

# Securitate Web

Laboratorul 02

## Securizarea formularului de înregistrare a unui utilizator nou

În acest laborator veți continua aplicația web din laboratorul precedent și veți securiza formularul de înregistrare a unui utilizator nou atât la client, cât și la server. În cadrul laboratorului veți lucra în grup pentru a răspunde la întrebări. Trebuie să completați răspunsurile direct în documentul de laborator.

Nu uitați să încărcați ce lucrați pe GitHub Classroom.

Membrii grupului:

- Barcan Nicoleta-Gabriela
- Apostol Roxana-Maria
- Vocurek Denisa-Maria

### 1. Securizarea formularului client-side

Câmpurile care ar trebui să fie prezente sunt:

Nume câmp	Tip câmp	Număr minim de caractere	Număr maxim de caractere	Expresie regulată de validare
Nume utilizator	varchar	5	15	/^[\a-zA-Z0-9 ]{5,15}\$/
Email	varchar	8	50	/^[\a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{8,50}\$/
Parolă	varchar	10	20	/^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9]).{10,20}\$/
Confirmarea parolei	-	-	-	Functie de verificare aceeasi parola
Nume	varchar	3	30	/^[\A-Za-zA-Â-Â-Î-Î-Ş-Ş-]{3,30}\$/
Prenume	varchar	3	30	/^[\A-Za-zA-Â-Â-Î-Î-Ş-Ş-]{3,30}\$/
Nume afișat	varchar	3	15	/^[\A-Za-zA-Â-Â-Î-Î-Ş-Ş-]{3,15}\$/
Adresă URL site propriu	varchar	10	200	/^https?:\/\/[a-zA-Z0-9.-]+\.[a-zA-Z]{10,200}:/
Imagine profil	fisier	10 KB	2MB	^.+\.(jpg jpeg png gif bmp webp)\$

#### 1. Care sunt restricțiile pentru fiecare câmp în parte la nivel de client?

- > **Nume utilizator** trebuie sa contine intre 3 si 15 caractere si poate include doar litere, cifre si underscore.
- > **Emailul** trebuie sa fie intr-un format valid de email, trebuie sa contine intre 8 si 50 caractere.
- > **Parola** trebuie sa aiba intre 6 si 20 de caractere, sa contine cel putin o litera mare, una mica si o cifra.
- > **Confirmarea parolei** trebuie sa fie identica cu valoarea introdusa in campul „Parola”.
- > **Numele** trebuie sa aiba intre 3 si 20 de caractere, continand doar litere românești si cratime.

- > **Prenumele** trebuie sa aiba intre 3 si 30 de caractere si sa respecte aceleasi reguli ca si campul „Nume”.
- > **Numele afisat** trebuie sa contine intre 3 si 15 caractere si poate include litere, cifre, cratime si caractere specifice limbii romane.
- > **Adresa URL** a site-ului propriu trebuie sa inceapa cu http:// sau https://, sa aiba un domeniu valid si o extensie formata din 10 pana la 60 de litere.
- > **Imaginea de profil** trebuie sa aiba intre 10KB si 20MB si sa extensia jpg|jpeg|png|gif|bmp|webp.

## **2. Ce feedback ar trebui sa primeasca utilizatorul in caz de completare incorecta?**

- > La fiecare camp in parte, utilizatorul va primi un mesaj de eroare, unde i se va comunica conform regexului cum ar trebui sa completeze campul

The screenshot shows a registration form with two fields highlighted in red:

- Nume utilizator:** The input field contains "ac". Below it, an error message in red reads: "Nume utilizator invalid (3-15 caractere, doar litere, cifre și )."
- Email:** The input field contains "blabla.ro". Below it, an error message in red reads: "Email invalid (ex: nume@domeniu.com)."

## **3. Cand ajunge browser-ul sa trimita cererea la server?**

- > După completarea formularului, toate datele sunt verificate că respectă formatul cerut
- > La apasarea butonului de Submit de catre utilizator, daca formularul este valid, se apeleaza fetch pentru <http://localhost:3000/register>
- > Browser-ul trimite cererea http post catre server

## **4. De cate ori ar trebui sa incerce inregistrarea?**

- > Inregistrarea ar trebui incercata de maximum 3 ori intr-un interval de 1 minut pentru a evita atacuri de tip DOS sau SQL Injection.
- > Daca dupa 3 incercari inregistrarea esueaza, utilizatorul ar trebui pus sa astepte cateva minute sau sa fie blocat temporar.

## **5. Este necesar un Captcha?**

- > Pentru un site public, Captcha nu este o cerinta obligatorie, dar este foarte util pentru ca permite diferentierea dintre un utilizator om si bot
- > Ajuta la oprirea incarcarii automate a datelor in formular

- > Dupa ce utilizatorulincearca de mai mult de 3 ori sa se inregistreze, ar trebui pus sa treaca un test Captcha

## 2. Sanitizarea datelor primite la server

Care sunt metodele de sanitizare în funcție de serverul web / limbajul de programare ales?

NodeJS cu limbajul JavaScript

- > Utilizarea regexurilor si verificarilor ca datele completate in campurile formularului respecta un anumit format si nu depaseste un numar de caractere
- > Verificarea formatului si dimensiunii pozelor incarcate
- > Se poate folosi Cors Controlat ca sa trimita cereri doar de la propriul client
- > Se pot face verificari asupra datelor incarcate pentru a nu primi cod html sau javascript, un regex de tip /<[^>]\*>/g
- > Constrageri pentru coloanele tabelei utilizatori in baza de date

## 3. Stocarea datelor la server

Cum ar trebui de stocat imaginea de profil?

Tabela `utilizatori` conține proprietățile din formularul de înregistrare (fără câmpul de confirmare a parolei, iar parola va fi reținută deocamdată în clar, fără hash pe ea) și proprietățile: id (cheie primă autoincrement), tip (utilizator, administrator), timpInregistrare.

Componeti scriptul SQL de creare a bazei de date și a tabelei `utilizatori`. Atenție la restricțiile aplicate câmpurilor.

SQL

```
CREATE DATABASE swProject;
CREATE TABLE swProject.utilizatori (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tip ENUM('utilizator', 'administrator') DEFAULT 'utilizator',
    timpInregistrare TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    numeUtilizator VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL UNIQUE,
    parola VARCHAR(255) NOT NULL,
    nume VARCHAR(50),
    prenume VARCHAR(50),
    numeAfisat VARCHAR(50),
    adresaURL VARCHAR(255),
    caleImagineProfil VARCHAR(255) DEFAULT NULL
);
```