Code für die Datenbank: user\_accounts

CREATE DATABASE user\_accounts;

USE user\_accounts;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY, // Das ist der Primary Key und id create immer eine neue nummer bei einem neuen Eintrag

email VARCHAR(255) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

reset\_token VARCHAR(64), -- Für Passwort-Reset-Token

token\_expires\_at DATETIME -- Ablaufdatum für das Token

);

Wie funktioniert die Password -vergessen Funktion:

Wenn ein Kunde sein Passwort vergisst und auf den **"Passwort vergessen"**-Button klickt, funktioniert der Prozess meist folgendermaßen:

**1. "Passwort vergessen"-Button**

* Der Nutzer klickt auf den Button.
* Die Aktion startet einen **Passwort-Wiederherstellungs-Workflow**.

**2. E-Mail-Adresse des Nutzers erfassen**

* Der Nutzer wird aufgefordert, seine registrierte **E-Mail-Adresse** einzugeben.
* Die eingegebene E-Mail wird an den Server geschickt.

**3. Prüfen, ob die E-Mail existiert**

* Der Server führt eine **Abfrage in der Datenbank** durch, um zu prüfen, ob die eingegebene E-Mail-Adresse existiert:
* SELECT \* FROM users WHERE email = 'eingegebene\_email';
* **Antwortmöglichkeiten:**
  + **E-Mail gefunden**: Der Server fährt mit der Passwort-Wiederherstellung fort.
  + **E-Mail nicht gefunden**: Eine Fehlermeldung wird angezeigt (z.B. "Diese E-Mail-Adresse ist nicht registriert.").

**4. Generieren eines sicheren Passwort-Reset-Tokens**

* Wenn die E-Mail existiert, wird ein **sicheres Token** generiert. Dieses Token sollte:
  + Zufällig und lang genug sein.
  + Ein Ablaufdatum haben (z.B. 1 Stunde).
* Ein Token kann z.B. so generiert werden:
* **import secrets**
* **token = secrets.token\_hex(32) # 64 Zeichen langer sicherer Token**

**5. Speichern des Tokens in der Datenbank**

* Das Token wird in der Datenbank gespeichert, zusammen mit:
  + Der **E-Mail-Adresse** des Nutzers.
  + Der **Ablaufzeit** des Tokens.
* Beispiel-SQL-Abfrage:
* **UPDATE users**
* **SET reset\_token = 'generiertes\_token',**
* **token\_expires\_at = NOW() + INTERVAL 1 HOUR**
* **WHERE email = 'eingegebene\_email';**

**6. Token-Link per E-Mail versenden**

* Der Server sendet eine E-Mail an die registrierte Adresse mit einem Link zur Passwort-Zurücksetzung.
* Der Link sieht z.B. so aus:
* https://www.deinewebsite.de/reset-password?token=GENERIERTER\_TOKEN

**7. Überprüfen des Tokens (wenn Nutzer Link klickt)**

* Wenn der Nutzer auf den Link klickt:
  1. Der Server liest das **Token** aus der URL.
  2. Es wird geprüft, ob das Token gültig ist und noch nicht abgelaufen:
  3. **SELECT \* FROM users**
  4. **WHERE reset\_token = 'eingabe\_token'**
  5. **AND token\_expires\_at > NOW();**
  6. Wenn das Token gültig ist, kann der Nutzer ein neues Passwort setzen.

**8. Neues Passwort speichern**

* Das neue Passwort wird verschlüsselt (gehasht) gespeichert, um die Sicherheit zu gewährleisten:
  + **Hashing mit bcrypt** (empfohlen):
  + **import bcrypt**
  + **hashed\_pw = bcrypt.hashpw(neues\_passwort.encode(), bcrypt.gensalt())**
* Das verschlüsselte Passwort wird in der Datenbank gespeichert:
* **UPDATE users**
* **SET password = 'hashed\_password',**
* **reset\_token = NULL,**
* **token\_expires\_at = NULL**
* **WHERE reset\_token = 'eingabe\_token';**

**9. Bestätigung und Login**

* Der Nutzer erhält eine Bestätigung, dass das Passwort geändert wurde.
* Anschließend kann er sich mit seinem neuen Passwort einloggen.

**Wichtige Sicherheitsaspekte:**

1. **Tokens sicher und zufällig generieren:** Nutze kryptografisch sichere Funktionen wie secrets oder bcrypt.
2. **Ablaufzeit des Tokens:** Tokens sollten nur für kurze Zeit gültig sein (z.B. 1 Stunde).
3. **Passwort-Hashing:** Speichere niemals Passwörter im Klartext. Nutze sichere Hash-Funktionen wie **bcrypt** oder **Argon2**.
4. **Token-Löschung nach Gebrauch:** Lösche das Token sofort, nachdem es verwendet wurde.

**Ablauf zusammengefasst:**

1. Nutzer klickt auf **"Passwort vergessen"**.
2. Server prüft E-Mail und generiert ein Token.
3. Ein **Link mit Token** wird per E-Mail versendet.
4. Nutzer klickt auf den Link und setzt ein neues Passwort.
5. Neues Passwort wird gehasht und in der Datenbank gespeichert.

Damit ist der Passwort-Reset-Prozess abgeschlossen! 😊