

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	-	-	-	Funcție de verificare aceeași parola
Nume	varchar	3	30	/^[A-Za-zăâîșțĂÂÎȘȚ-]{3,30}\$/
Prenume	varchar	3	30	/^[A-Za-zăâîșțĂÂÎȘȚ-]{3,30}\$/
Nume afișat	varchar	3	15	/^[A-Za-z0-9ăâîșțĂÂÎȘȚ-]{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 să conțină între 3 și 15 caractere și poate include doar litere, cifre și underscore.
- > **Emailul** trebuie să fie într-un format valid de email, trebuie să conțină între 8 și 50 caractere.
- > **Parola** trebuie să aibă între 6 și 20 de caractere, să conțină cel puțin o literă mare, una mică și o cifră.
- > **Confirmarea parolei** trebuie să fie identică cu valoarea introdusă în câmpul „Parola”.
- > **Numele** trebuie să aibă între 3 și 20 de caractere, conținând doar litere românești și cratime.

-> **Prenumele** trebuie sa aiba intre 3 si 30 de caractere si sa respecte aceleasi reguli ca si campul „Nume”.

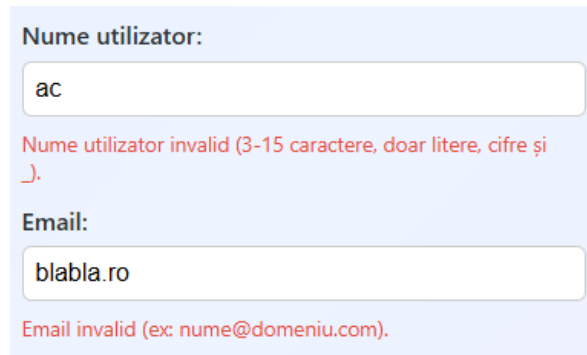
-> **Numele afisat** trebuie sa contina intre 3 si 15 caractere si poate include litere, cifre, cratime si caractere specifice limbii romane.

-> **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 să primească utilizatorul în caz de completare incorectă?

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



The image shows a registration form with two input fields. The first field is labeled 'Nume utilizator:' and contains the text 'ac'. Below it, a red error message reads: 'Nume utilizator invalid (3-15 caractere, doar litere, cifre și _)'. The second field is labeled 'Email:' and contains the text 'blabla.ro'. Below it, a red error message reads: 'Email invalid (ex: nume@domeniu.com)'.

3. Când ajunge browser-ul să trimită 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 câte ori ar trebui să încerce înregistrarea?

-> Înregistrarea ar trebui încercata de maximum 3 ori într-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

-> După ce utilizatorul încearcă de mai mult de 3 ori să se înregistreze, ar trebui pus să treacă 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 și verificărilor ca datele completate în câmpurile formularului respectă un anumit format și nu depășește un număr de caractere
- > Verificarea formatului și dimensiunii pozelor încărcate
- > Se poate folosi Cors Controlat ca să trimită cereri doar de la propriul client
- > Se pot face verificări asupra datelor încărcate pentru a nu primi cod HTML sau JavaScript, un regex de tip `/<[>]*>/g`
- > Constrângeri pentru coloanele tabelului utilizatori în baza de date

3. Stocarea datelor la server

Cum ar trebui să 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 primară autoincrement), tip (utilizator, administrator), timpÎnregistrare.

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

SQL

```
CREATE DATABASE swProiect;
CREATE TABLE swProiect.utilizatori (
  id INT AUTO_INCREMENT PRIMARY KEY,
  tip ENUM('utilizator', 'administrator') DEFAULT 'utilizator',
  timpÎnregistrare 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
);
```