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 № 6

Название: **Основы Back-End разработки на Golang**

Дисциплина: **Языки Интернет-программирования**

Студент ИУ6-31Б _____ О.И.Ельничных
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ _____
(Подпись, дата) (И.О. Фамилия)

Москва, 2024

Цель работы

Изучение основ сетевого взаимодействия и серверной разработки с использованием языка Golang. В рамках данной лабораторной работы предлагается продолжить изучение Golang и познакомиться с набором стандартных библиотек, используемых для организации сетевого взаимодействия и разработки серверных приложений.

Ход работы

Делаем fork репозитория (Рисунок 1).

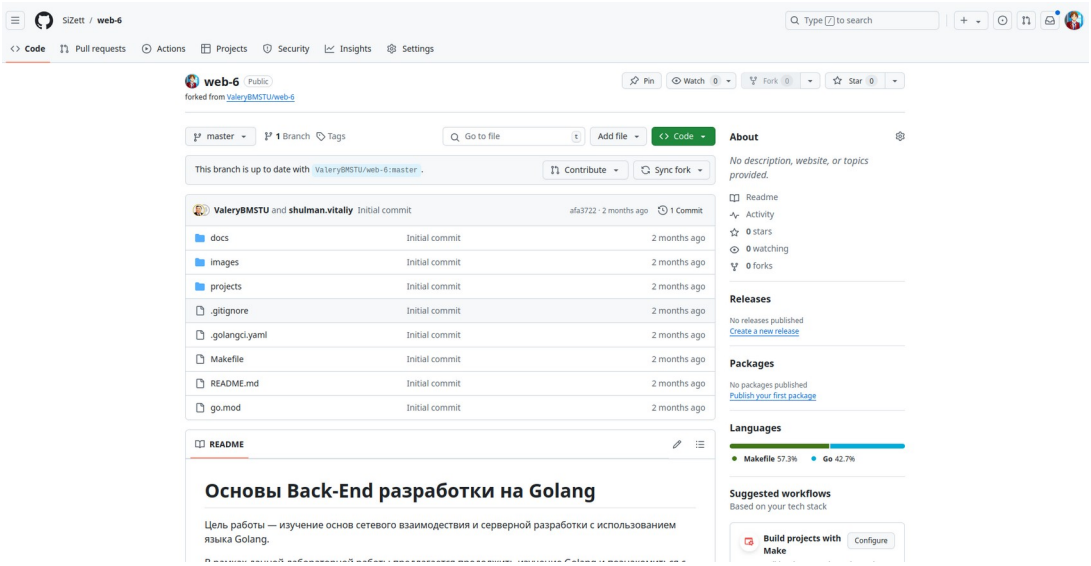

Рисунок 1

Код задания 1_hello:

```
package main
// некоторые импорты нужны для проверки
import (
 "fmt"
 "io"
 "net/http"
 "os"
 "time"
)
```

```
func handler(w http.ResponseWriter, r *http.Request) {
 w.Write([]byte("Hello, web!"))
}

func main() {
    http.HandleFunc("/get", handler)

 // Запускаем веб-сервер на порту 8080
 err := http.ListenAndServe(":8080", nil)
 if err != nil {
  fmt.Println("Ошибка запуска сервера:", err)
 }
}
```

Код задания 2_query:

```
            package main
// некоторые импорты нужны для проверки
import (
  "fmt"
  "io"
  "log"
  "net/http"
  "net/url"
  "os"
  "time"
    "strconv" // вдруг понадобиться вам ;)
)

var count1 int = 0

func handler(w http.ResponseWriter, r *http.Request) {
  if r.Method == "GET" {
        w.WriteHeader(http.StatusOK)
    w.Write([]byte(strconv.Itoa(count1)))
    return
  } else if r.Method == "POST" {
    r.ParseForm()
    s := r.FormValue("count")
    if s == "" {
      w.WriteHeader(http.StatusBadRequest)
      w.Write([]byte("это не число"))

      return
    }
    number, err := strconv.Atoi(s)
    if err != nil {
      w.WriteHeader(http.StatusBadRequest)
      w.Write([]byte("это не число"))

      return
```

```
    }
    count1 += number
    return
  } else {
    w.WriteHeader(http.StatusMethodNotAllowed)
    w.Write([]byte("Метод не поддерживается"))
    return
  }

}

func main() {
  http.HandleFunc("/count", handler)

  err := http.ListenAndServe(":3333", nil)
  if err != nil {
    fmt.Println("Ошибка запуска сервера!")
  }
}
```

Код задания 3_count:

```
package main
// некоторые импорты нужны для проверки
import (
 "fmt"
 "io"
 "net/http" // пакет для поддержки HTTP протокола
 "os"
 "time"
)

func handler(w http.ResponseWriter, r *http.Request) {
    s := r.URL.Query().Get("name")
 w.Write([]byte("Hello," + s + "!"))
}

func main() {
 http.HandleFunc("/api/user", handler)

 err := http.ListenAndServe(":9000", nil)
 if err != nil {
  fmt.Println("Ошибка запуска сервера:", err)
 }
}
```

Заключение

При выполнении заданий лабораторной работы №5 мы познакомились с основами сетевого взаимодействия и серверной разработки с использованием языка Golang и выполнили три задания с разными условиями.