DoD bas

**📌 Algemene Criteria**

De code is volledig objectgeoriënteerd (OOP) geschreven in PHP.

PDO is gebruikt voor database-interactie (geen mysqli).

Voor elke entiteit (klant, artikel, verkooporder, etc.) is een volledige CRUD-functionaliteit aanwezig.

De applicatie bevat een extern CSS-bestand voor layout en styling.

Alle schermen bevatten:

- Het logo van het bedrijf en de volledige naam bovenaan.

- Een footer met het telefoonnummer 0800.11.11.216 en de melding: 'Gratis nummer, alleen bereikbaar vanuit Nederland.'

🧪 **Testcriteria**

Voor elke methode is een PHPUnit test aanwezig (unit test).

Voor elke afgeronde user story is een acceptatietest geschreven en succesvol uitgevoerd.

**📋 Functionele Criteria per Gebruikerstype**

Verkoper

Kan klanten toevoegen, aanpassen, verwijderen en zoeken op naam/postcode.

Kan verkooporders aanmaken, inzien, wijzigen, verwijderen en zoeken op id.

Kan artikelvoorraad controleren bij bestelling.

Kan klantgegevens bekijken.

Magazijnmeester

Kan artikelen toevoegen, bewerken, verwijderen, en alle artikelen tonen.

Kan zoeken op artikelID (must) en omschrijving (could).

Ziet overzicht met voorraad en locatie.

Magazijnmedewerker

Kan artikelen overzichtelijk bekijken met locatie.

Kan zoeken op artikelid en optioneel op omschrijving.

Kan verkooporderstatus bijwerken (naar picking/meegegeven aan bezorger).

Bezorger

Kan via klantID het adres van de klant vinden.

Kan verkooporderstatus bijwerken (naar afgeleverd).

Inkoper

Ziet overzicht van alle artikelen.

Kan zoeken op artikelnummer.

Ziet automatisch welke artikelen besteld moeten worden (lager dan minimum).

Kan inkooporders beheren (CRUD) en zoeken op id.

Kan leveranciers beheren (CRUD) en zoeken op id/naam.

Bestelhoeveelheid wordt automatisch berekend: max voorraad - huidige voorraad.

**🔐 Gebruikersinterface / Navigatie**

Bij het starten van de applicatie verschijnt een hoofdmenu.

Het menu toont enkel opties die passen bij de ingelogde rol (must).

Gebruikers kunnen inloggen en zien direct hun eigen werkomgeving (could have).

Elke gebruiker werkt via een gescheiden interface, gebaseerd op zijn rol.

**💾 Database en Backend**

Alle genoemde tabellen uit de datadictionary zijn correct geïmplementeerd en genormaliseerd.

De relaties (foreign keys) zijn correct gelegd waar nodig.

Er is een PDO databaseconnectieklasse aanwezig en wordt hergebruikt.

SQL-injectie is voorkomen (prepared statements).

**📊 Sprintadministratie**

Elke sprint bevat:

- Wireframe van de interface

- Classdiagram

- User story

- Acceptatietest

- (Eventueel) burndownchart

Code is individueel geschreven. Indien samenwerking: enkel op reviewniveau.

Code staat in een private repository gedeeld met de docent(en).