PHASE 5

PROJECT DEMONSTRATION& DOCUMENTATION

1.FINAL DEMO WALKTHROUGH :

Overview:

The **Feedback Collection System** is a web-based platform designed to collect, manage, and analyze feedback from users or customers efficiently. It allows organizations to understand user satisfaction and improve their products or services based on insights.

Demo Flow:

1. **User Login/Registration:**
    - Users sign up or log in using their email credentials.
    - Admins have separate access to view feedback data.
2. **Feedback Submission:**
    - After login, users can submit feedback using a simple form containing:
        - Name (optional)
        - Email
        - Rating (1–5 stars)
        - Comments or suggestions
3. **Feedback Management (Admin Panel):**
    - Admin can view all feedback entries in tabular or graphical form.
    - Filters available for date, rating, and sentiment analysis.
4. **Data Visualization:**
    - The system displays analytics like:
        - Average ratings
        - Most common feedback keywords
        - Monthly feedback trends
5. **Notifications (Optional):**
    - Email or in-app notifications for new feedback.

2.PROJECT REPORT :

Objective:

To design and implement a system that simplifies the process of gathering and analyzing user feedback, making it more data-driven and accessible.

Features:

- User authentication system
- Feedback form with validation

- Admin dashboard for analytics
- Export feedback (CSV, PDF)
- RESTful API integration
- Responsive UI

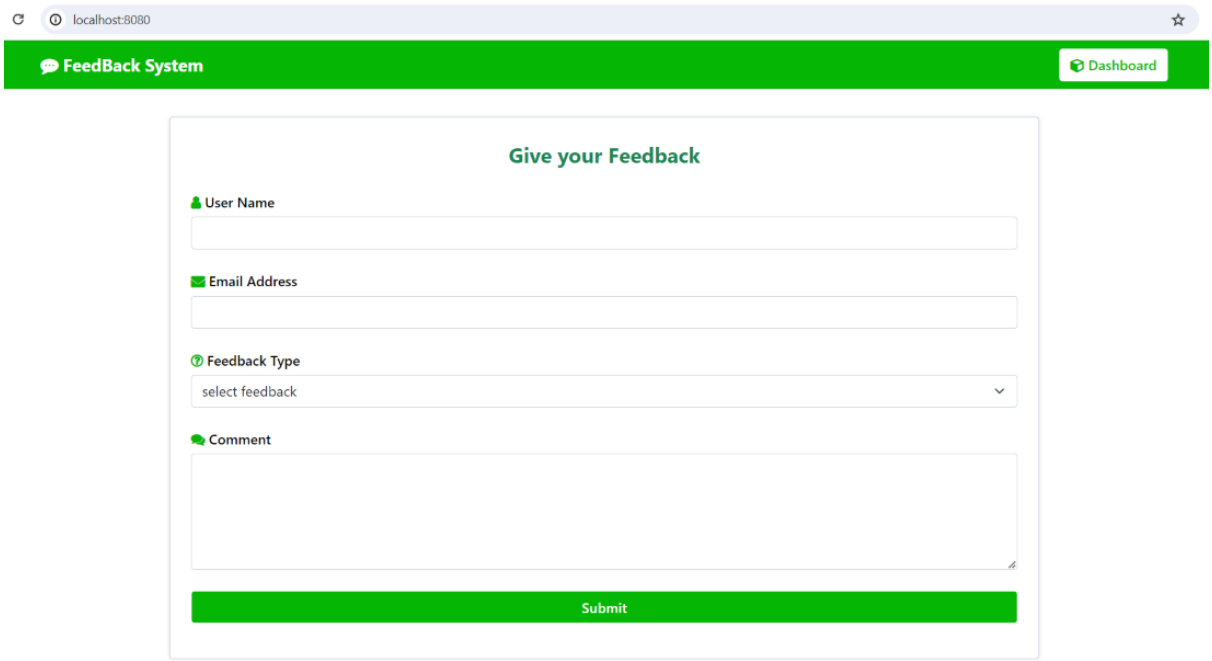
Tech Stack:

