# Smart-Pantry

# Inhoudstafel

[Smart-Pantry 1](#_Toc207042988)

[Inhoudstafel 2](#_Toc207042989)

[Inleiding 3](#_Toc207042990)

[Project Overzicht 3](#_Toc207042991)

[Doelstellingen 3](#_Toc207042992)

[Architectuur 3](#_Toc207042993)

[Front-end 3](#_Toc207042994)

[Back-end 4](#_Toc207042995)

[Firestore structuur 4](#_Toc207042996)

[Realtime & PWA Functionaliteit 4](#_Toc207042997)

[Structuur 4](#_Toc207042998)

[Functionaliteiten 5](#_Toc207042999)

[Huidige functionaliteiten 5](#_Toc207043000)

[Toekomstige functionaliteiten 5](#_Toc207043001)

[Zelf-reflectie 6](#_Toc207043002)

## Inleiding

Dit project is een Progressive Web Application (PWA) die gebruikers helpt met het beheren van hun voorraad, het vinden van winkels en het instellen van herinneringen. De applicatie maakt gebruik van moderne webtechnologieën zoals React, Firebase en service workers om zowel online als offline bruikbaar te zijn.

De app combineert meerdere functionaliteiten: van voorraadbeheer tot locatiebepaling en notificaties. Hierdoor ontstaat een praktische tool die gebruikers ondersteunt bij dagelijkse taken zoals boodschappen doen en producten beheren.

## Project Overzicht

Het project bestaat uit een webapplicatie die werkt als een slimme boodschappen- en voorraadassistent. Gebruikers kunnen hun voorraad beheren, winkels op een kaart markeren, producten scannen met hun camera en pushmeldingen ontvangen wanneer het tijd is om een winkel te bezoeken of voorraad aan te vullen.

De toepassing is ontwikkeld met **React (Vite)** voor de frontend en **Firebase** voor authenticatie en databankfunctionaliteit. Daarnaast biedt de app **PWA-mogelijkheden** zoals offline gebruik, installatie op toestellen en pushmeldingen.

## Doelstellingen

De belangrijkste doelstellingen van dit project zijn:

* Een gebruiksvriendelijke PWA ontwikkelen die offline kan functioneren.
* Integratie van Firebase voor login, database en opslag.
* Voorraadbeheer aanbieden via een Pantry-pagina.
* Winkelbeheer via een interactieve kaart met markers en locatie.
* Push notificaties en herinneringen voor boodschappen.
* Camera- en barcodefunctionaliteit integreren voor snelle productherkenning.
* Een schaalbare en modulaire projectstructuur aanbieden die uitbreidbaar is.

## Architectuur

### Front-end

* **Framework:** React met Vite bundler.
* **Styling:** CSS modules en globale stijlen.
* **Componenten:** BarcodeScanner, CameraCapture, CurrentLocation, MapClickHandler, StoreMarker, Modal, enz.
* **Pages:** PantryPage, StoresPage, SettingsPage.
* **State management:** Context API (AuthContext) en custom hooks.

### Back-end

* **Firebase Authentication:** Google Sign-in en e-mail login.
* **Firebase Firestore:** opslag van voorraad, winkels en gebruikersdata.
* **Firebase Storage:** opslaan en ophalen van foto’s via imagestore.jsx.
* **Service Worker:** voor offline functionaliteit en caching.

### Firestore structuur

* **users**: bevat gegevens van gebruikers zoals email, naam en authenticatie-info.
* **pantry**: verzameling van voorraaditems, gestructureerd per gebruiker.
* **stores**: locaties van winkels met hun markers en metadata.
* **reminders**: pushmeldingen en herinneringen voor producten of winkelbezoeken.

## Realtime & PWA Functionaliteit

* **Realtime updates:** via Firestore onSnapshot(), wijzigingen worden onmiddellijk zichtbaar in de app.
* **Push notificaties:** via usePushNotifications.tsx en useStoreReminders.tsx.
* **Offline ondersteuning:** met service worker en offline.html.
* **Installatie als app:** dankzij manifest.json en iconen (192px, 256px, 384px, 512px).

## Structuur

De projectstructuur is als volgt opgebouwd:

public

---- icons/ # App iconen

---- manifest.json # PWA manifest

---- service-worker.js # Offline en caching

/src

----components/ # UI componenten (Camera, Scanner, Modals, Maps)

---- context/ # AuthContext en state management

---- css/ # Styling met modules en globals

---- hooks/ # Custom hooks (auth, geolocatie, notificaties)

---- lib/ # Firebase db, login en opslag

---- pages/ # PantryPage, StoresPage, SettingsPage

---- utils/ # Hulpfuncties (distance calculation)

---- App.jsx # Hoofdcomponent

---- main.jsx # Entry point

---- firebase.jsx # Firebase config

## Functionaliteiten

### Huidige functionaliteiten

* Inloggen met Google of e-mail via Firebase.
* Voorraad beheren in PantryPage (toevoegen, aanpassen, verwijderen).
* Camera en barcode scanner om producten toe te voegen.
* Winkelbeheer via StoresPage met interactieve kaart en markers.
* Pushmeldingen en reminders instellen.
* Offline werken met service worker.
* Installatie als PWA op toestel.

### Toekomstige functionaliteiten

* Accountbeheerpagina (profiel aanpassen, uitloggen, account verwijderen).
* Geavanceerd voorraadbeheer (categorieën, prioriteiten, houdbaarheidsdata).
* Multi-user functionaliteit (samen voorraad en winkels beheren).
* Drag & drop voor voorraad en winkelmarkers.
* Integratie met externe API’s voor productinformatie.

## Zelf-reflectie

Het ontwikkelen van dit project was een leerrijke ervaring. Ik heb gewerkt met verschillende nieuwe technologieën zoals service workers, PWA-functionaliteiten en de combinatie van React en Firebase. Vooral het integreren van de camera en barcode scanner was uitdagend, maar het gaf veel voldoening om dit correct werkend te krijgen. Ik heb geleerd hoe belangrijk een duidelijke projectstructuur is en hoe handig het is om met custom hooks en context te werken.

De grootste leermomenten waren het opzetten van Firebase authenticatie en het omgaan met real-time updates via Firestore. Ook het bouwen van een PWA met offline ondersteuning en push notificaties heeft mijn kennis sterk verbreed.