In this task, I have implemented a category rating system using Vue.js for the frontend. Here’s a brief explanation of the approach taken, along with any trade-offs and assumptions made:

#### **1. Frontend Implementation**

Components:

* CategoryRatingDisplay: This is the main component that displays the category rating system. It fetches the categories from a JSON file and renders them using the **RatingStars** component. It also includes buttons to submit and reset ratings.
* RatingStars: This component renders the stars for each category using the **RatingSingleStar** component. It handles the logic for displaying the correct number of filled stars based on the rating.
* RatingSingleStar: This component represents a single star. It handles user interactions such as clicking and hovering, updating the store with the corresponding rating.

Composable:

* useCategoryRatings: This composable function fetches the categories data from a JSON file.

Store:

* CategoryRatingStore: This Pinia store manages the state of the categories, including fetching categories, setting ratings, handling mouse hover, submitting, and resetting ratings.

Data:

* data.json: Contains the initial categories and their ratings.

#### **2. End-to-End Testing**

Cypress Tests:

* categoryRating.cy.ts: Contains end-to-end tests for the category rating feature. It tests various scenarios such as opening the root URL, checking the category stars, clicking stars to set ratings, and the validation for submitting without selecting a rating.

#### **3. Unit Testing**

Vitest Tests:

* categoryRating.spec.ts: Contains unit tests for the CategoryRatingStore. It mocks the useCategoryRatings composable and tests the store's functionality, including initializing categories, setting ratings, handling hover, resetting ratings, and submitting ratings.

### 

### **Trade-offs and Assumptions**

Trade-offs:

1. Fetching categories from a local JSON file is suitable for development but might need to be replaced with an API call in production.

Assumptions:

1. Static Data: It is assumed that the categories data does not change frequently, hence fetching from a static JSON file.
2. Simple Rating System: The rating system is assumed to be simple, with no advanced features like dynamic categories.
3. The implementation assumes the use of Vue 3 and the Composition API, along with Pinia for state management.