## Project Report #2

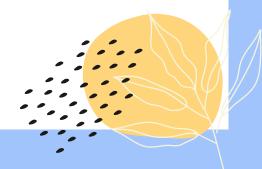
Yerzhanov Beksultan 200103513 Tashim Darkhan 200103208



#### First of all we installed fiber package for usage

```
√ 17 ■■■■■ go.mod 
□

              @@ -0,0 +1,17 @@
         1 + module sdustore
         3 + go 1.18
         5 + require (
                      github.com/andybalholm/brotli v1.0.0 // indirect
                      github.com/gofiber/fiber v1.14.6 // indirect
                      github.com/gofiber/utils v0.0.10 // indirect
                      github.com/gorilla/schema v1.1.0 // indirect
                      github.com/klauspost/compress v1.10.7 // indirect
                      github.com/mattn/go-colorable v0.1.7 // indirect
        11 +
                      github.com/mattn/go-isatty v0.0.12 // indirect
        12 +
                      github.com/valyala/bytebufferpool v1.0.0 // indirect
        13 +
                      github.com/valyala/fasthttp v1.16.0 // indirect
        14 +
                      github.com/valyala/tcplisten v0.0.0-20161114210144-ceec8f93295a // indirect
        15 +
                      golang.org/x/sys v0.0.0-20200602225109-6fdc65e7d980 // indirect
        16 +
        17 + )
```



#### Logic for registration changed

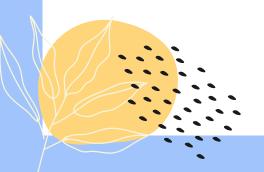
```
8 type User struct {
             name, surname, password, email string
            Username, Surname, Password, Email string
13 var users []User
15 func main() {
- start()
16 + http.HandleFunc("/", home)
17 + http.HandleFunc("/register", register)
           http.HandleFunc("/login", login)
          http.ListenAndServe(":8080", nil)
 - func start() {

    fmt.Println("Choose a action:\n1) Sign up\n2) Log in")

           _, err := fmt.Scanf("%d", &choice)
           if err != nil {
                 log.Fatal(err)
22 th func home(w http.ResponseWriter, r *http.Request) {
23 + t, _ := template.ParseFiles("home.html")
24 +
          t.Execute(w, nil)
25 + }
26
            if choice -- 1 {
                   register()
           } else if choice == 2 {
27 + func register(w http.ResponseWriter, r *http.Request) {
28 + if r.Method == "POST" {
```

```
return rand.Inth(1000000)
68
            - func authorize() {
                      var loggedUser User
71
                      var inputEmail string
72
                     var pswrd string
73
                      fmt.Printf("Enter your email: ")
                      fmt.Scan(&inputEmail)
                     for i := 0; i < len(users); i++ {
                              if users[i].email == inputEmail {
                                     fmt.Printf("Enter your password: ")
                                     fmt.Scan(&pswrd)
                                     if pswrd == users[i].password {
                                             loggedUser = users[i]
83
                                     } else {
                                             fmt.Println()
                                             fmt.Println("Password is incorrect. Try again.")
                                             fmt.Println()
                                             start()
        40 + func login(w http.ResponseWriter, r *http.Request) {
                     if r.Method == "POST" {
        42 +
                              username := r.FormValue("username")
        43 +
                              password := r.FormValue("password")
                              for _, user := range users {
                                     if user.Username == username && user.Password == password {
                                             http.Redirect(w, r, "/welcome", http.StatusSeeOther)
```

# Changed Main.go file and now it can be launched in localhost







```
- var users []User
       16 func main() {
15
                     http.HandleFunc("/", home)
                     http.HandleFunc("/register", register)
                     http.HandleFunc("/login", login)
18
                     http.ListenAndServe(":8080", nil)
       17 +
                     db, err := sql.Open("mysql", "root:password@tcp(127.0.0.1:3306)/maindb")
                     if err != nil {
                             panic(err.Error())
       20 +
       21 +
                     defer db.Close()
       22 +
       23 +
                    router := mux.NewRouter()
       24 +
                     router.HandleFunc("/", home)
       25 +
                     router.HandleFunc("/register", register).Methods("POST")
       26 +
                     router.HandleFunc("/register", showRegisterForm).Methods("GET")
       27 +
                     router.HandleFunc("/login", login).Methods("POST")
                     router.HandleFunc("/login", showLoginForm).Methods("GET")
                     router.HandleFunc("/welcome", welcome)
       29 +
       30 +
                     http.ListenAndServe(":8080", router)
20
       31
21
       32
             func home(w http.ResponseWriter, r *http.Request) {
                     t, _ := template.ParseFiles("home.html")
                     t, := template.ParseFiles("templates/home.html")
       34 +
                     t.Execute(w, nil)
       35 +
       36 + }
       38 + func showRegisterForm(w http.ResponseWriter, r *http.Request) {
                    t, _ := template.ParseFiles("templates/register.html")
       40
                     t.Execute(w, nil)
```

### Main.go changed and updated with db and can be launched

```
users = append(users, newUser)
                      http.Redirect(w, r, "/login", http.StatusSeeOther)
              } else {
                      t, _ := template.ParseFiles("register.html")
                      t.Execute(w, nil)
              db, err := sql.Open("mysql", "root:password@tcp(127.0.0.1:3306)/testdb")
              if err != nil {
                      panic(err.Error())
              defer db.Close()
              username := r.FormValue("username")
              password := r.FormValue("password")
              err = db.QueryRow("SELECT id FROM users WHERE username=?", username).Scan(&user.ID)
              if err != sql.ErrNoRows {
                      t, _ := template.ParseFiles("templates/register.html")
                      t.Execute(w, "Username already taken.")
                      return
              _, err = db.Exec("INSERT INTO users(username, password) VALUES(?, ?)", username, password)
62 +
              if err != nil {
                      panic(err.Error())
64
              http.Redirect(w, r, "/login", http.StatusSeeOther)
67 + }
69 th func showLoginForm(w http.ResponseWriter, r *http.Request) {
              t, _ := template.ParseFiles("templates/login.html")
         t.Execute(w. nil)
```