Security

Bij dit project heb ik zowel een inlogscherm als een registratiescherm ontwikkeld. Het registratiescherm is ontworpen met een sterke focus op beveiliging en zorgt ervoor dat gebruikers alleen veilige wachtwoorden kunnen aanmaken. Dit maakt het registratieproces bewust iets minder eenvoudig, maar verhoogt de algehele veiligheid van het systeem aanzienlijk.

Ik heb verschillende controles toegevoegd aan de registratie om te garanderen dat gebruikers een sterk wachtwoord instellen. Enkele van deze controles zijn:

- Het wachtwoord moet minimaal één **hoofdletter** bevatten, zodat het moeilijker te raden is.
- Het wachtwoord moet minimaal één **cijfer** bevatten, wat voorkomt dat eenvoudige, voorspelbare wachtwoorden worden gebruikt.
- Het wachtwoord moet een **speciaal teken** bevatten, zoals @, #, $ of %, wat helpt bij het verhogen van de complexiteit.
- Het wachtwoord mag niet lijken op de **gebruikersnaam**, zodat het niet makkelijk te raden is door anderen.

Naast het registratiescherm heb ik ervoor gezorgd dat gebruikers niet bij de **indexpagina** of andere beveiligde onderdelen van de applicatie kunnen komen zonder eerst in te loggen. Dit betekent dat ongeautoriseerde toegang volledig wordt geblokkeerd totdat een gebruiker succesvol is geauthentiseerd.

Verder heb ik een **rol gebaseerd toegangscontrolesysteem** geïmplementeerd om onderscheid te maken tussen gewone gebruikers en administrators. Dit houdt in dat:

- **Klanten** alleen toegang hebben tot de functionaliteiten die voor hen bedoeld zijn en geen mogelijkheid hebben om gegevens te wijzigen die zij niet mogen aanpassen.
- **Admins** speciale rechten hebben om beheerfuncties uit te voeren, zoals het wijzigen van instellingen, het bekijken van rapportages, en het beheren van andere gebruikers.

Door deze maatregelen is het systeem zowel gebruiksvriendelijk als veilig, met een duidelijke scheiding van rechten en toegangsniveaus. Dit zorgt ervoor dat gebruikers alleen toegang hebben tot wat voor hen relevant is, terwijl de integriteit van het systeem wordt beschermd tegen ongewenste aanpassingen of inbreuken.

Waarom is Django goed voor de beveiliging?

Bron: https://escape.tech/blog/best-django-security-practices

Django wordt algemeen erkend als een veilig webframework. Het bevat ingebouwde beveiligingsfuncties om te beschermen tegen veelvoorkomende kwetsbaarheden zoals Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) en SQL-injectie. Met een robuust authenticatiesysteem, ORM-beveiliging en andere beveiligingsmaatregelen biedt Django een solide basis voor het ontwikkelen van veilige webapplicaties. Daarnaast dragen de actieve Django-community (18 jaar oud!) en regelmatige updates bij aan het handhaven van een hoog beveiligingsniveau.