Feature dark /light theme

The feature can synced with the browser / os mode

It lets the user switch the theme of the app between:

(Light Theme)

- **Bright background**
- **Dark text**
- **High visibility**
- **Feels clean and energetic**
- **Better in daylight**

(Dark Theme)

- **Dark background**
- **Light text**
- **Reduces eye strain**
- **Saves battery on OLED screens**
- **Better at night or low light**

Its for better user experience

Alert Generation page :

The goal is to:
Detects dangerous conditions early

Helps farmers/operators act fast

Saves crops from damage

Tracks historical issues for analysis

Makes the system "smart" and proactive

Functional requirement :

Whenever the measured value (sensor reading) goes below the minimum or above the maximum of some crop's min or max , the system automatically creates a new row in the Alerts table specify the rest .

What the Alert row Contains

ThresholdType

What type of threshold was violated

- **BelowMin**
- **AboveMax**

ReadingTypeId

Example: Temperature, Humidity, Soil pH
This tells **which type of reading** triggered the alert.

Zone

The specific zone/area where the issue happened.

Crop

Which crop is affected

Stage

Crop growth stage → seedling, vegetative, flowering, etc.

Value

The **actual measured value** that caused the alert.
You know exactly what the sensor reads.

Date

The specific date where the issue happened

Message

A human-readable explanation.
Example: *“Temperature above max limit: 38°C.”*

STATUS

- Not acknowledged
- acknowledged

Non - Functional requirement :

Performance

- The alert creation process must complete in **under 1 second** after a threshold violation is detected.

- The page must load all alerts in **less than 2 seconds** even with large data (10,000+ alerts).

Reliability

- No alert should be duplicated for the same reading.

Usability

- The page must clearly show: ThresholdType, ReadingType, Zone, Crop, Stage,date, Value, Message.

- The user should be able to acknowledge an alert with **one click**.

- UI should highlight critical alerts (e.g., icons/colors).

Security

- Only authorized users can view or acknowledge alerts.

- All operations (view, acknowledge) must be logged.

. Availability

- The alert feature must be operational **24/7** with minimal downtime.

- If the server is down, alerts must queue and process once the system is back.

Accuracy

- Alerts must use the **correct limits** based on Crop, Zone, Stage, and ReadingType.

- The system must avoid false positives or false negatives.

Nots:

Status controls filtering

The Status field is what the user uses to filter alerts on the page.
Examples of statuses:

- New
- In-Progress
- Acknowledged
- Resolved

This makes it easy for the user to see what needs attention.

Action controls status change

The **Action** column is **not a status** — it is the **button that triggers changing the status**.

Example:

- **Action = "Acknowledge"** → changes `Status` from **New** → **Acknowledged**
- **Action = "Mark as Resolved"** → changes `Status` from **Acknowledged** → **Resolved**

So:

- **Status = what the alert currently is**
- **Action = what the user can do next**