

Inhaltsverzeichnis

Pflichtenheft - Landwirtschaft Management System	2
Ausgangslage	2
Untersuchungsanliegen der individuellen Themenstellung	2
Zielsetzung & geplantes Ergebnis	2
Zielkriterien	2
Erweiterte Ziel- oder Wunschkriterien	2
Meilensteine	3
Erstes Mockup – Geplantes Design	3
Landing Page	3
Tierbestand	4
Fahrzeugbestand	4
Agrar Flächen Besitz	5
Futterbestand	5
Ernteertrag	6
Datenmodell	6

Pflichtenheft - Landwirtschaft Management System

Ausgangslage

Ziel ist es ein funktionsfähiges Landwirtschaft Management System zu bauen, um eine Übersicht über den Tierbestand, Fahrzeuge, Flächen, Futterbestand, Ernteertrag zu gewährleisten. Aufgeteilt wird das Projekt in Frontend und Backend. Über Docker Container soll die Anwendung gestartet werden können.

Untersuchungsanliegen der individuellen Themenstellung

Benutzern soll es ermöglicht werden einen Überblick über ihre Landwirtschaft zu haben. Der zeitliche Ernteertrag soll graphisch visualisiert werden.

Zielsetzung & geplantes Ergebnis

Dem Benutzer soll es ermöglicht werden folgende Daten in das Informationssystem hinzuzufügen, zu löschen, zu updaten oder anzuzeigen:




- Tierbestand
- Fahrzeuge
- Flächen
- Futterbestand
- Ernteertrag

Zielkriterien

Das Projekt wird in Frontend und Backend aufgegliedert. Beim Backend wird auf Python und Flask gesetzt und die CRUD-Operationen werden hier implementiert. Für das Frontend kommt HTML, CSS und JavaScript zum Einsatz. Bei der Datenbank wird auf PostgreSQL gesetzt (die Datenbank soll mittels Docker Container gestartet werden) - bei der Modellierung des Datenmodells auf SQLAlchemy.

Erweiterte Ziel- oder Wunschkriterien

Wünschenswert wäre die Integration einer Wettervorhersage der nächsten 3 Tage. Um dies zu erreichen, sollen notwendige Wetterinformationen wie z.B. Wetterbeschreibung, Min. Temperatur, Max. Temperatur von einer API (z.B. *weatherbit.io*) ermittelt werden. Die Wettervorhersage soll in „Kacheln“ angezeigt werden und über die Wetterinformationen soll je nach Wetter ein passendes Wettericon angezeigt werden. z.b.: Ein Icon einer Sonne, wenn das Wetter schön ist.

Stadt	Ort	PLZ
<div> Bewölkt Min. Temperatur: 10°C Max. Temperatur: 19°C</div>	<div> Sonnenschein Min. Temperatur: 18°C Max. Temperatur: 25°C</div>	<div> Regen Min. Temperatur: 12°C Max. Temperatur: 14°C</div>

Meilensteine

Datum	Meilenstein
16.02.2024	Entwicklungsumgebung - Container, Aufteilung: Frontend - Backend
01.03.2024	CRUD-Funktionalitäten des Tierbestands, der Fahrzeuge
15.03.2024	CRUD-Funktionalitäten des Futterbestands, des Ernteertrags
23.03.2024	CRUD-Funktionalitäten der Flächen
01.04.2024	Visualisierung des Ernteertrags
15.04.2024	Integration der Wettervorhersage

Erstes Mockup – Geplantes Design

Landing Page

Beim Starten der Anwendung werden dem Benutzer Auswahlmöglichkeiten gegeben, um den Tierbestand, Fahrzeugbestand, Flächenbestand, Futterbestand oder Ernteertrag anzuzeigen, updaten, löschen oder hinzufügen zu können. In den jeweiligen Auswahlmöglichkeiten kann der Benutzer Datensätze hinzufügen, löschen, ändern und anzeigen. Beim Ernteertrag soll ein Graph der vergangenen Ernten einen Überblick über die Zeit schaffen. Das „Platzhalterbild“ auf der Landing Page soll später durch die Wettervorhersage ersetzt werden.



Tierbestand

Fahrzeuge	Flächen	Futterbestand	Ernteertrag
Tierbestand anzeigen			
Ohrmarkennummer Tierart Geschlecht Geburtsdatum		Update	
Hinzufügen	Löschen		

Fahrzeugbestand

Tierbestand	Flächen	Futterbestand	Ernteertrag
Fahrzeuge anzeigen			
Marke Modell Baujahr		Update	
Hinzufügen	Löschen		

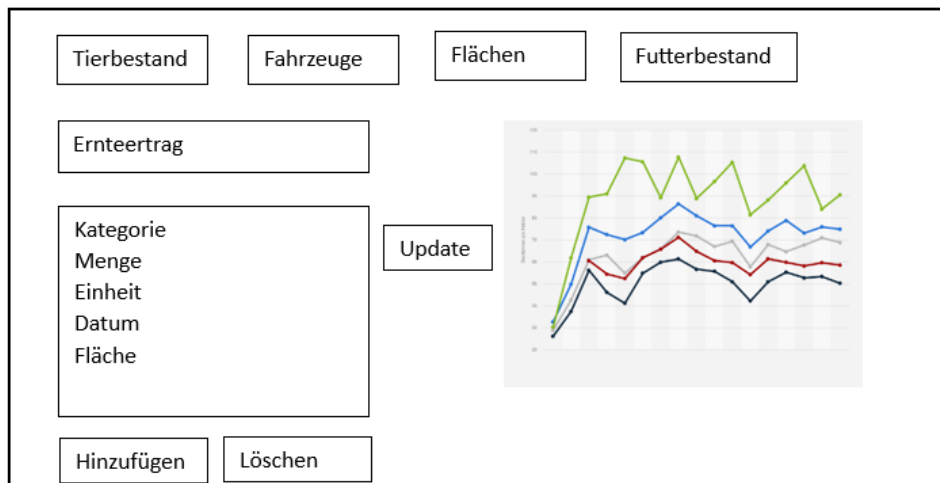
Agrar Flächen Besitz

Tierbestand	Fahrzeuge	Futterbestand	Ernteertrag
Eigenbesitz	Gepachtet		
Typ Fläche Einheit Besitz	Typ Fläche Einheit Besitz Personendaten Adressdaten		Update
Hinzufügen	Löschen	Hinzufügen	Löschen

Futterbestand

Tierbestand	Fahrzeuge	Flächen	Ernteertrag
Futterbestand anzeigen			
Kategorie Menge Einheit	Update		
Hinzufügen	Löschen		

Ernteertrag



Datenmodell

