**Customer Feedback Platform**

**Objective**

The objective of this project is to design and develop a lightweight, user-friendly feedback collection platform. This platform allows users to authenticate using Google OAuth, submit categorized feedback, and view previously submitted feedback grouped by category. It ensures a simple yet functional flow with persistent data storage using browser localStorage, without requiring an external database or API.

**System Overview**

The system is a React-based frontend application supported by a Node.js backend for handling Google OAuth authentication. The feedback functionality is handled entirely on the frontend using localStorage for persistence.

**Key Functional Components**

**1. User Authentication using Google OAuth**

* The user is authenticated using Google’s OAuth 2.0 mechanism.
* The backend initiates the authentication process by redirecting the user to the Google login page.
* On successful login, Google redirects back to the frontend at the route /login/success with user details.
* Authentication status is maintained using cookies or session identifiers handled by the backend.

**Purpose**:

* Avoids manual account creation and login.
* Provides a secure and streamlined user login process.

**2. Feedback Submission Form**

* Component Name: FeedbackForm.js
* Users are provided with a form that includes:
  + A dropdown field to select a feedback category.
  + A textarea to input detailed feedback or suggestions.
* The available feedback categories include:
  + Product Features
  + Product Pricing
  + Product Usability
* On form submission:
  + A feedback object is constructed containing a unique identifier (timestamp), selected category, and comment text.
  + This feedback object is appended to the list stored in localStorage under the key "feedbacks".
  + The user is redirected to the /feedback-list route to view all feedback.

**3. Feedback Storage and Persistence**

* Storage Mechanism: Browser localStorage.
* Key: "feedbacks"
* Format: An array of feedback objects with the following structure:

{

"id": 1712731082000,

"category": "Product Features",

"comment": "Add multi-language support"

}

* Existing feedback is preserved by retrieving the current list, appending the new entry, and saving it back to localStorage.

**4. Feedback Display**

* Component Name: FeedbackList.js
* Functionality:
  + Retrieves all feedback entries from localStorage.
  + Groups the feedback items based on their category using the Array.reduce() function.
  + Displays each category as a section with its associated feedback items listed below it.
* This allows users to visually navigate through different categories and understand the overall feedback distribution.

**5. Feedback Deletion**

* Each feedback entry is displayed with a delete button.
* On clicking the delete button:
  + The corresponding feedback entry is removed from the list.
  + The updated list is stored back in localStorage.
  + The component state is updated to reflect the changes immediately in the user interface.

**Component Flow and Routing**

The application uses React Router to manage navigation between views:

1. /login: Initiates the Google OAuth login process.
2. /login/success: Redirected after successful login, displays options to navigate to the form or list.
3. /feedback-form: Displays the feedback submission form.
4. /feedback-list: Displays categorized feedback entries.

**Data Flow Summary**

1. The user logs in using Google OAuth.
2. After authentication, the user accesses the feedback form.
3. Upon submission, feedback is stored in localStorage.
4. The user is redirected to the feedback list page where feedback is grouped and displayed.
5. The user may delete any feedback entry, which updates the stored data and the display in real-time.

**Design Considerations**

* The system avoids backend storage to maintain simplicity and focus on frontend behavior.
* Grouping feedback by category enhances readability and usability.
* Feedback storage is persistent across sessions due to the use of localStorage.
* Google OAuth ensures secure and convenient user authentication.