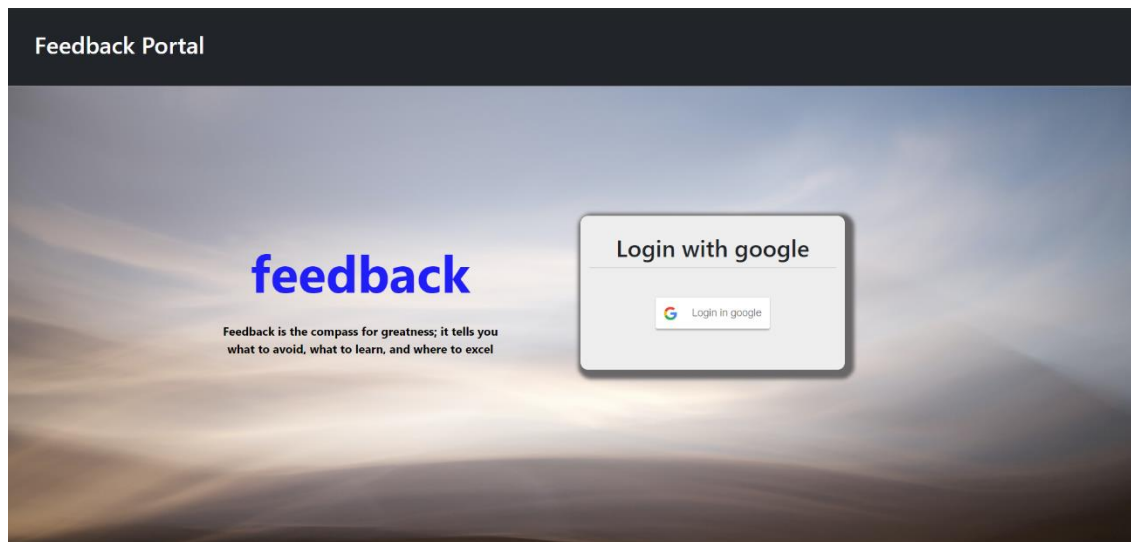


Feedback Portal Design Document

Introduction- Welcome to our Feedback Portal! Enjoy a seamless login experience with Google authentication. Once logged in, explore feedback effortlessly categorized into All, Feature, Pricing, and Usability. Share your valuable insights by clicking "Add Feedback," actively participating in our improvement process. The portal offers a user-friendly interface for efficient navigation. After providing feedback, securely log out using the designated button.

UI design-

1. **Login Page-** I provided login service by google auth using Google Authentication API. With just a few clicks, you can log in using your Google credentials, ensuring a seamless and trusted authentication experience. Once logged in, you will be directed to home page where you can explore categorized feedback in All, Feature, Pricing, and Usability sections.

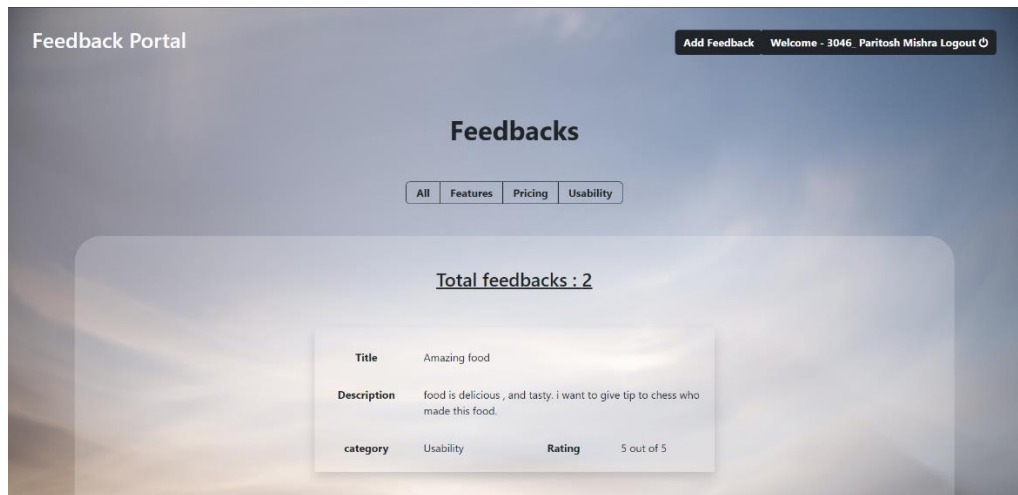


2. **Feedback Page (home page)** - Once you successfully logged in you will be directed to the feedback page. Where I provided feedbacks in **four categories**. You can explore through feedback categories effortlessly using the designated buttons—All, Feature, Pricing, and Usability. Each category offers a curated selection of feedback for your convenience.

In right side above I provided two buttons **Addfeedback** and **logout**.

- Addfeedback for adding new feedback or ideas.
- Logout for logged out from the portal.

