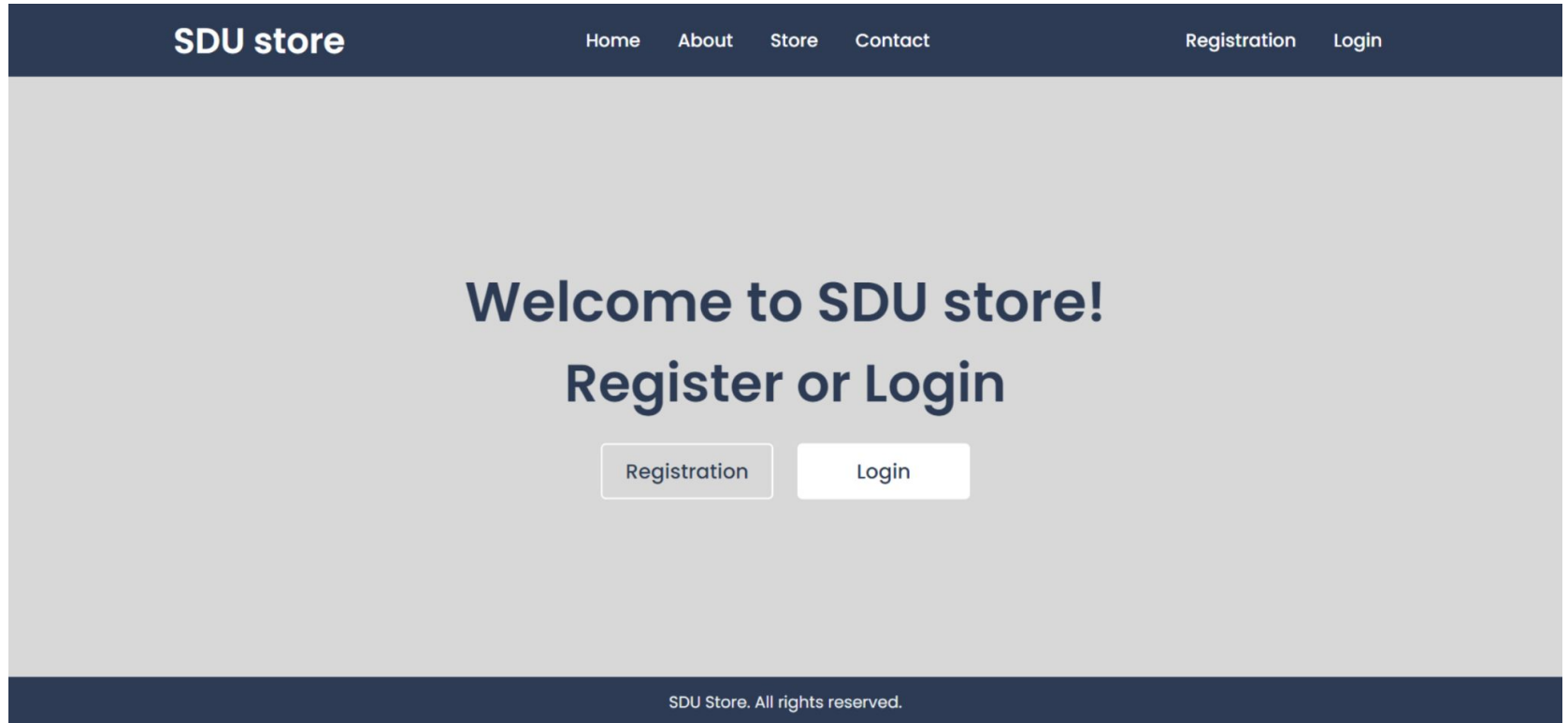


# Project report #4



Yerzhanov Beksultan 200103513  
Tashim Darkhan 200103208

This week we added design of the site and created like template (header and footer) for every .html files.



**ID:**

200103208

**Name:**

Darkhan

**Surname:**

Tashim

**Password:**

\*\*\*\*\*

**Email:**

200103208@stu.sdu.edu.kz

[Register](#)

# Here they are

## Registration successful!

You have been successfully registered.

