



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the core backend architecture for **AgroData - Crop Yield Logger**. The service enables farmers to securely log and retrieve crop yield data, supporting user authentication and persistent storage using Node.js, Express, and MySQL.

1. System Components

Component	Description	Key Responsibilities
API Server	Express-based HTTP server	Handle API requests, route management
Auth Module	Authentication middleware	User login, JWT token validation
Data Access Layer	MySQL ORM/queries abstraction	CRUD operations on data models
Models	Data schema definitions	Represent Users, Yields

2. Class/Interface Overview

Class/Module	Relationships	Key Methods/Attributes
UserController	Uses Auth, UserModel	register() , login() , getProfile()
YieldController	Uses Auth, YieldModel	addYield() , getYields()
AuthMiddleware	Used by routes	verifyToken()
UserModel	Maps to users table	id , username , passwordHash
YieldModel	Maps to yields table	id , userId , cropType , amount , date

3. Data Structure Overview

Model	Fields
User	id (PK), username (unique), passwordHash
Yield	id (PK), userId (FK), cropType , amount , date

4. Algorithms / Logic

User Registration/Login:



```
// Registration
if (username not exists) {
  hash password
  store user
}

// Login
if (username exists && password matches hash) {
  generate JWT token
  return token
}
```

Yield Logging:

```
if (JWT valid) {
  validate input
  insert yield record with userId
}
```

Yield Retrieval:

```
if (JWT valid) {
  fetch yields where userId = current user
  return yield list
}
```

5. Error Handling

Scenario	Handling Approach
Invalid credentials	Return 401 Unauthorized
JWT missing/invalid/expired	Return 401 Unauthorized
DB connection/query failure	Return 500 Internal Server Error
Validation error (input)	Return 400 Bad Request
Resource not found (e.g. yields)	Return 404 Not Found

End of Document