_safe, @result]

37. end

38. end

39.

Routes.rb

1. Rails.application.routes.draw do

2. root "page1#index"

3. get "/result", to: "page2#result"

4. end

5.

Friends\_controller\_test.rb



1. require "/Users/pluttan/Desktop/\_Sem3/BMSTU-3/IP/labs/lab8/trysrever/test/test\_helper.rb"

2.

3. class TwinsControllerTest < ActionDispatch::IntegrationTest

4. test "should get input" do

5. get root\_url

6. assert\_response :success

7. end

8.

9. test "should get view" do

10. get result\_url

11. assert\_response :success

12. end

13.

14. test "get inputText = 10000" do

15. get result\_url, params: { inputText: 10000 }

16. puts assigns[:result]

17. target = [[220, 284],

18. [1184, 1210],

19. [2620, 2924],

20. [5020, 5564],

21. [6232, 6368]]

22.

23. assert\_equal assigns[:result], target

24. end

25.

26. test "get inputText = 1000" do

27. get result\_url, params: { inputText: 1000 }

28. puts assigns[:result]

29. target = [[220, 284]]

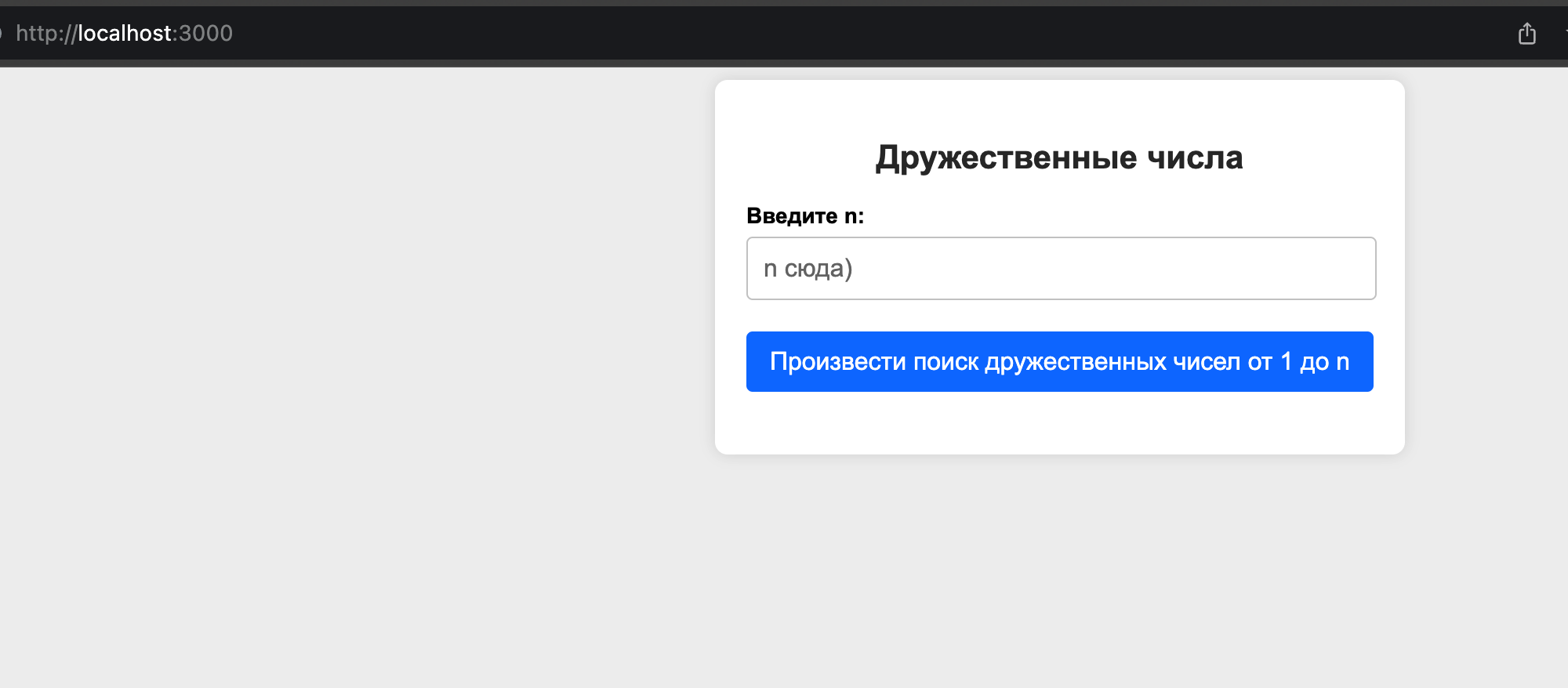
30.

