



# Inleiding

Deze debriefing behandelt de projectvereisten en de gekozen aanpak om deze te vervullen, bespreekt de geselecteerde technologieën – **Vue.js**, **ionic** en **Capacitor** – en hoe zij samen een efficiënte, schaalbare en toekomstbestendige Progressieve Web App (PWA) mogelijk maken.

## Analyse van de belangrijkste projectvereisten:

- Inspectiesysteem waarmee inspecteurs veldinspecties kunnen uitvoeren, schade kunnen documenteren en foto's en opmerkingen direct aan rapporten kunnen toevoegen.
- Offline toegang tot gegevens, zodat inspecteurs ook zonder internetverbinding kunnen werken.
- Gegevens ophalen en verzenden via een RESTful API voor efficiënte communicatie met andere systemen.
- Flexibele en schaalbare applicatie die in de toekomst eenvoudig geavanceerde functies kan toevoegen, zoals CRM-integratie.

## Overwegingen bij het kiezen van technologie

De keuze voor Vue.js, Ionic en Capacitor was gebaseerd op de volgende sleutelfactoren:

### Ecosysteem

**Vue.js** en **ionic** worden veel gebruikt door ontwikkelaars. Ze hebben een grote community, uitstekende documentatie en veel tutorials. Dit maakt het een stuk makkelijker om met deze technologieën te werken en snel door problemen heen te komen.

**Capacitor** maakt het daarnaast heel eenvoudig om toegang te krijgen tot apparaatfunctionaliteiten zoals de camera en de lokale opslag, iets wat we echt nodig hebben voor deze app.

### Schaalbaarheid

**Vue.js** werkt met een componentgebaseerde structuur, wat betekent dat we de app makkelijk kunnen uitbreiden zonder alles opnieuw te bouwen. We kunnen dus klein beginnen en de app later zonder problemen groter maken als dat nodig is.

### Onderhoudbaarheid

Omdat **Vue.js** met herbruikbare componenten werkt, kunnen we de code makkelijk onderhouden en snel updates doorvoeren. Dat maakt het beheer van de app veel eenvoudiger op de lange termijn.

### Prestaties

**ionic** maakt gebruik van Tree Shaking-technologie, die alleen de noodzakelijke delen van de applicatie laadt. Dit zorgt voor een lichte en snelle applicatie.

### Beveiliging

**Vue.js** biedt ingebouwde beveiligingsmaatregelen, zoals automatische bescherming tegen XSS-aanvallen door gegevens te coderen met de notatie `{{ }}`. Daarnaast kunnen we met Vue Router specifieke pagina's afschermen, zodat alleen geautoriseerde gebruikers toegang hebben.

## Conclusie

De keuze voor **Vue.js**, **ionic** en **Capacitor** biedt een flexibele, schaalbare en veilige oplossing voor het ontwikkelen van een PWA die aan alle eisen van dit project voldoet en een verbeterde gebruikerservaring biedt voor inspecteurs.