COLLEGE CODE: 8203

COLLEGE: AVC COLLEGE OF ENGINEERING

DEPARTMENT: INFORMATION TECHNOLOGY

STUDENT NM-ID: A8497BAAFDD3B8D816E4BACCB59D2039

ROLL NO: 23IT119

DATE:15/09/2025

Completed the project named as Phase II

TECHNOLOGY PROJECT NAME: Feedback Collection System

SUBMITTED BY,

NAME: YASARDEEN.P

MOBILE NO: 6369675068

TECH STACK SELECTION:

FRONTEND:

- HTML, CSS, JavaScript (for simple UI)
- (Optional Enhancement: React for dynamic UI, Bootstrap/Tailwind for styling)

BACKEND:

- Node.js with Express.js (REST API framework)

DATABASE:

- MongoDB (NoSQL database for flexible schema, timestamp storage, and feedback retrieval)

ADDITIONAL TOOLS:

- Nodemailer (email confirmation)
- Sentiment Analysis (npm sentiment package)
- Postman (API testing tool)

UI STRUCTURE / API SCHEMA DESIGN:

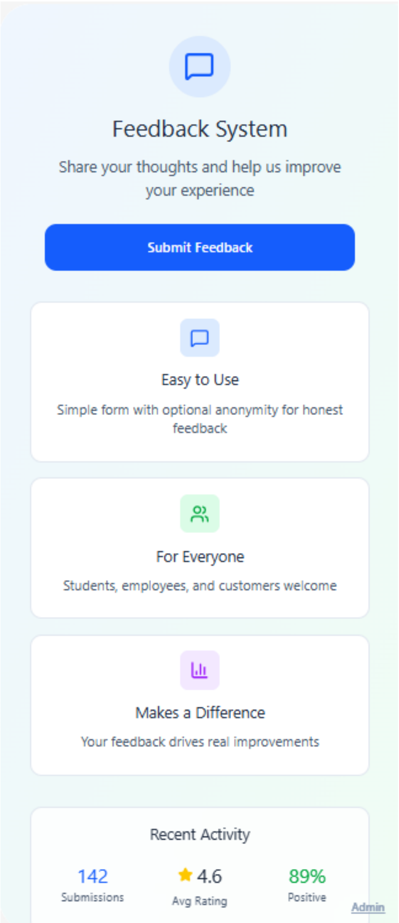
UI STRUCTURE:

Feedback Form Page:

1. input fields: Name, Email, Feedback, Rating.
2. Submit button

Admin Dashboard (future enhancement):

