



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 \_\_\_\_\_ «Теоретическая информатика и компьютерные технологии»

## **Рубежный контроль №1** **по курсу «Компьютерные сети»**

**«Создание посадочной страницы с ссылками на разные  
поддомены»**

Студент группы ИУ9-32Б Лавров Р. Д.

Преподаватель Посевин Д. П.

*Москва 2025*

# 1 Задание

Приобретение опыта настройки DNS-сервера и создание посадочной страницы с ссылками на разные поддомены.

## 2 Результаты

Листинг 1: main.go

```
1 package main
2
3 import (
4     "fmt"
5     "html/template"
6     "net/http"
7
8     log "github.com/mgutz/logxi/v1"
9 )
10
11 const mainPage = `
12 <!doctype html>
13 <html lang="ru">
14 <head>
15     <meta charset="utf-8">
16     <titleГлавная></title>
17 </head>
18 <body>
19     <h1Главная> страница</h1>
20
21     <a href="http://1.veterok.2kurs.iu9.org.ru:4001">
22         <button style="height:100px;width:200px;font-size:30px">1</button>
23     </a>
24
25     <a href="http://2.veterok.2kurs.iu9.org.ru:3000">
26         <button style="height:100px;width:200px;font-size:30px">2</button>
27     </a>
28
29     <a href="http://3.veterok.2kurs.iu9.org.ru:2003">
30         <button style="height:100px;width:200px;font-size:30px">3</button>
31     </a>
32 </body>
33 </html>
34 `
35
```

```

36 var tpl = template.Must(template.New("main").Parse(mainPage))
37
38 func handler(w http.ResponseWriter, r *http.Request) {
39     log.Info("request", "method", r.Method, "path", r.URL.Path)
40     if r.URL.Path != "/" {
41         http.NotFound(w, r)
42         return
43     }
44     w.Header().Set("Content-Type", "text/html; charset=utf-8")
45     _ = tpl.Execute(w, nil)
46 }
47
48 func main() {
49     fmt.Println("server started")
50     http.HandleFunc("/", handler)
51     log.Info("listen", "addr", "185.102.139.161:8455")
52     log.Error("listen failed", "err",
53         http.ListenAndServe("185.102.139.161:8455", nil))
54 }

```

## Листинг 2: s1.go

```

1 package main
2
3 import (
4     "html/template"
5     "net/http"
6
7     log "github.com/mgutz/logxi/v1"
8 )
9
10 const pageOne = `
11 <!doctype html>
12 <html lang="ru">
13 <head>
14     <meta charset="utf-8">
15     <titleСайт> один</title>
16 </head>
17 <body>
18     <pДоменное> имя:
19     <script type="text/javascript">
20         document.write(document.domain);
21     </script>
22     </p>
23 </body>
24 </html>

```

```

25 '
26
27 var tpl = template.Must(template.New("one").Parse(pageOne))
28
29 func handler(w http.ResponseWriter, r *http.Request) {
30     log.Info("request", "method", r.Method, "path", r.URL.Path)
31     if r.URL.Path != "/" {
32         http.NotFound(w, r)
33         return
34     }
35     w.Header().Set("Content-Type", "text/html; charset=utf-8")
36     _ = tpl.Execute(w, nil)
37 }
38
39 func main() {
40     http.HandleFunc("/", handler)
41     log.Info("listen", "addr", "185.102.139.161:4001")
42     log.Error("listen failed", "err",
43         http.ListenAndServe("185.102.139.161:4001", nil))
44 }

```

### Листинг 3: s2.go

```

1 package main
2
3 import (
4     "html/template"
5     "net/http"
6
7     log "github.com/mgutz/logxi/v1"
8 )
9
10 const pageTwo = `
11 <!doctype html>
12 <html lang="ru">
13 <head>
14     <meta charset="utf-8">
15     <titleСайт> два</title>
16 </head>
17 <body>
18     <pДоменное> имя:
19     <script type="text/javascript">
20         document.write(document.domain);
21     </script>
22     </p>
23 </body>

```

```

24 </html>
25 ‘
26
27 var tpl = template.Must(template.New("two").Parse(pageTwo))
28
29 func serve(w http.ResponseWriter, r *http.Request) {
30     log.Info("request", "method", r.Method, "path", r.URL.Path)
31     if r.URL.Path != "/" {
32         http.NotFound(w, r)
33         return
34     }
35     w.Header().Set("Content-Type", "text/html; charset=utf-8")
36     _ = tpl.Execute(w, nil)
37 }
38
39 func main() {
40     http.HandleFunc("/", serve)
41     log.Info("listen", "addr", "185.102.139.161:3000")
42     log.Error("listen failed", "err",
43         http.ListenAndServe("185.102.139.161:3000", nil))
44 }

```

## Листинг 4: s3.go

```

1 package main
2
3 import (
4     "html/template"
5     "net/http"
6
7     log "github.com/mgutz/logxi/v1"
8 )
9
10 const pageThree = `
11 <!doctype html>
12 <html lang="ru">
13 <head>
14     <meta charset="utf-8">
15     <titleСайт> три</title>
16 </head>
17 <body>
18     <pДоменное> имя:
19     <script type="text/javascript">
20         document.write(document.domain);
21     </script>
22     </p>

