**Dokumentation: HederligeHarrysBilar - Användarhantering och Inloggningssystem**

**Systembeskrivning**

Detta system hanterar användare, inloggningar och rollhantering för webbplatsen HederligeHarrysBilar.   
Det gör det möjligt att:

* **Registrera användare** med säker inloggning och verifiering.
* **Hantera inloggningsförsök**, inklusive misslyckade försök och IP-adresser.
* **Ge användare en roll** (Customer/Admin).
* **Låsa och låsa upp konton** efter tre misslyckade inloggningsförsök.
* **Spara och hantera återställning av lösenord.**
* **Rapportera inloggningsaktivitet och felaktiga försök per IP-adress.**

**Databasens struktur - Tabeller och deras betydelse  
  
Users tabellen:**

|  |  |  |
| --- | --- | --- |
| Kolumn | Typ | Beskrivning |
| UserID | INT (PK) | Unikt ID för varje användare |
| FirstName | NVARCHAR(50) | Användarens förnamn |
| LastName | NVARCHAR(50) | Användarens efternamn |
| EmailAddress | NVARCHAR(255) UNIQUE | Unik e-postadress för inloggning |
| PasswordHash | NVARCHAR(128) | Hashat lösenord för säker inloggning |
| Salt | NVARCHAR(50) | Unikt salt för extra säkerhet |
| Address | NVARCHAR(100) | Gatuadress |
| PostalCode | NVARCHAR(20) | Postnummer |
| City | NVARCHAR(50) | Stad |
| Country | NVARCHAR(50) | Land |
| PhoneNumber | NVARCHAR(20) | Telefonnummer |
| IsVerified | BIT | Om kontot är verifierat (0/1) |
| IsLockedOut | BIT | Om kontot är låst pga misslyckade försök (0/1) |
| Role | NVARCHAR(20) | Användarens roll (Customer eller Admin) |
| PasswordResetCode | NVARCHAR(255) | Kod för återställning av lösenord |
| PasswordResetExpiry | DATETIME | Utgångstid för återställningskod |
| CreatedAt | DATETIME | Skapelsedatum |
| UpdatedAt | DATETIME | Senaste uppdateringsdatum |

**LoginAttempts** tabellen:

|  |  |  |
| --- | --- | --- |
| Kolumn | Typ | Beskrivning |
| AttemptID | INT (PK) | Unikt ID för varje inloggningsförsök |
| UserID | INT (FK) | Användaren som försökte logga in |
| Email | NVARCHAR(255) | E-postadress använd vid försöket |
| IPAddress | NVARCHAR(50) | Användarens IP-adress |
| AttemptDate | DATETIME | Datum och tid för försöket |
| Success | BIT | Om inloggningen lyckades (1) eller ej (0) |

**Stored Procedures och deras funktion**

* **RegisterUser** - Registrerar nya användare och hashar lösenordet med salt.
* **UpdateUserRole** - Uppdaterar en användares roll till 'Customer' eller 'Admin'.
* **VerifyAccount** - Bekräftar en användares e-post.
* **ForgotPassword** - Skickar en återställningskod och giltighetstid.
* **SetForgottenPassword** - Uppdaterar lösenordet efter att en giltig återställningskod använts.
* **TryLogin** - Hanterar inloggningsförsök, loggar dem och låser konton vid tre misslyckade försök.
* **LockUnlockUser** - Låser eller låser upp en användare.

**Rapporter och vyer**

* **UserLoginReport** - Visar den senaste lyckade och misslyckade inloggningen för varje användare.
* **LoginAttemptsPerIP** - Sammanställer antal lyckade och misslyckade försök per IP-adress

**Optimering och förbättringar**

**Nuvarande optimeringsstrategier:**

* **Indexering**:
  + UserID är primärnyckel för snabb sökning.
  + EmailAddress är unik för att effektivt identifiera användare.
  + Index för snabbare sökning på e-postadress
  + Index för snabbare sökning på IP-adresser i LoginAttempts
* **Säkerhet:**
  + Lösenordshantering genom SHA2\_256 och unikt salt.
  + Låsning av konto efter 3 misslyckade försök.
* **Effektiv hantering av data:**
  + Temporär tabell som ##TempLoginLogs används för att lagra inloggningsförsök under exekvering utan att påverka databasen.
  + Views skapad för att snabbt hämta data som tex (UserLoginReport,LoginAttemptsPerIP)

**Möjliga förbättringar för framtiden:**

* **Index på AttemptDate i LoginAttempts** för snabbare sökning av inloggningsförsök per tid.
* **Skapa Clustred index på CreatedAt i Users** för snabbare sökning av nya användare.
* **Bättre e-postverifiering** genom att skicka riktiga e-postmeddelanden vid registrering.
* **Tvåfaktorsautentisering (2FA)** för extra säkerhet vid inloggning.

**Sammanfattning**

Det här systemet är byggt för att vara säkert och snabbt. Det gör det möjligt för användare att registrera sig, logga in och hantera sina konton. Det skyddar mot upprepade felaktiga inloggningsförsök. Systemet kan bli ännu bättre med snabbare sökningar och extra säkerhet som tvåstegsverifiering.