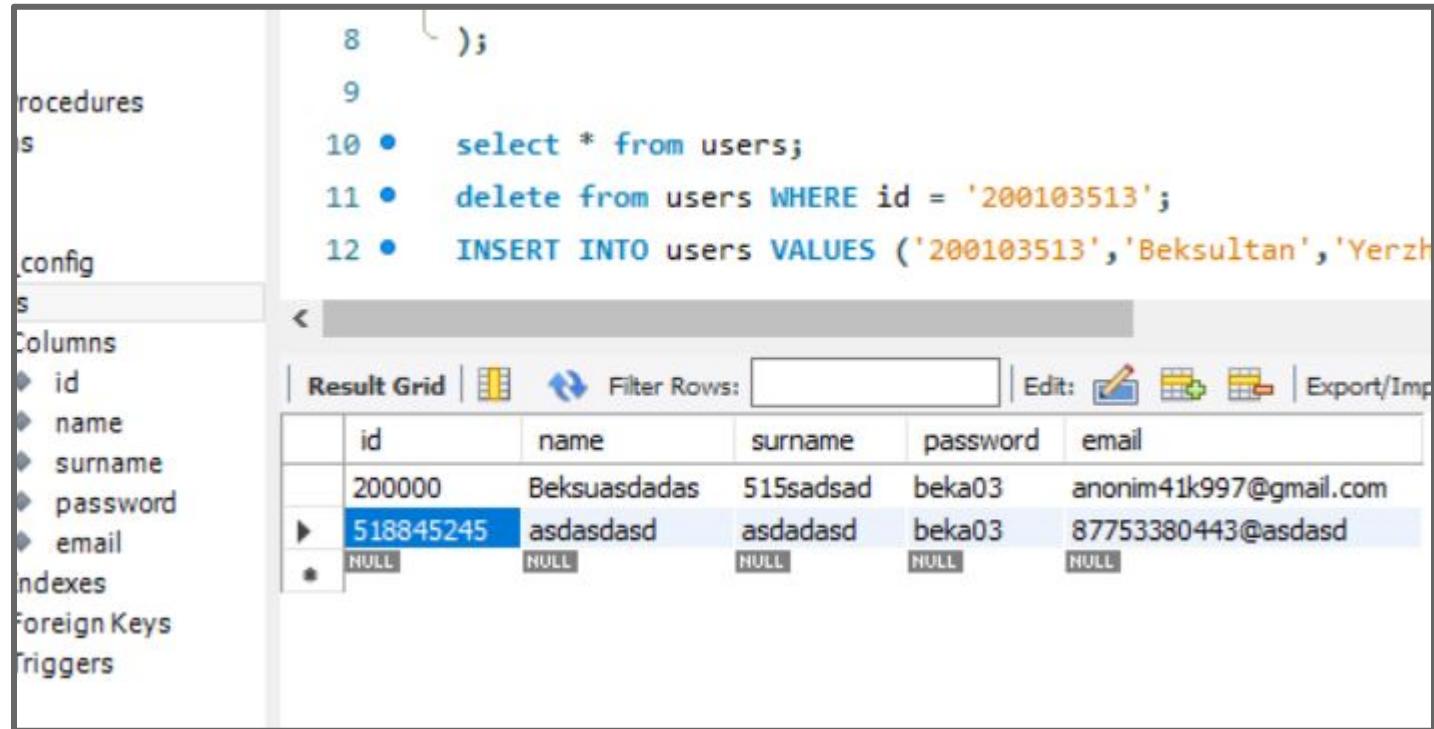
[Return to home page](#)

**ID:****Password:**[Login](#)**Login successful!**

You have been successfully logged in.

[Return to home page](#)

## We created new MySQL Database



The screenshot displays a MySQL database management interface. On the left sidebar, the 'Columns' section is expanded, showing a list of columns: id, name, surname, password, and email. The main area shows a list of SQL queries with line numbers 8 through 12. The queries are:

- 8 `);`
- 9
- 10 `• select * from users;`
- 11 `• delete from users WHERE id = '200103513';`
- 12 `• INSERT INTO users VALUES ('200103513', 'Beksultan', 'Verzh`

Below the queries, there is a 'Result Grid' section. It includes a 'Filter Rows' input field and several icons for editing and exporting. The result grid shows a table with the following data:

	id	name	surname	password	email
	200000	Beksuasdasdas	515sadsad	beka03	anonim41k997@gmail.com
▶	518845245	asdasdasd	asdadasd	beka03	87753380443@asdasd
*	NULL	NULL	NULL	NULL	NULL

# Connect it to our Main.go file

```
5  username, dbname, password, EmptyString
6  }
7
8  var db *sql.DB
9  var err error
10
11  func main() {
12      db, err = sql.Open(driverName: "mysql", dataSourceName: "root:Bekasql20232003@/sys")
13      if err != nil {
14          panic(err.Error())
15      }
16      http.HandleFunc("/home", homeHandler)
```

## Also some Insert Function to Registration

```
}  
t.ExecuteTemplate(w, name: "register", data: nil)  
} else if r.Method == "POST" {  
    id, _ := strconv.Atoi(r.FormValue(key: "id"))  
    name := r.FormValue(key: "username")  
    surname := r.FormValue(key: "surname")  
    password := r.FormValue(key: "password")  
    email := r.FormValue(key: "email")  
    query := "INSERT INTO users VALUES ('" + strconv.Itoa(id) + "','" + name +  
        "','" + surname + "','" + password + "','" + email + "');"   
    insert, err := db.Query(query)  
    if err != nil {  
        fmt.Println(query)  
        panic(err.Error())  
    }  
    defer insert.Close()  
    t, err := template.ParseFiles(filenamees...: "templates/register-success.html", "templates  
    if err != nil {
```

# Checking it for login page

```
id, _ := strconv.Atoi(r.FormValue( key: "id"))
password := r.FormValue( key: "password")

row := db.QueryRow( query: "SELECT * FROM users WHERE id = ? AND password = ?", id, pass

var userid uint
var name string
var surname string
var userEmail string
var userPassword string
err = row.Scan(&userid, &name, &surname, &userPassword, &userEmail)
if err == sql.ErrNoRows {
    fmt.Println( a...: "No user found with the given email and password.")
} else if err != nil {
    panic(err)
} else {
    t, err := template.ParseFiles( filenames...: "templates/login-success.html", "templates
    if err != nil {
```