Overview

This project appears to be a full-stack application with a React frontend and multiple NestJS backend microservices. The system includes modules for feedback, requests, and user management, connected to a MongoDB database.

Backend Services

Core Structure

Each NestJS microservice follows a similar pattern:

* **AppModule**: Main module that imports necessary dependencies
* **AppController**: Handles HTTP requests
* **AppService**: Contains business logic
* **Main bootstrap**: Starts the application

Microservices

1. **Main Service** (port 3002)
   * Mongoose configuration for MongoDB
   * Feedback module integration
2. **Request Service** (port 3001)
   * Environment configuration using ConfigModule
   * Request module functionality
3. **User Service** (port 3003)
   * Environment configuration
   * User management module

Common Features

* MongoDB integration via Mongoose
* Environment variable support
* Basic "Hello World" endpoint for health checks

Frontend

The React application provides a web interface with:

Routing

* Home page
* Request management
* User management
* Assignments
* Feedback

UI Components

* Navigation bar
* Page-specific components

Styling

* CSS animations and responsive design
* Modern, clean layout

Testing

Both frontend and backend include test suites:

* **React**: Jest with React Testing Library
* **NestJS**: Built-in testing utilities

Environment Requirements

The project requires:

* Node.js environment
* MongoDB connection (configured via MONGO\_URI)
* React for frontend
* NestJS for backend services

Setup Instructions

1. Install dependencies for each service and frontend
2. Create .env files with required variables (MONGO\_URI, PORTs)
3. Start each microservice individually
4. Launch the React application

Architecture

The system follows a microservice architecture with:

* Separate services for different domains
* Frontend aggregating functionality
* Shared MongoDB database (with separate collections per service)

This modular approach allows for independent development and scaling of different application components.