User Feedback System - Architecture & Flow

Project Architecture Overview

The User Feedback System is a full-stack web application with a clear separation of concerns across three layers:
React (Frontend) Express API (Backend) MongoDB (Database)

Technologies Used

Frontend: React, Axios
Backend: Node.js, Express
Database: MongoDB, Mongoose
Environment: dotenv

Application Flow

1. User opens the web app in the browser (localhost:3000)
2. Fills out the feedback form (name, email, feedback text, category)
3. Form submits data using Axios via a POST /feedback API request
4. Backend API receives the request, validates the input
5. Saves the feedback to MongoDB using the Mongoose model
6. Optional: Dashboard fetches feedback using GET /feedback to display feedbacks on-screen

API Endpoints

POST /feedback: Submits user feedback
GET /feedback: Retrieves all feedbacks

Folder Structure

```
user-feedback-system/
 backend/
   models/
   routes/
   app.js
   .env
 frontend/
   src/
      components/
        FeedbackForm.js
        FeedbackDashboard.js (optional)
      App.js
 README.md
```

Data Flow

[React Form] (POST /feedback via Axios) [Express Backend Route] [Mongoose Model] [MongoDB Database]
[React Dashboard] (GET /feedback) [Express Route] [MongoDB via Mongoose]

Security and Clean Code Practices

User Feedback System - Architecture & Flow

- Data validation on backend
- .env file for hiding sensitive data
- CORS enabled for frontend-backend communication
- Modular file structure (routes, models, components separated)

Possible Future Improvements

- Add filter/sort options in dashboard
- Add admin panel with login
- Deploy frontend (Netlify) and backend (Render)
- Add feedback analytics or charts