Front end analyse  
  
De opdracht voor casusgroep 3 vraagt om een dashboard te ontwikkelen dat winstgerelateerde metrieken van een pizza webshop visualiseert, met als belangrijkste onderdelen:

1. **Data-analyse en visualisatie** in Python en Excel, inclusief het identificeren en uitsluiten van outliers om duidelijke inzichten te geven.
2. **Front-end ontwikkeling van het dashboard**, waarbij gebruikers eenvoudig de belangrijkste metrieken kunnen bekijken.

Voor een eenvoudig te realiseren front-end zijn er een paar opties:

### Optie 1: Streamlit

* **Voordelen**: Streamlit is gebruiksvriendelijk en integreert goed met Python, waardoor je jouw visualisaties en data makkelijk kunt presenteren in een webapp. Het heeft ingebouwde componenten voor interactie en lay-out.
* **Geschikt voor**: Snel ontwikkelen van interactieve dashboards, vooral als je minder ervaring hebt met webontwikkeling.
* **Voorbeeld**: Je kunt jouw Python-visualisaties, zoals bar charts en scatterplots, eenvoudig toevoegen en interacties zoals filters of selecties implementeren.

### Optie 2: Flask/Django gecombineerd met een JavaScript library (zoals Chart.js of Plotly)

* **Voordelen**: Met Flask of Django kun je meer controle hebben over de serverkant en gegevensverwerking. Door Chart.js of Plotly toe te voegen, kun je dynamische grafieken en visualisaties rechtstreeks in de browser renderen.
* **Geschikt voor**: Grotere projecten met meer ervaring in webontwikkeling.
* **Voorbeeld**: In Flask kun je de Python-visualisaties omzetten naar afbeeldingen of dataformaat (zoals JSON) en deze in de front-end laden met JavaScript.

### Optie 3: Jupyter Notebooks met Voila

* **Voordelen**: Hiermee kun je een Jupyter Notebook omzetten in een interactieve web-app zonder dat je diep in HTML of JavaScript hoeft te duiken. Het werkt goed als je alleen eenvoudige visualisaties hoeft te laten zien.
* **Geschikt voor**: Snelle prototyping en kleinere dashboards.