1. Table view of all feedback.
2. Filters for rating/date.



Feedback System

Share your thoughts and help us improve your experience

Submit Feedback

Easy to Use

Simple form with optional anonymity for honest feedback

For Everyone

Students, employees, and customers welcome

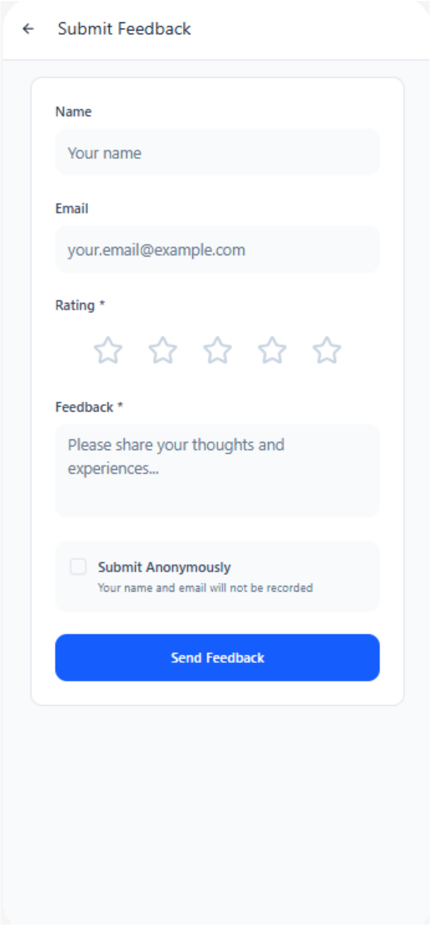
Makes a Difference

Your feedback drives real improvements

Recent Activity

142	★ 4.6	89%
Submissions	Avg Rating	Positive

Admin



← Submit Feedback

Name

Your name

Email

your.email@example.com

Rating *

☆ ☆ ☆ ☆ ☆

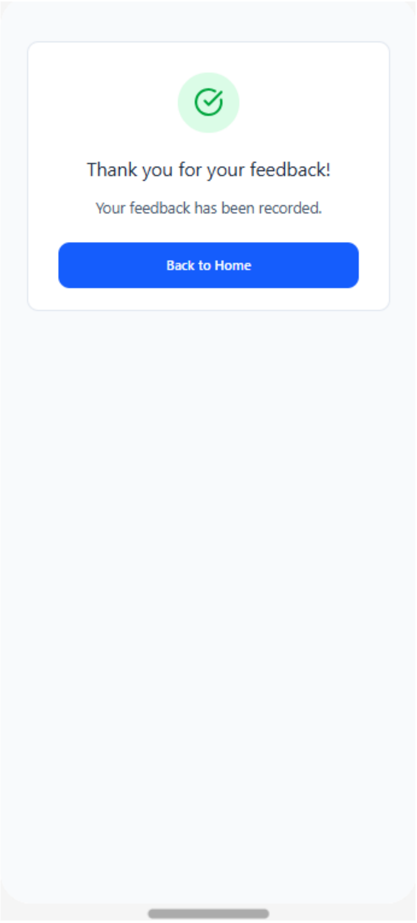
Feedback *

Please share your thoughts and experiences...

□ Submit Anonymously

Your name and email will not be recorded

Send Feedback



Thank you for your feedback!

Your feedback has been recorded.

Back to Home

API SCHEMA DESIGN (FEEDBACK DOCUMENT IN MONGODB):

```
{
  "name": "John Doe",
  "email": "john@example.com",
  "feedback": "Great service!",
  "rating": 5,
  "sentiment": "Positive",
  "createdAt": "2025-09-24T10:30:00Z"
}
```

DATA HANDLING APPROACH:

- ## FRONTEND → BACKEND:
  - Data submitted via form (POST request).
- ## BACKEND → DATABASE:
  - Input validated using Mongoose schema.
  - Feedback stored in MongoDB with timestamp.
- ## DATABASE → ADMIN API:
  - Admin can fetch all feedback (GET /feedback).
  - Filtering supported via query parameters (GET /feedback?rating=4).
- ## ENHANCEMENTS:
  - Sentiment analysis added automatically.
  - Email confirmation sent to user after submission.

COMPONENTS OF FEEDBACK COLLECTION SYSTEM:

1. FRONTEND (USER INTERFACE):

- Feedback Form → Fields: Name, Email, Feedback, Rating (1–5)
- Submit Button → Trigger POST request to backend
- Confirmation Message → Shows success/failure response
- Error Message Section → Displays validation errors (invalid email, missing fields, rating out of range)
- (Optional) Admin Dashboard → Table view of all feedback with filters (rating/date)

2. BACKEND (APPLICATION LAYER – NODE.JS + EXPRESS):

- Route Handlers
  - POST /feedback → Submit feedback and store in DB
  - GET /feedback → Fetch all feedback
  - GET /feedback?rating=5 → Filter feedback by rating
  - GET /feedback/:id → Get single feedback entry (optional)

- Controller Module → Handles validation, saving to DB, and sentiment analysis
- Error Handling Module → Returns user-friendly error messages

3. DATABASE (MONGODB):

- Feedback Collection
  - Fields: Name, Email, Feedback, Rating, Sentiment, Timestamp
  - Stores structured feedback in JSON-like documents
- Indexes
  - On createdAt (for sorting by date)
  - On rating (for quick filtering)

4. UTILITY & SUPPORT SERVICES:

- Sentiment Analysis (npm sentiment)
  - Analyzes text feedback → returns Positive/Negative score
- Email Service (Nodemailer)
  - Sends confirmation email to user after submission
- Validation (Mongoose + Frontend)
  - Ensures required fields, rating within 1–5, valid email

