

Early Warning System - Backend Feature Documentation

Overview

This backend system supports:

- Secure teacher authentication via JWT
- Dynamic class-based attendance recording
- Dynamic incident reporting tied to students
- Clean collection separation per class (e.g., `classa_attendance`, `classa_incidents`)
- Scalable and token-driven architecture

In Scope

- Secure login using JWT tokens
- Fetching students based on teacher's assigned class
- Taking attendance once per day per teacher per class
- Auto-saving attendance into dynamic collections
- Reporting incidents for students in class
- Auto-filling teacher and student info
- Recording category and severity
- One-to-many relationships (teacher -> students, incidents)

Out of Scope

- Teachers managing multiple classes
- Student-side interaction

- Admin dashboards
- File/image uploads
- Escalation or email workflows

Authentication Flow

- JWT-based login
- Token used in Authorization header
- Token expires in 1 hour
- Used on: /attendance/students, /attendance/submit, /incident/report

Relationships

- Teacher -> Class: One-to-One
- Class -> Students: One-to-Many
- Teacher -> Attendance/Incidents: One-to-Many

Collections

Teachers (schoolteachers)

```
{  
  "full_name": "Masola Kabelo",  
  "email": "masola@example.com",  
  "password": "<hashed>",  
  "class_assigned": "Class A"  
}
```

Learners (learners_classa)

```
{  
  
  "student_id": "1234567890123",  
  
  "student_name": "Andile Ndlovu",  
  
  "dob": "2003-02-10",  
  
  "gender": "Male"  
  
}
```

Attendance (classa_attendance)

```
{  
  
  "session_id": "session_171...",  
  
  "teacher_name": "Masola Kabelo",  
  
  "students": [{ "student_id": "...", "status": "Present" }],  
  
  "attendance_date": "2025-05-16T..."  
  
}
```

Incidents (classa_incidents)

```
{  
  
  "student_id": "5678901234567",  
  
  "teacher_name": "Masola Kabelo",  
  
  "description": "Student threw a chair",  
  
  "category": "Disruption",  
  
  "severity": "High",  
  
  "reported_at": "2025-05-16T..."  
  
}
```

Logic Flows

Login: POST /teachers/login -> Get JWT token

Attendance:

- GET /attendance/students -> fetch class list
- POST /attendance/submit -> submit attendance

Incidents:

- Select student from class list
- Fill out incident form with category + severity
- POST /incident/report -> log to dynamic collection

Scalability

- Dynamic collections per class
- Centralized logic using token context
- Easy to extend with filters/dashboards later