Visualiseren van Data uit Nexedge

Brecht Dobbelaere 20/1/25

Abstract: Via de webpagina is het mogelijk om snel en makkelijk opzoekingen of analyse uit te voeren op de data uit het Nexedge radiosysteem.

### Introductie:

Ik werk als technieker bij de communicatiedienst, met een specifieke focus op radio's (walkietalkies). Een deel van onze radio's (ongeveer 600) maakt gebruik van het digitale Kenwood Nexedge-protocol. NEXEDGE is een digitale radiotechnologie van Kenwood die een betere geluidskwaliteit, verbeterde beveiliging en hogere efficiëntie biedt in vergelijking met analoge systemen.

Onze huidige infrastructuur omvat drie Nexedge-centrales die elk hun eigen data beheren. Deze data worden uitgelezen en omgezet naar CSV-bestanden. Het probleem is echter dat de data van elke site afzonderlijk moet worden uitgelezen en opgeslagen. Hierdoor wordt het zoeken naar specifieke gegevens erg lastig, omdat er door meerdere bestanden moet worden gegaan, vaak met een grote hoeveelheid data. Eén bestand bevat bij een volledig geheugen bijvoorbeeld 42.000 datalijnen. Dit maakt handmatige analyse zeer complex en tijdrovend.

### Functionaliteiten:

De applicatie is gebouwd op basis van het Flask-framework, waarmee een intuïtieve en gebruiksvriendelijke web omgeving is gecreëerd. Dankzij de flexibiliteit van Flask kunnen gebruikers eenvoudig navigeren en toegang krijgen tot diverse functies binnen de applicatie. De structuur is overzichtelijk opgezet en bevat pagina's waar data wordt weergegeven, filters kunnen worden toegepast, en uploads beheerd worden.

De data wordt verzameld uit een reeks CSV-bestanden die per site worden gegenereerd vanuit het Nexedge-systeem. Gebruikers hebben de mogelijkheid om deze data te doorzoeken met behulp van specifieke filters. Dit maakt het eenvoudig om gericht te zoeken naar bepaalde ID-nummers die mogelijk problemen veroorzaken. Zodra de gewenste data is gefilterd, kan deze worden geëxporteerd naar een nieuw CSV-bestand. Dit exportmechanisme is van cruciaal belang, omdat de geëxporteerde data vaak wordt doorgestuurd naar de leverancier van het systeem voor verdere analyse en probleemoplossing.

Daarnaast ondersteunt de applicatie het toevoegen van nieuwe bestanden, maar deze £££  
functionaliteit is alleen beschikbaar voor gebruikers die inloggen met een beheerdersaccount. Dit waarborgt de integriteit van de gegevens en voorkomt ongeoorloofde wijzigingen.

Een bijkomend voordeel van de applicatie is de schaalbaarheid: nieuwe functionaliteiten kunnen eenvoudig worden toegevoegd, en de filters en exportopties zijn flexibel aan te passen aan veranderende behoeften. Dankzij deze eigenschappen biedt de applicatie een robuuste en efficiënte oplossing voor het beheren en analyseren van systeemdata.

### Problemen en oplossingen:

Een van de grootste uitdagingen bij het ontwikkelen van de applicatie was het ontwerpen van een goede en gebruiksvriendelijke gebruikersinterface (UI) binnen het Flask-framework. Het combineren van functionaliteit en overzichtelijkheid bleek lastig, maar dankzij voorbeelden uit de lessen en de ondersteuning van ChatGPT heb ik dit opgelost.

Om de data overzichtelijk te presenteren, heb ik ervoor gekozen om deze op te delen in pagina's van maximaal 50 rijen. Dit verbeterde niet alleen de leesbaarheid, maar zorgde er ook voor dat gebruikers eenvoudig door grote datasets kunnen navigeren. Het implementeren van deze paginering was een uitdaging die veel onderzoek en experimenteren vergde, maar uiteindelijk resulteerde in een solide en visueel aantrekkelijke oplossing.

Voor het toepassen van verschillende filters op de data maakte ik gebruik van de krachtige Q()-functie uit het **MongoEngine**-pakket. Deze functie bood de flexibiliteit om complexe query's te bouwen en maakte het mogelijk om eenvoudig aan specifieke zoekcriteria te voldoen. Dankzij de combinatie van MongoEngine en de filterfunctionaliteit kan de applicatie nu efficiënt werken met dynamische datasets, zonder dat dit ten koste gaat van de prestaties.

### Conclusie

Al met al heb ik een werkend en gebruiksvriendelijk programma gemaakt dat doet wat het moet doen. Tijdens het proces heb ik veel geleerd, vooral over hoe je met een MongoDB-database werkt en hoe je gebruikersbeheer opzet. Het was soms een uitdaging, maar ook echt interessant om hiermee aan de slag te gaan.

Er zijn zeker nog dingen die ik aan het programma wil toevoegen. Ik heb al een paar ideeën om het verder uit te breiden en nóg beter te maken. Ik kijk ernaar uit om hiermee door te gaan en het steeds verder te verbeteren.

Dit project was niet alleen leerzaam, maar ook heel tof om aan te werken, en ik ben blij met wat ik tot nu toe heb bereikt!