```

```

23 </body>
24 </html>
25 '
26
27 var tpl = template.Must(template.New("three").Parse(pageThree))
28
29 func root(w http.ResponseWriter, r *http.Request) {
30     log.Info("request", "method", r.Method, "path", r.URL.Path)
31     if r.URL.Path != "/" {
32         http.NotFound(w, r)
33         return
34     }
35     w.Header().Set("Content-Type", "text/html; charset=utf-8")
36     _ = tpl.Execute(w, nil)
37 }
38
39 func main() {
40     http.HandleFunc("/", root)
41     log.Info("listen", "addr", "185.102.139.161:2003")
42     log.Error("listen failed", "err",
43         http.ListenAndServe("185.102.139.161:2003", nil))
44 }

```

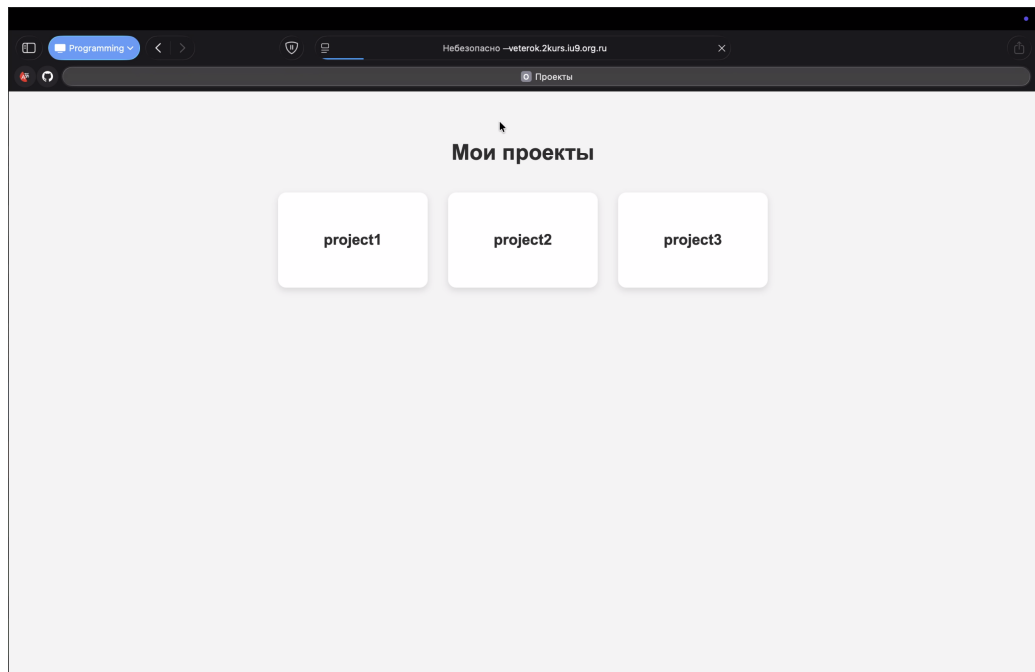


Рис. 1 — Результат работы

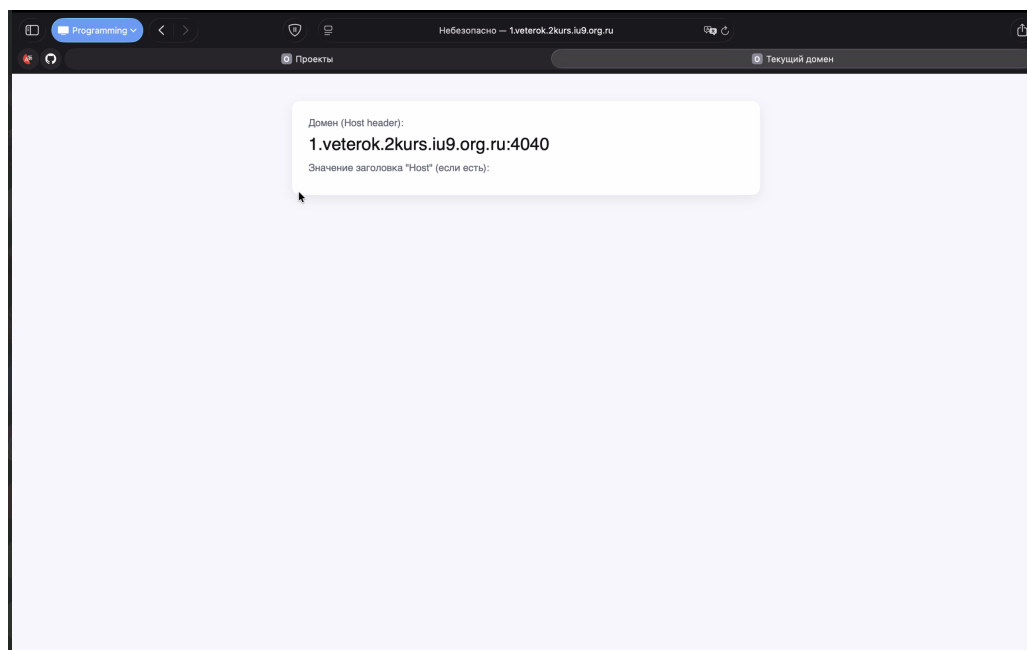


Рис. 2 — Результат работы

### 3 Вывод

Данная работа позволила приобрести опыт настройки DNS-сервера и создания посадочной страницы с ссылками на разные поддомены.