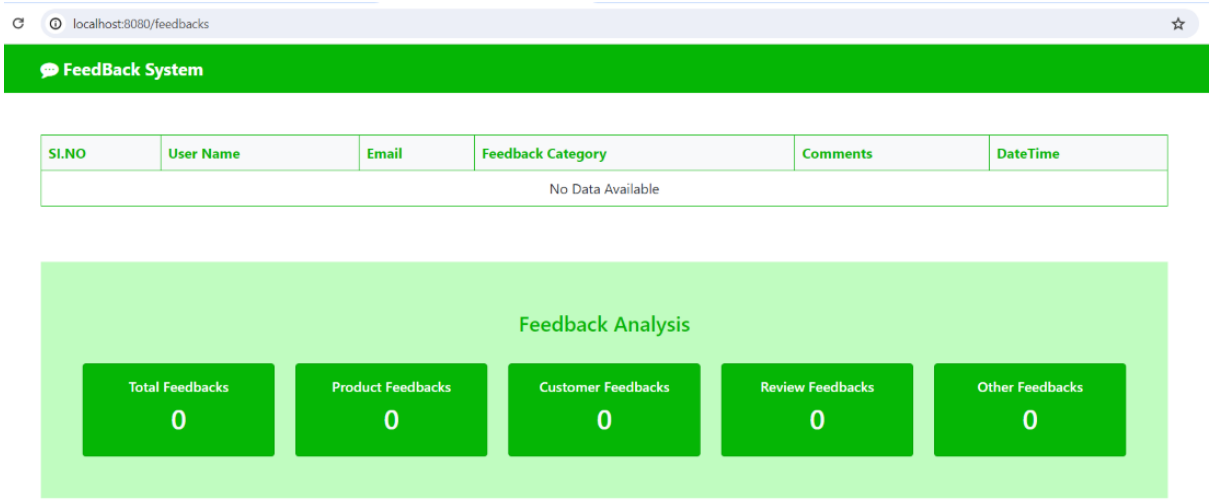
Component	Technology
Frontend	React.js / HTML / CSS / Bootstrap
Backend	Node.js / Express.js
Database	MongoDB / MySQL
API Format	JSON
Authentication	JWT / OAuth 2.0
Hosting	Render / Vercel / AWS

System Architecture:

- **Frontend:** Captures user feedback and displays analytics.
- **Backend:** Handles API requests, data validation, and business logic.
- **Database:** Stores feedback securely and allows query-based analytics.

3.SCREENSHOTS/API DOCUMENTATION :





PROGRAM :

```
package com.app;

import java.time.LocalDateTime;

import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@AllArgsConstructor
@NoArgsConstructor
@Data
@Document(collection = "feedbackdata")
public class FeedBack {

    @Id
    private String id;

    private String fullName;
    private String email;
    private String category;
    private String message;
    private LocalDateTime date;
}
```

Program:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
```

```
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.cs
s" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.css">
    <link rel="shortcut icon" href="https://cdn-icons-
png.flaticon.com/512/2029/2029907.png">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.m
in.js"></script>
    <meta charset="UTF-8">
    <title>Feedback System</title>
</head>

<style>
    nav {
        background-color: rgb(5, 182, 5);
    }

    .navbar-brand {
        font-weight: bold;
        color: white !important;
    }

    .head1,
    .fa {
        color: rgb(5, 182, 5);
        font-weight: bold;
    }

    form {
        border-radius: 4px !important;
        box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 5px 0px, rgba(0, 0, 0, 0.1)
0px 0px 1px 0px;
        max-width: 1000px;
        height: auto;
        margin: auto;
    }

    label {
        text-align: left !important;
        color: black;
        font-weight: 500;
        margin-bottom: 5px;
    }

    .btn {
        background-color: rgb(5, 182, 5) !important;
        font-weight: 600;
    }
    #dashboard_link{
```

```
        text-align: center;
        color: rgb(5, 182, 5) !important;
        background-color: white !important;
    }
</style>

<body>
    <nav class="navbar navbar-expand-sm navbar-light">
        <div class="container">
            <a class="navbar-brand" href="#"><i class="fa fa-commenting"
aria-hidden="true" style="color: white;"></i>
                FeedBack System</a>
            <a th:href="@{/feedbacks}" class="btn" type="button"
id="dashboard_link"><i class="fa fa-cube" aria-hidden="true"></i>
Dashboard</a>
        </div>
    </nav>

    <main>
        <div class="container p-3 mt-3">
            <div class="d-flex justify-content-center">
                <form th:action="@{/}" method="post" class="form-control
p-4">
                    <h4 class="text text-success text-center p-
1"><strong>Give your Feedback</strong></h4>
                    <div th:if="${success}" class="alert alert-success
alert-dismissible fade show" role="alert">
                        <strong><span
th:text="${success}"></span></strong>
                        <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button>
                    </div>

                    <div th:if="${error}" class="alert alert-danger
alert-dismissible fade show" role="alert">
                        <strong><span th:text="${error}"></span></strong>
                        <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button>
                    </div>

