SRP - LAB 5

Spore kriptografske hash funkcije

zadatak: Implementirati proces inicijalne registracije i login korisnika korištenjem sigurne Argon2 *password hashing* funkcije.

Stvaramo bazu podataka u koju ćemo spremat podatke o korisničkom imenu korisnika i hash vrijednost njegove lozinke.

Sign in

Prije unosa novih korisnika, potrebno je osigurati da je korisničko ime jedinstveno za svakog korisnika.

Unosimo nove korisnike:

```
register_user(username="jdoe", password="password")
register_user(username="jdoe", password="password2")
register_user(username="jean_doe", password="password")
```

Provjeravamo bazu.

U bazu su se unijela samo dva podatka, jdoe i jean_doe, jer je pri pokušaju registracije drugog jdoe-a u bazi pronađen korisnik s istim korisničkim imenom i registracija je bila neuspješna.

Primjećujemo da iako jdoe i jean_doe imaju identične lozinke (password), njihove hash vrijednosti su drugačije. To je zbog soli koju generira argon2 kriptografska hash funkcija.

getpass in python - prompts the user for a password without echoing

Log in

Pri prijave korisnika u sustav, od njega se traži da unese svoje korisničko ime i lozinku.

Lozinka koju je unio korisnik provlači se kroz argon2 i dobivena hash vrijednost uspoređuje se s onom u bazi. Ako se samo jedan karakter ne poklapa, prijava korisnika bit će odbijena.

Da bi argon2 uspješno validirao lozinku, mora imati spremljeno i pripadajuću sol za tu lozinku.

Zašto u funkciji `do_sign_in_user()` *tražimo od korisnika da uvijek unese oboje,* `username` *i* `password`*, čak iako* `username` *potencijalno nije ispravan?*

Cilj je što manje informacija odati napadaču. Zato printamo poruku: `print("Invalid username or password.").`