5. Admin & Reporting Module:

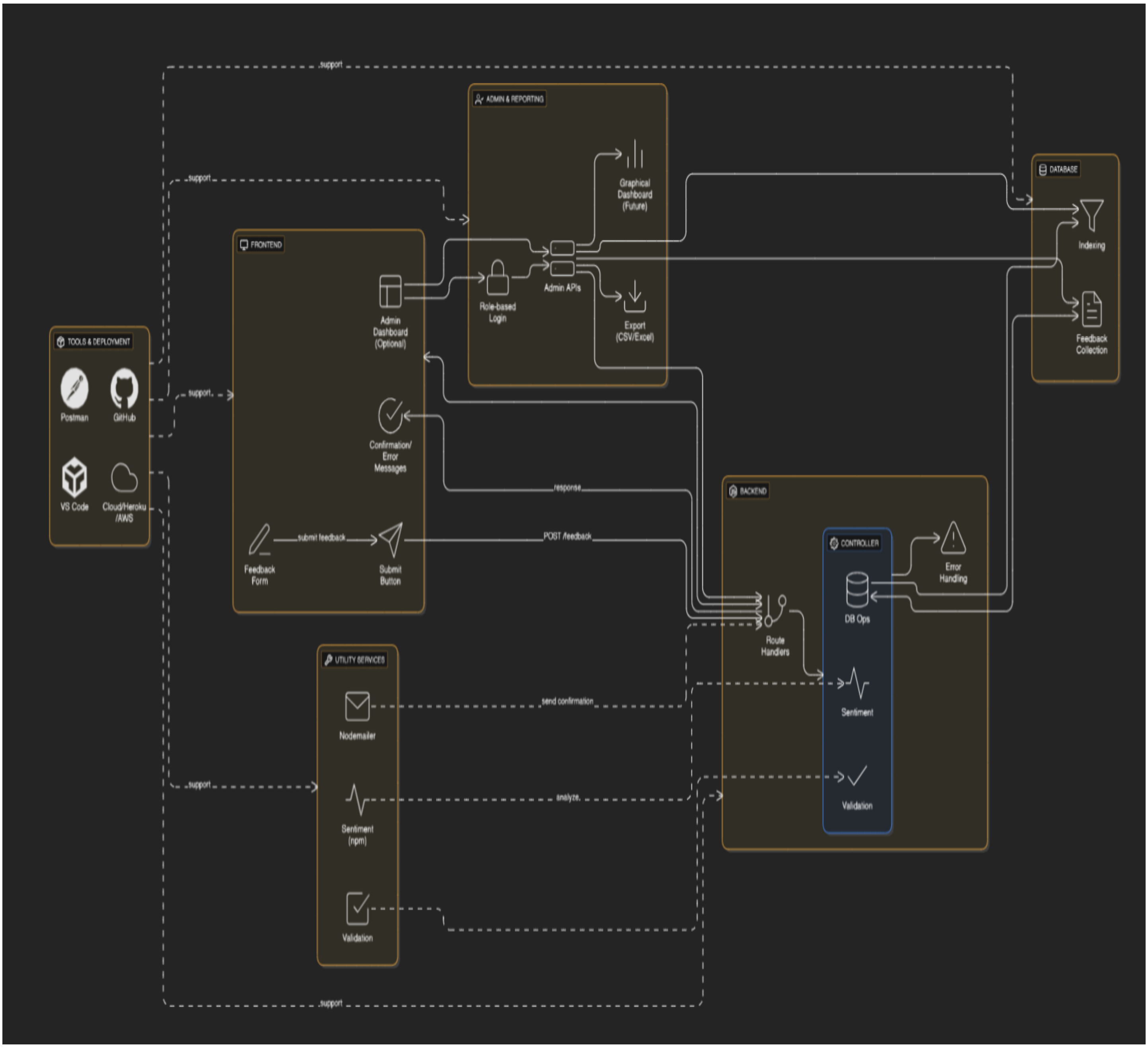
- Admin APIs
  - GET /feedback → Returns all feedback in JSON
  - Query filters (rating, date) for advanced search
- Future Enhancements
  - Graphical dashboard (charts for sentiment trends)
  - Export feedback as CSV/Excel
  - Role-based admin login for security

