

Rakhymbayeva Zhaniya
Madiyar Ernar

Our website for searching about cars selling.
Work parts:

Zhaniya - searching + creating database+connection
Ernar - registration +authorization

To run code should to create database testdb with tables users(id,username,password) and products(id,car_name,details, price) and run code with go run main.go

searching items by looking key(searchValue) word from car_name value of products table.

Search for Cars

Search

Cars for Sale

Seville

elit sodales scelerisque mauris sit amet eros
suspendisse accumsan tortor quis turpis sed ante

Price: 79

Eclipse

tincidunt in leo maecenas pulvinar lobortis est
phasellus sit amet erat nulla tempus vivamus in
felis

Price: 97

```

name := "%" + r.FormValue("productName") + "%"
fmt.Println("name:", name)
db, err := sql.Open("mysql", "root:password@/world")

result, err := db.Query("SELECT * FROM products WHERE car_name like ?;", name)
products := []Product{}
for result.Next() {
    var P Product
    err = result.Scan(&P.Id, &P.Car_name, &P.Details, &P.Price)
    products = append(products, P)
    if err != nil {
        panic(err)
    }
}

toInData := toIndexData{
    Products: products,
    Username: savUsername,
}
tpl.ExecuteTemplate(w, "index.html", toInData)

```

registration which saves users data to users table
check if username exists or not

passwords must contain a uppercase

username already taken

Name:

Password:

Register

```
}
r.ParseForm()
username := r.FormValue("username")
password := r.FormValue("password")
fmt.Println("password:", password, "\npswdLength:", len(password))

stmt := "SELECT id FROM users WHERE username = ?"
db, err := sql.Open("mysql", "root:password@/world")
row := db.QueryRow(stmt, username)
var uID int
err = row.Scan(&uID)
if err != sql.ErrNoRows {
    fmt.Println("username already exists, err:", err)
    tpl.ExecuteTemplate(w, "registration.html", "username already taken")
    return
}
```

checking part

```

// func (db *DB) Prepare(query string) (*stmt, error)
var insertStmt *sql.Stmt
insertStmt, err = db.Prepare("INSERT INTO users (username, password) VALUES (?, ?);")
if err != nil {
    fmt.Println("error preparing statement:", err)
    tpl.ExecuteTemplate(w, "registration.html", "there was a problem registering account")
    return
}
defer insertStmt.Close()
var result sql.Result

result, err = insertStmt.Exec(username, password)
rowsAff, _ := result.RowsAffected()
lastIns, _ := result.LastInsertId()
fmt.Println("rowsAff:", rowsAff)
fmt.Println("lastIns:", lastIns)
fmt.Println("err:", err)
if err != nil {
    fmt.Println("error inserting new user")
    tpl.ExecuteTemplate(w, "registration.html", "there was a problem registering account")
    return
}
savUsername = username
http.Redirect(w, r, "/", http.StatusSeeOther)

```

saving username and pass then redirecting to main page

login is working in same way

checking from database and getting response