**Table of Contents:**

1. [Introduction](#Introduction)
2. [Main Features](#MainFeatures)
3. [Project Architecture](#ProjectArchitecture)
4. [Technologies Used](#TechnologiesUsed)
5. [API Documentation](#APIDocumentation)

**Introduction:**

HopeConnect is a System Made to help people who want to donate or sponsor the orphan or provide assistance for orphaned children in Gaza after the war, the idea is to connect people who want to help whether donors, sponsors, or volunteers to children in need and orphanages The main goal of the app is to make sure donations reach the people who truly need them.

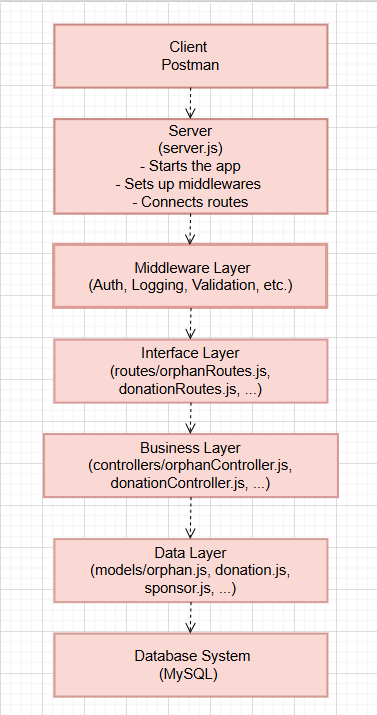
**Main Features:**

1. **Orphan Profiles & Sponsorships**: this feature includes a profile for each orphan with their basic info. People can sponsor them in different ways, and they'll get instant updates about their condition.
2. **Donation Management System**: users can easily donate different things, everything is organized by donation type, the payment process is smooth through the app, and they’ll get updates showing where their donations went.
3. **Volunteer & Service Matching**: volunteers can sign up to help with teaching or healthcare, orphanages ask for help, and the system connects them.
4. **Trust & Transparency**: Donors see how their donations are used, orphanages are verified to prevent scams, and donors can leave reviews.
5. **Emergency Support System**: People can help in emergencies and get notification.
6. **Logistics & Resource Distribution**: A system tracks deliveries in real-time and coordinates pickup and drop-off of material donation ) clothes and food (
7. **Revenue Model & Sustainability**: Small fees on donations cover costs, and there are partner with NGOs and charities.

**Project Architecture:**

We used Node.js because it’s suitable for building REST APIs, provides high performance, has many helpful libraries, and works well with MySQL.

We designed HopeConnect using a Layered Architecture approach to ensure clear separation of concerns, maintainability, and scalability. The main layers in the backend API as show:



**Technologies Used:**

* Node.js
* Express.js
* MySQL
* bcrypt
* dotenv
* nodemon
* body-parser
* Axios
* Git
* Postman
* npm
* nodemailer

**API Documentation:**

Here’s the full API documentation on Postmanhttps : https://documenter.getpostman.com/view/42991095/2sB2qaiMjh