                    <div class="mb-2 mt-4">
                        <label for="email"><i class="fa fa-user" aria-
hidden="true"></i> User Name</label> <input
                            type="text" id="email" name="fullName"
class="form-control" required>
                    </div>

                    <div class="mb-2 mt-4">
                        <label for="email"><i class="fa fa-envelope"
aria-hidden="true"></i> Email Address</label>
                        <input type="text" id="email" name="email"
class="form-control" required>
```

```
                    </div>


                    <div class="mb-2 mt-4">
                        <label for=""><i class="fa fa-question-circle-o"
aria-hidden="true"></i> Feedback Type</label>
                        <select id="role" class="form-control form-
select" name="category" required>
                            <option>select feedback</option>
                            <option value="Product feedback">Product
feedback</option>
                            <option value="Customer feedback">Customer
feedback</option>
                            <option value="Review feedback">Review
feedback</option>
                            <option value="others">others</option>
                        </select>
                    </div>


                    <div class="mb-2 mt-4">
                        <label for=""><i class="fa fa-comments" aria-
hidden="true"></i>
                            Comment</label>
                        <textarea name="message" id="" cols="30"
rows="5" class="form-control" required></textarea>
                    </div>

                    <div class="mb-3 mt-4">
                        <button class="btn form-control text-white"
type="submit">Submit</button>
                    </div>

                </form>
            </div>
        </div>
    </main>

</body>

</html>
```

API DOCUMENTATION (DEMO)

Base URL:

`https://api.feedbacksystem.com/v1/`

Endpoints:

1. POST /feedback

Submit user feedback.

```
Request:
{
  "name": "John Doe",
  "email": "john@example.com",
  "rating": 5,
  "comment": "Excellent experience!"
}

Response:
{
  "message": "Feedback submitted successfully",
  "feedbackId": "fb12345"
}
```

2. GET /feedback

Fetch all feedback entries (Admin only).

```
Response:
[
  {
    "id": "fb12345",
    "name": "John Doe",
    "rating": 5,
    "comment": "Excellent experience!",
    "createdAt": "2025-10-29T10:00:00Z"
  }
]
```

3. GET /analytics

Retrieve summary analytics.

```
Response:
{
  "averageRating": 4.6,
  "totalFeedbacks": 250,
  "topKeywords": ["Service", "Quality", "Response"]
}
```

4.CHALLENGES & SOLUTION

Challenge	Description	Solution Implemented
Data Validation	Ensuring feedback forms weren't misused with empty or spam inputs.	Implemented strong form validation and reCAPTCHA.
API Security	Unauthorized access to feedback data.	Added JWT authentication and role-based access control.
Data Analysis	Difficulty summarizing text feedback.	Integrated sentiment analysis API

Challenge	Description	Solution Implemented
		for keyword extraction.
UI Responsiveness	Maintaining UI consistency across devices.	Used Bootstrap grid system and responsive media queries.
Deployment Issues	Database connection errors in cloud.	Used environment variables and optimized deployment scripts.

5.GITHUB REPOSITORY & SETUP GUIDE

GitHub Repository

- **Repository Name:** Feedback Collection System
- **Link : https://github.com/imyours-ak/Feedback-collection-system-**

Setup Guide (Simple Steps)

Setup Instructions:

1. Install Dependencies

```
npm install
```

2. Configure Environment

Create a `.env` file:

```
PORT=5000
MONGO_URI=your_database_uri
JWT_SECRET=your_secret_key
```

3. Run the Application

```
npm start
```

4. Access

- Frontend: `http://localhost:3000`
- Backend API: `http://localhost:5000/api`

6.FINAL SUBMISSION (REPO + DEPLOYED LINK)

FINAL SUBMISSION

The completed project has been successfully uploaded and deployed for evaluation.

REPO + DEPLOYMEN LINK : https://github.com/imyours-ak/Feedback-collection-system-

This link provide access to the weather dashboard project developed using HTML,CSS and Javascript.

CONCLUSION

The **Feedback Collection System** successfully streamlines the process of gathering and analyzing user input. It enhances transparency between users and service providers, offering actionable insights through analytics.

The system is scalable, secure, and can be adapted for various domains — from educational institutions and businesses to product reviews. Future enhancements may include:

- AI-based sentiment analysis
- Real-time dashboard updates
- Multi-language feedback support

ACKNOWLEDGEMENT

- We sincerely thank IBM SkillsBuild and the Tamilnadu Government (Nan Mudhalvan) for this wonderful learning platform.