

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
2 +
3 + go 1.18
4 +
5 + require (
6 +     github.com/andybalholm/brotli v1.0.0 // indirect
7 +     github.com/gofiber/fiber v1.14.6 // indirect
8 +     github.com/gofiber/utls v0.0.10 // indirect
9 +     github.com/gorilla/schema v1.1.0 // indirect
10 +     github.com/klauspost/compress v1.10.7 // indirect
11 +     github.com/mattn/go-colorable v0.1.7 // indirect
12 +     github.com/mattn/go-isatty v0.0.12 // indirect
13 +     github.com/valyala/bytebufferpool v1.0.0 // indirect
14 +     github.com/valyala/fasthttp v1.16.0 // indirect
15 +     github.com/valyala/tcplisten v0.0.0-20161114210144-ceec8f93295a // indirect
16 +     golang.org/x/sys v0.0.0-20200602225109-6fdc65e7d980 // indirect
17 + )
```

Logic for registration changed

```
5 + "net/http"
6 +
7 +
8 +
9 +
10 + type User struct {
11 +     Id int
12 +     name, surname, password, email string
13 +     Id int
14 +     Username, Surname, Password, Email string
15 + }
16 +
17 + var users []User
18 +
19 + func main() {
20 +     start()
21 +
22 +     http.HandleFunc("/", home)
23 +     http.HandleFunc("/register", register)
24 +     http.HandleFunc("/login", login)
25 +     http.ListenAndServe(":8080", nil)
26 + }
27 +
28 + func start() {
29 +     fmt.Println("Choose a action:\n1 Sign up\n2 Log in")
30 +     var choice int
31 +     _, err := fmt.Scanf("%d", &choice)
32 +
33 +     if err != nil {
34 +         log.Fatal(err)
35 +     }
36 +
37 +     func home(w http.ResponseWriter, r *http.Request) {
38 +         t, _ := template.ParseFiles("home.html")
39 +         t.Execute(w, nil)
40 +     }
41 +
42 +     if choice == 1 {
43 +         register()
44 +     } else if choice == 2 {
45 +         authorize()
46 +     }
47 + }
48 +
49 + func register(w http.ResponseWriter, r *http.Request) {
50 +     if r.Method == "POST" {
```

```
60 +     return rand.Intn(1000000)
61 + }
62 +
63 + func authorize() {
64 +     var loggedUser User
65 +     var inputEmail string
66 +     var psword string
67 +
68 +     fmt.Printf("Enter your email: ")
69 +     fmt.Scan(&inputEmail)
70 +
71 +     for i := 0; i < len(users); i++ {
72 +         if users[i].email == inputEmail {
73 +             fmt.Printf("Enter your password: ")
74 +             fmt.Scan(&psword)
75 +             if psword == users[i].password {
76 +                 loggedUser = users[i]
77 +             } else {
78 +                 fmt.Println()
79 +                 fmt.Println("Password is incorrect. Try again.")
80 +                 fmt.Println()
81 +                 start()
82 +             }
83 +         }
84 +     }
85 +
86 +     func login(w http.ResponseWriter, r *http.Request) {
87 +         if r.Method == "POST" {
88 +             username := r.FormValue("username")
89 +             password := r.FormValue("password")
90 +             for _, user := range users {
91 +                 if user.Username == username && user.Password == password {
92 +                     http.Redirect(w, r, "/welcome", http.StatusSeeOther)
93 +                     return
94 +                 }
95 +             }
96 +         }
97 +     }
98 + }
```

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

```
nd 0 deletions.

v 10 templates/home.html
...  ... @@ -0,0 +1,10 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +   <meta charset="UTF-8">
5 +   <title>Title</title>
6 + </head>
7 + <body>
8 +
9 + </body>
10 + </html>

v 10 templates/register.html
...  ... @@ -0,0 +1,10 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +   <meta charset="UTF-8">
5 +   <title>Title</title>
6 + </head>
7 + <body>
8 +
9 + </body>
10 + </html>
```



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

```
12 15
13 - var users []User
14 -
15 16 func main() {
16 -     http.HandleFunc("/", home)
17 -     http.HandleFunc("/register", register)
18 -     http.HandleFunc("/login", login)
19 -     http.ListenAndServe(":8080", nil)
20 +
21 +     db, err := sql.Open("mysql", "root:password@tcp(127.0.0.1:3306)/maindb")
22 +     if err != nil {
23 +         panic(err.Error())
24 +     }
25 +     defer db.Close()
26 +
27 +     router := mux.NewRouter()
28 +     router.HandleFunc("/", home)
29 +     router.HandleFunc("/register", register).Methods("POST")
30 +     router.HandleFunc("/register", showRegisterForm).Methods("GET")
31 +     router.HandleFunc("/login", login).Methods("POST")
32 +     router.HandleFunc("/login", showLoginForm).Methods("GET")
33 +     router.HandleFunc("/welcome", welcome)
34 +     http.ListenAndServe(":8080", router)
35 }
36
37 func home(w http.ResponseWriter, r *http.Request) {
38 -     t, _ := template.ParseFiles("home.html")
39 +     t, _ := template.ParseFiles("templates/home.html")
40 +     t.Execute(w, nil)
41 + }
42
43 + func showRegisterForm(w http.ResponseWriter, r *http.Request) {
44 +     t, _ := template.ParseFiles("templates/register.html")
45 +     t.Execute(w, nil)
46 + }
```

```
1 -     users = append(users, newUser)
2 -     http.Redirect(w, r, "/login", http.StatusSeeOther)
3 - } else {
4 -     t, _ := template.ParseFiles("register.html")
5 -     t.Execute(w, nil)
6 +
7 +     db, err := sql.Open("mysql", "root:password@tcp(127.0.0.1:3306)/testdb")
8 +     if err != nil {
9 +         panic(err.Error())
10 +     }
11 +     defer db.Close()
12 +
13 +     username := r.FormValue("username")
14 +     password := r.FormValue("password")
15 +
16 +     var user User
17 +     err = db.QueryRow("SELECT id FROM users WHERE username=?", username).Scan(&user.ID)
18 +     if err != sql.ErrNoRows {
19 +         t, _ := template.ParseFiles("templates/register.html")
20 +         t.Execute(w, "Username already taken.")
21 +         return
22 +     }
23 +
24 +     _, err = db.Exec("INSERT INTO users(username, password) VALUES(?, ?)", username, password)
25 +     if err != nil {
26 +         panic(err.Error())
27 +     }
28 +
29 +     http.Redirect(w, r, "/login", http.StatusSeeOther)
30 + }
31
32 + func showLoginForm(w http.ResponseWriter, r *http.Request) {
33 +     t, _ := template.ParseFiles("templates/login.html")
34 +     t.Execute(w, nil)
35 + }
```