To show feedbacks /getreviews Api have been called from the backend nodejs service with query parameter for each category.



Api endpoints used –

- Get category all reviews - "http://localhost:5556/getreviews?category=all"
- Get category feature reviews - "http://localhost:5556/getreviews?category=Feature"
- Get category Pricing reviews - "http://localhost:5556/getreviews?category=Pricing"
- Get category Usability reviews- <http://localhost:5556/getreviews?category=Usability>

3. **Addfeedback** - When you click on Addfeedback button you will be directed to addfeedback page that have a form with fields Title, Description, categories (Product feature, Product Pricing, Product Usability) & Rating. **I have been used validation in this form title , description and categories field required to submit the form.**

While submitting the form /savereview Api have been called from the backend nodejs.

Api endpoints used –

- Url = "http://localhost:5556/savereview"
- Method: Post
- Body: { name, topics, description, author, rating }
- author field get from localStorage.getItem("fullname");

Backend Services –

1. /savereview –

- Url = "http://localhost:5556/savereview"
- Method: Post
- Body: { name(String), topics(String), description(String), author(String), rating(Integer) }
- author field get from localStorage.getItem("fullname");

Getting topic_idx from Frill.co –

- **Description / Execution flow –**
 - Calling Get Api endpoint - "<https://api.frill.co/v1/topics>"
 - Also we are passing **Api Key** Provided by frill.co by sending **Bearer Authorization** header with the request.
 - Frill.co sends json object containing topics in json array. so we are traversing all the topics from response and extracting desired topic's topic_idx which we'll need to pass with the request body while sending POST request.

Posting data to Frill.co –

- **Description / Execution flow –**
 - Calling Api endpoint - "<https://api.frill.co/v1/ideas>"
Method: Post
Body: {name, description, author_idx, topic_idx}
 - Also we are passing **Api Key** Provided by frill.co by sending **Bearer Authorization** header with the request.
 - Before posting we are attaching the rating field that we get from frontend request object to the description field because as per the documentation from frill.co there is no rating field exists in the request body of /ideas end point . we are creating below json object that contain all the required fields and sending it as request body.
{ name, description, author_idx, topic_idx}
Example:

```
{
  "name": "Test Title",
  "description": "Test Description rating:2",
  "author_idx": "follower_67dzp46j",
  "topic_idx": ["topic_jv1oryjd"]
}
```

2. /getreviews Api –

- Url : "http://localhost:5556/getreviews "
- Method: get
- Query parameter: category values – [all, Feature, Pricing, Usability]

Getting data from Frill.co –

- **Description / Execution flow –**
 - Calling Get Api endpoint - "<https://api.frill.co/v1/ideas>"
Query parameter- sortBy=created_at
 - Also we are passing **Api Key** Provided by frill.co by sending **Bearer Authorization** header with the request.
 - Frill.co sends json object containing ideas in json array. so we are extracting each idea from it and generating response object with required fields then sending it to the frontend. In each idea we are extracting rating from description field because it was stored in the description field while posting data to frill.co endpoint.

Setup Details:

- Clone the repository <link> to a directory.
- Generate google OAuth Key and replace it in the login.js file in variable oAuthKey.
- Raise a request for frill.co API key using the "Request API Access" button on the page <https://paritosh.frill.co/settings/company/api>, you'll get the API key on your mail and it may take 1-2 days. Once you get the API key, replace it in the Api.js file in clientId variable.
- Start the react server by going to the directory of the feedback_portal > reactapp and run 'npm start'
- Start the node js server by going to the directory of feedback_portal > nodeapp and run 'nodemon Api'. Here Api is the name of the node js service file.

Reference:

<https://developers.frill.co/api/authentication>

<https://developers.frill.co/api/reference/ideas>

<https://developers.frill.co/api/reference/topics>

Challenges:

- **Authorization Key from Frill.co:** On the documentation page of Authentication (<https://developers.frill.co/api/authentication>) it is mentioned that API key will be present on Dashboard, it was not clear that user has to request for it and it can take more than a day to get the Key for use.
- **Request Body for Post API request:** On the documentation page of /ideas endpoint (<https://developers.frill.co/api/reference/ideas>) the format of request body for POST method is not given hence it was difficult to find out the required parameters for the POST request to be executed successfully. I used Postman tool to try out request with different parameters and found the correct request body.
- **Handling of Rating:** According to the documentation there is no field in request body for sending the rating given by the user, to handle that I have appended the rating value to the description string and while populating the data in the UI, I'm extracting that rating from the description.
- **Author_idx field value:** The author_idx field in POST request body of /ideas endpoint is a mandatory field and this field is used to identify the user uniquely on the frill.co. This value is not given in the dashboard or profile page. Since we need the value of this field to send POST request I have taken this value from the GET response of /ideas endpoint and hardcoded it in the code so that we can pass this for each Add Feedback request to frill.co.