COLLEGE CODE : 9222

COLLEGE NAME : THENI KAMMAVAR SANGAM COLLEGE OF TECHNOLOGY

DEPARTMENT : B.TECH(INFORMATION TECHNOLOGY)

STUDENT NM-ID : CEA656C79AEDCA7A7504402B590C6649

REG NO : 922223205029

DATE : 26.09.2025

Completed the project named as Phase_3_ TECHNOLOGY

PROJECT NAME : IBM-NJ- FEEDBACK COLLECTION SYSTEM

SUBMITTED BY,

NAME : NITHES KUMAR L

MOBILE : 9345983608

FEEDBACK COLLECTION SYSTEM

MVP IMPLEMENTATION

Project Setup :

Project Overview

A web-based Feedback Collection System that allows users to submit feedback, which can be stored, reviewed, and managed. Useful for websites, apps, events, or customer services.

Tech Stack

- Frontend: React.js / HTML, CSS, JavaScript
- Backend (optional): Node.js + Express (for storing feedback in a DB)
- Database: MongoDB / Firebase / LocalStorage (for local state)
- Version Control: Git + GitHub

Installation & Setup

```
# Clone the repo

git clone https://github.com/username/feedback-collection-system.git

cd feedback-collection-system

# Install dependencies

npm install

# Start development server

npm start
```

Core Features Implementation

Feedback Form

- User inputs:
  - Name
  - Email (optional)
  - Rating (1-5 stars or emojis)
  - Comments
- Validation for empty fields and proper formats.

Display Feedback

- List of submitted feedbacks.
- Show feedback details in cards or a table.

Edit/Delete Feedback

- Option to edit or remove submitted feedback.

Admin Panel (Optional)

- Admin login to view/export/manage feedback.
- Filters: Rating, Date, Keyword Search.

Data Storage (Local State / Database)

Option A: Local State (Frontend Only)

- Use React useState + useEffect.
- Store feedbacks in localStorage:

JS :

```
localStorage.setItem('feedbacks', JSON.stringify(feedbackList));
```

Option B: Database Storage (Full Stack)

- Use MongoDB or Firebase to store feedback remotely.
- Sample MongoDB schema:

JS :

```
{
  name: String,
  email: String,
  rating: Number,
  comment: String,
  date: { type: Date, default: Date.now }
}
```

Testing Core Features :

Manual Testing :

- Test form validation.
- Submit feedback and check localStorage/database.
- Edit and delete feedback.
- UI responsiveness across devices.

Unit Testing (Optional) :

- Use Jest or React Testing Library for:
  - Form input validations
  - API calls (mocking)
  - Component rendering

JS :

```
test('renders feedback form', () => {

  render(<FeedbackForm />);

  expect(screen.getByText(/submit feedback/i)).toBeInTheDocument();

});
```

Version Control (GitHub) :

Git Commands :

bash

```
git init

git add .

git commit -m "Initial commit"

git remote add origin https://github.com/username/feedback-collection-system.git

git push -u origin main
```

GitHub Practices :

- Create a new repo on GitHub.
- Use .gitignore for node_modules.
- Branches:
  - main – Stable production code
  - dev – Development branch
- Pull Requests for feature merges.
- Descriptive commit messages:

- feat: add feedback form component
- fix: validation error for rating input