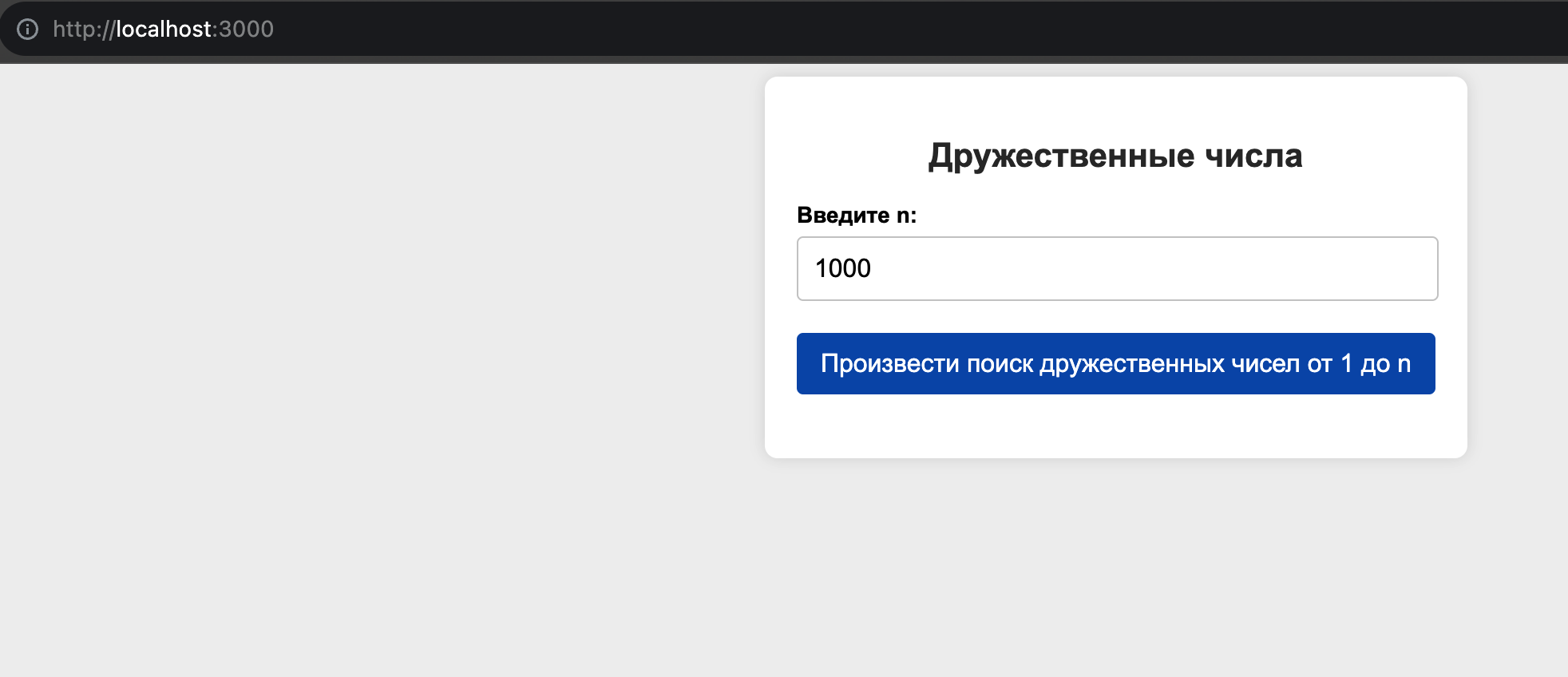
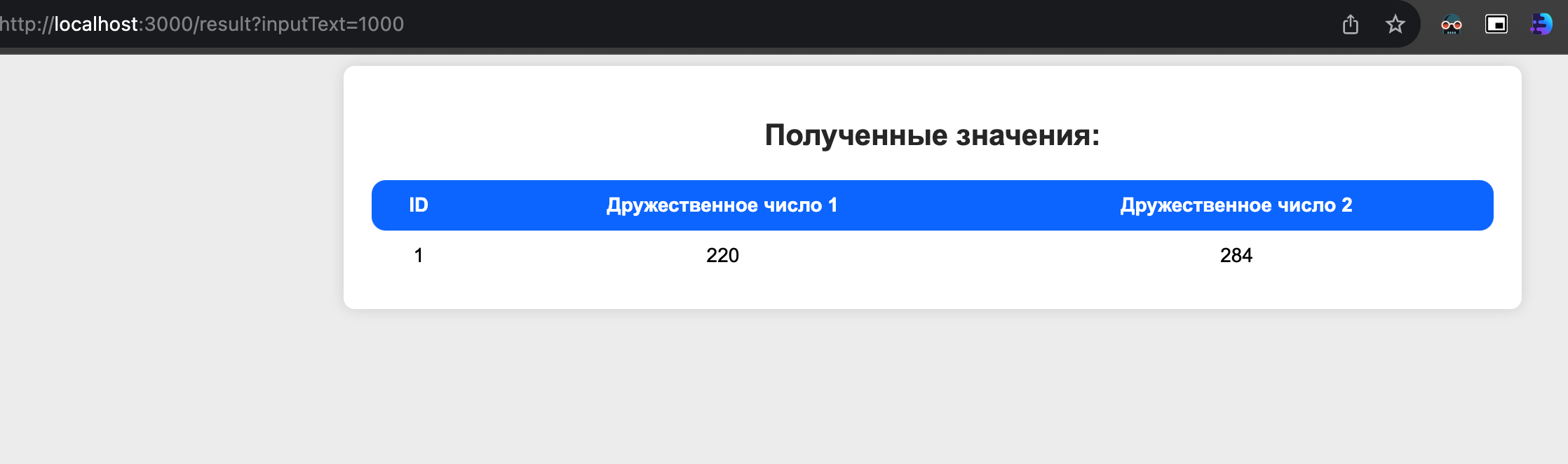
31. assert\_equal assigns[:result], target