6. UTILITIES & TOOLS:

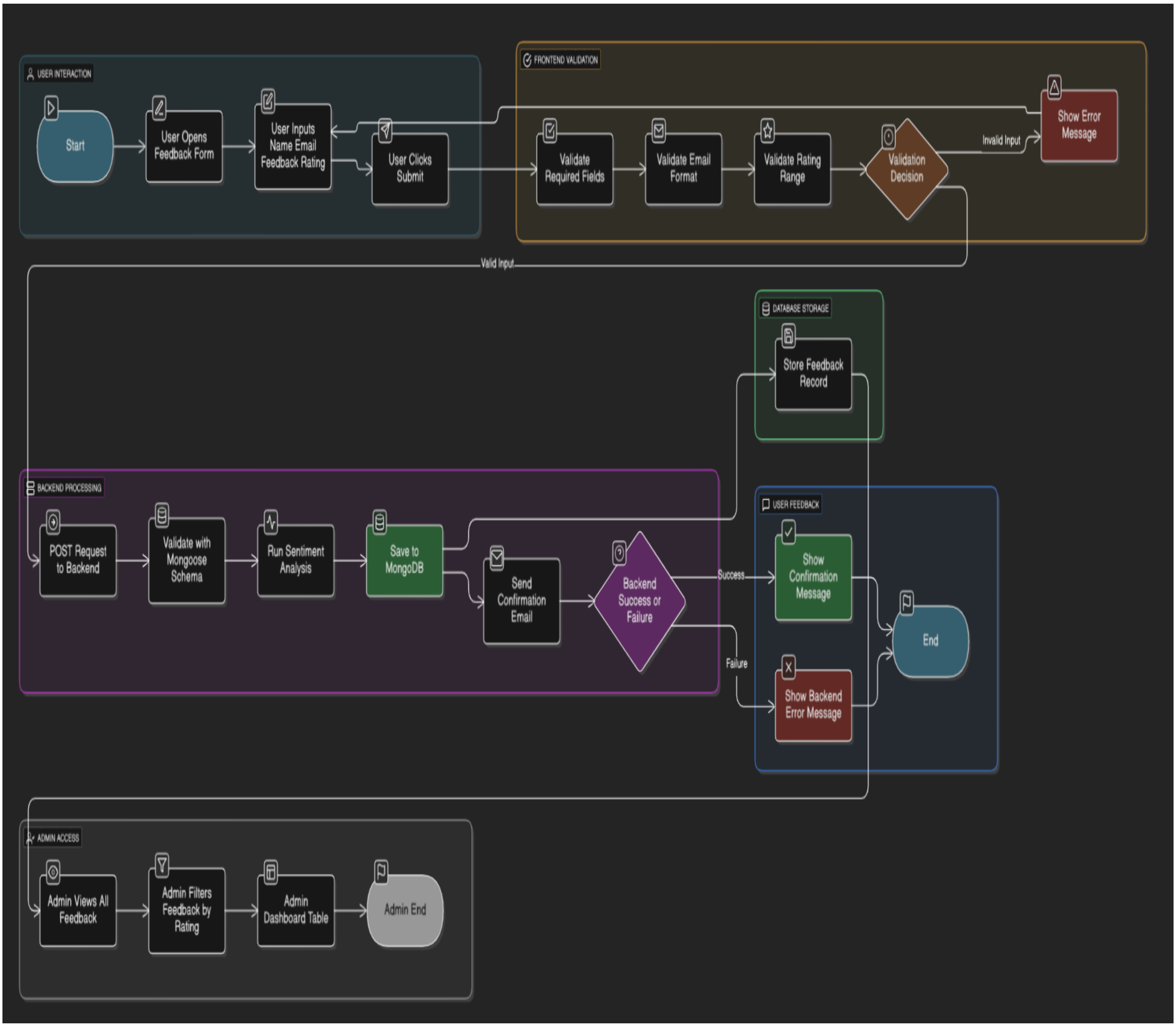
- Postman → API testing
- Git/GitHub → Version control & collaboration

- VS Code → Development environment
- Deployment → Localhost for testing, Cloud (Heroku/AWS) for production

MODULE DIAGRAM:



BASIC FLOW DIAGRAM:


USER INTERACTION
Start
User Opens Feedback Form
User Inputs Name Email Feedback Rating
User Clicks Submit
FRONTEND VALIDATION
Validate Required Fields
Validate Email Format
Validate Rating Range
Validation Decision
Invalid Input
Show Error Message
Valid Input
DATABASE STORAGE
Store Feedback Record
BACKEND PROCESSING
POST Request to Backend
Validate with Mongoose Schema
Run Sentiment Analysis
Save to MongoDB
Send Confirmation Email
Backend Success or Failure
Success
Failure
USER FEEDBACK
Show Confirmation Message
Show Backend Error Message
End
ADMIN ACCESS
Admin Views All Feedback
Admin Filters Feedback by Rating
Admin Dashboard Table
Admin End