Frontend Handover Document

Project Name: Kanban Board

Name: Juan Diego Dumez

Version/Branch: dev-dumez

1. Project Overview

This is the frontend for the Kanban Board system. It is a React + Vite + Tailwind CSS project structured to support multiple user roles: Admin, Project Manager (PM), and Volunteer. The application includes login, password reset, success messages, and error handling pages.

2. Tech Stack

- **Framework**: React (Vite-based build)
- **Styling**: Tailwind CSS + custom styles (index.css, App.css)
- **Routing**: React Router
- **CI/CD**: GitLab CI (via .gitlab-ci.yml)
- **Linting**: ESLint
- **PostCSS**: Tailwind integrated via postcss.config.js

3.How to set it up locally

To run the project locally, start by cloning the repo and installing the dependencies. Once that's done, you'll need to configure the environment and tweak the API URLs to point to your local backend. The authentication flow is based on JWT, but because the backend is currently only running locally (the AWS deployment is still pending), you'll need to switch all API base URLs in the frontend to **http://localhost:3000** for now.

Once you've updated the API endpoints, run the usual setup commands:

```
npm install

npm run dev
```

The frontend will be up on **http://localhost:5173** (Vite default).

Before you can use the app properly, you'll need to generate a JWT token. Open Postman and follow this flow:

- First, **register** a user by sending a POST request to:
  http://localhost:3000/api/auth/register

- Then, **login** with the same credentials via:
  http://localhost:3000/api/auth/login

The login will return a JWT token. Copy that token and use it to access protected parts of the app. You'll need to inject the token into your authorization headers when testing API calls.

Until the backend team finishes resolving their AWS setup, everything runs and is tested locally through this flow. Once AWS is sorted, you'll just need to point the API URLs back to the production endpoints.

4. Folder Structure

```
/Frontend_Kanban
  /images                  # Static assets
  /node_modules            # Dependencies
  /public                  # Public static files
  /src
    /adminComponents       # Components for Admin role
    /pmComponents          # Components for PM role
    /volunteerComponents   # Components for Volunteer role
  /assets              # Icons, images, or fonts used inside src
  App.jsx              # Root component with router & providers
  forgotPassword.jsx   # Forgot Password page
  resetPassword.jsx    # Reset Password page
  login.jsx            # Login page
  main.jsx             # Likely entry point after login
  pageNotFound.jsx     # 404 page
  successMessage.jsx   # Generic success screen
.env                   # Environment variables
eslint.config.js       # ESLint rules
tailwind.config.js     # Tailwind customizations
vite.config.js         # Vite configuration
.gitlab-ci.yml         # CI/CD pipeline
branching-strategy.md  # Git branching guide
README.md              # Project instructions
```

5. Environment Variables

The following store it in a .env file

```
VITE_GAPI_CLIENT_ID="1067817291108-c4hlvf1l096nl9fiice06erb65uf8mfs.apps.googleusercontent.com"

VITE_SERVICE_ID="service_mlbjdkr"

VITE_TEMPLATE_ID="template_n4ilima"

VITE_PRIVATE_KEY="mO9nL1o9Nwlv9aY6S"
```

```
VITE_JWT_TOKEN
="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NmZlZmUyOTA5MTkwMDI1YjIwYTM3Z
jkiLCJmaXJzdF9uYW1lIjoiSm9obiIsImxhc3RfbmFtZSI6IkRvZSIsImVtYWlsIjoiam9obmRvZUBleGFtc
GxlLmNvbSIsInJvbGUiOiJQcm9qZWN0IE1hbmFnZXIiLCJpYXQiOjE3MjgzMTYzNDgsImV4cCI6MT
cyODQwMjc0OH0.uIPUyLIwqX-EkdUsCGi2Iuj-mgLMCdSv0wzBwN7uR6Y"

# VITE_API_URL_PROJECTS = "http://18.191.65.40:3000/api/projects/"

VITE_API_URL_PROJECTS = "http://localhost:3000/api/projects/"
```

---

6. Build & Deployment

- **Build Command**: npm run build

7. Work Summary (Frontend Contributions)

My frontline work centered on enhancing both the Admin and end-user experiences across the Kanban Board system:

Board Feature

- Established full Front-End and Back-End integration so that tasks created, edited or deleted persist across sessions.
- Built out Edit Task and Delete Task functionality merging them into the dashboard structure.
- Added a live count indicator to each Kanban column showing number of cards.
- Refined the drag-and-drop logic to reduce reordering glitches.

Profile Feature

- Restyled the Admin Profile page to match original design.
- Hooked up password-change flow and user-load logic to back end.
- Implemented Save Changes, Delete Account and Return buttons.
- Applied same board and profile updates for PM role and Volunteer role

8. Pending Work / Known Issues

- The account page still needs to handle the changes made to user features (make sure it is connected and function with the backend)
- The People feature should be the next step to implement (the real profiles created so far should be displayed)

- Add functionality and keep changes on the main page.

8. Gotchas & Tips

- Tailwind config overrides the default spacing scale.
- Make sure to clear API mocks when switching environments.
- Board state is cached locally; clear localStorage when testing board resets.