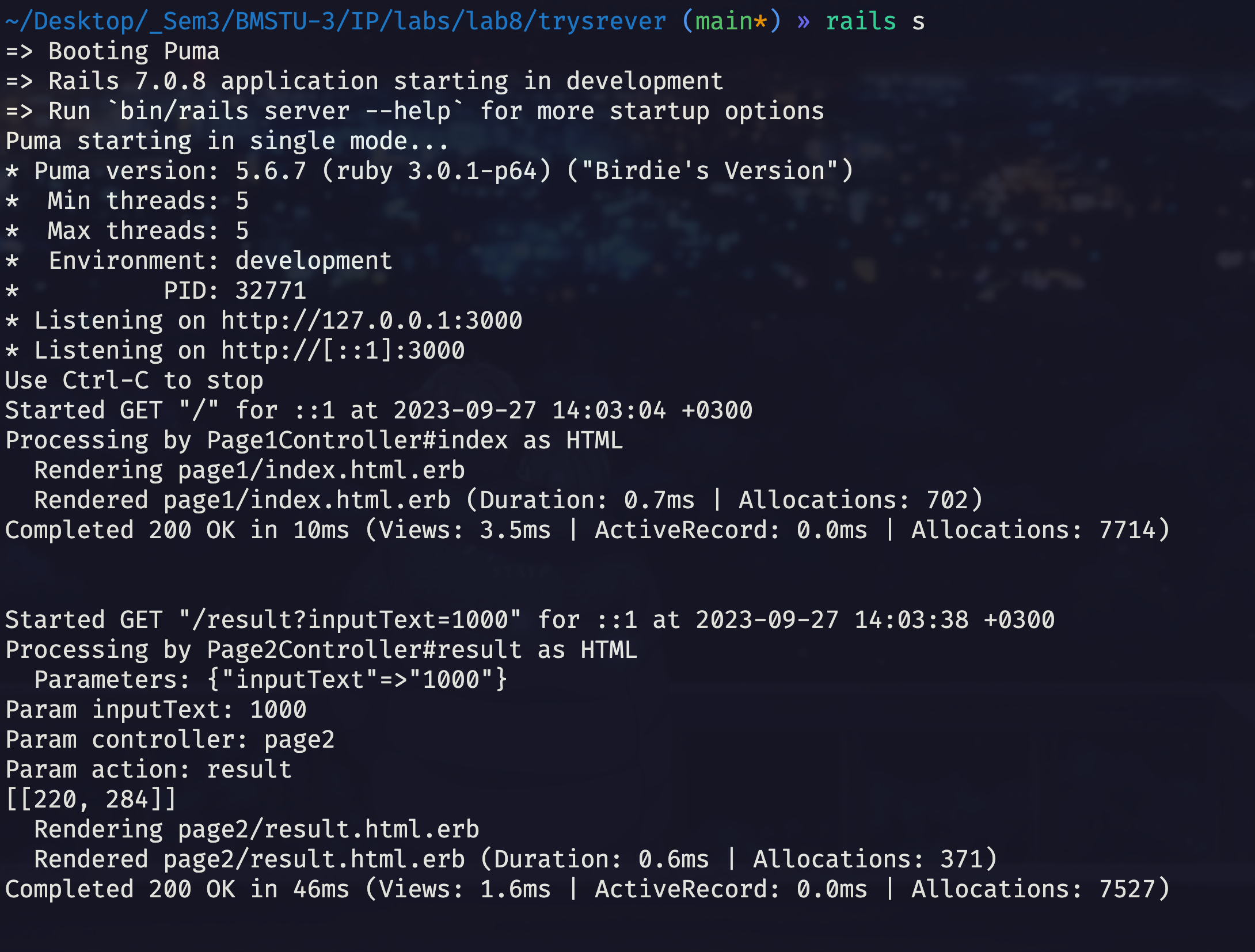
32. end

33. end

34.





Для проверки форматирования был установлен автоформатировщик rufo.