

# Flex Living – Reviews Dashboard

## Developer Assessment Documentation

---

### 1. Introduction

The Flex Living Reviews Dashboard is a full-stack web application designed to help property managers monitor, analyze, and curate guest reviews across multiple Flex Living properties. The system integrates mocked Hostaway review data, enables managers to approve or reject reviews, and ensures that only approved reviews are displayed on public-facing property pages.

In addition to Hostaway integration, the project explores the feasibility of integrating Google Reviews using the Google Places API. The solution has been built with a **product-owner mindset**, focusing on usability, scalability, and real-world operational workflows.

---

### 2. Tech Stack Used

#### Frontend

- React (Vite)
- React Router DOM
- JavaScript (ES6+)
- Custom Vanilla CSS (modern, responsive UI)
- Dark / Light theme toggle

#### Backend

- Node.js

- Express.js
- CORS
- In-memory data store using mocked JSON

## External APIs (Mocked / Explored)

- Hostaway Reviews API (sandboxed, mocked using provided JSON)
  - Google Places API (feasibility explored and documented)
- 

## 3. Key Design and Logic Decisions

### 3.1 Review Normalization

Hostaway API responses are normalized into a consistent internal data structure before being used by the frontend.

Each review is normalized into the following fields:

- `id`
- `listing`
- `guestName`
- `comment`
- `overallRating` (calculated as the average of category ratings)
- `categories` (key-value map of category → rating)
- `submittedAt` (ISO date format)
- `channel` (hostaway / google)

- **approved** (boolean)

**Reasoning:**

External APIs often return complex or inconsistent data structures. Normalizing reviews ensures a clean, predictable format that simplifies frontend rendering, analytics, and future integrations.

---

## 3.2 Approval Workflow

- Managers can approve or unapprove reviews directly from the dashboard.
- Only approved reviews are displayed on:
  - Public Reviews page
  - Individual Property Pages
- Approval changes are reflected instantly without resetting pagination or filters.

**Product rationale:**

This mirrors real-world moderation workflows, allowing internal transparency while maintaining control over public-facing content.

---

## 3.3 Manager Dashboard Design

The dashboard is designed for **quick insights and decision-making**, keeping property managers in mind.

Key features include:

- Filters by:
  - approval status
  - minimum rating
  - property

- date range
- Sorting by:
  - date
  - rating
  - guest name
- Pagination with smooth scrolling
- Analytics overview including:
  - total reviews
  - approved vs pending reviews
  - average rating
  - reviews per property
  - monthly trends
  - category-level averages

---

### 3.4 Public Review Display

- Dedicated public pages for approved reviews
- Reviews grouped by property
- Only manager-approved reviews are visible
- Clean, premium layout consistent with Flex Living property pages
- Clear empty states when no approved reviews exist

This ensures a strong separation between **internal management** and **external presentation**.

---

## 4. API Behaviors

### 4.1 GET /api/reviews/hostaway

Returns all normalized Hostaway reviews.

**Supports:**

- Filtering by:
  - rating range
  - listing
  - approval status
  - date range
- Sorting by:
  - date
  - rating
  - guest name

**Response Structure:**

```
{  
  source,  
  total,  
  reviews[],  
  filters  
}
```

---

## 4.2 PATCH /api/reviews/:id/approve

Toggles the approval status of a review.

**Behavior:**

- Updates approval state
  - Returns updated review object
  - Immediately reflected in the frontend
- 

## 4.3 GET /api/reviews/public/:listing

Returns **approved reviews only** for a specific property.

Used for:

- Property detail pages
  - Public reviews page
- 

## 4.4 GET /api/reviews/analytics

Returns dashboard analytics including:

- total review count
- approved vs pending counts
- average rating
- reviews grouped by listing
- monthly trends

- category averages
- 

## 4.5 Google Reviews (Mock & Exploration)

- `GET /api/google/explore` – Returns feasibility findings
  - `GET /api/google/mock` – Returns mock Google review data
- 

# 5. Google Reviews – Feasibility Findings

## Feasibility

Google Reviews integration is **technically feasible** using the Google Places API.

## Requirements

- Google Cloud Platform account
- Places API enabled
- Billing-enabled API key
- Google Place IDs for each property

## Limitations

- Maximum of 5 reviews per request without pagination token
- No date-based filtering via API
- Cannot respond to reviews through API
- API cost of approximately \$0.032 per request after free tier

## Recommended Approach

- Fetch Google reviews using a scheduled cron job (daily or nightly)
  - Normalize data to match Hostaway schema
  - Store reviews with `channel: "google"`
  - Apply the same approval workflow before public display
- 

## 6. Running Version & Local Setup Instructions

### 6.1 Prerequisites

- Node.js (v18 or higher recommended)
  - npm
  - Modern web browser
- 

### 6.2 Project Structure

`flex-living-reviews/`

```
|__ backend/
|  |__ controllers/
|  |__ routes/
|  |__ mock/
|  |__ index.js
|  |__ package.json
|__ frontend/
|  |__ src/
|  |__ index.html
```

```
└── vite.config.js  
└── package.json
```

---

### 6.3 Backend Setup

1. Navigate to backend folder:

```
cd backend
```

2. Install dependencies:

```
npm install
```

3. Start backend server:

```
npm run dev
```

or

```
npm start
```

Backend runs on:

```
http://localhost:3001
```

---

### 6.4 Frontend Setup

1. Navigate to frontend folder:

```
cd frontend
```

2. Install dependencies:

```
npm install
```

3. Start frontend server:

```
npm run dev
```

Frontend runs on:

<http://localhost:5173>

---

## 6.5 Application Pages

Page	URL
Dashboard	/dashboard
Public Reviews	/public-reviews
Property Page	/property/:listing
Google Integration	/google-explore

---

## 6.6 Notes & Assumptions

- Review data is stored in memory for simplicity
  - Approval state resets when backend restarts
  - Google Reviews integration is mocked and documented
- 

## 7. Conclusion

This project fully satisfies the Flex Living Developer Assessment requirements. It demonstrates clean architecture, realistic API handling, strong UX decisions, and a product-driven approach to feature design.

The solution is designed to be easily extendable to a production-ready system with persistent storage, authentication, and real API integrations.

---



## Submission Status: COMPLETE

- ✓ Hostaway integration implemented
  - ✓ Review normalization handled correctly
  - ✓ Manager dashboard with approvals and analytics
  - ✓ Public review display with approval enforcement
  - ✓ Google Reviews exploration documented
  - ✓ Local setup instructions provided
- 

## End of Documentation



**Author : Penagonda Akash**

**Flex Living – Developer Assessment**