

Introduction

The purpose of this project is to develop a proof of concept for a customer feedback form with sentiment analysis. The solution allows users to submit feedback via a frontend application, which is then analyzed for sentiment and stored in a database. The sentiment analysis results can be viewed by ADMIN users.

Overview

The project implements a frontend application built with Vite + React, and a backend REST API developed using Node.js with Fastify.

MongoDB serves as the database for storing feedback and users data.

The project is deployed on an AWS EC2 instance, where deployment management is handled through PM2 for process management and Nginx for SSL encryption and proxy configuration. Additionally, the project is under source control with Git.

Architecture

The architecture adopts a client-server model, with the frontend and backend communicating through Nginx acting as a proxy.

- The backend REST API, powered by Fastify, integrates with MongoDB using Mongoose for data interaction. Bcrypt handles password hashing, and JWT tokens are utilized for authentication.
- The frontend, constructed with Vite, React, and Tailwind CSS, interacts seamlessly with the backend API.

Backend Features

The backend API has essential features:

- Authentication and authorization: JWT tokens are generated and returned as response to "/login" endpoint. Middleware validates token presence in the "Authorization" header for access to admin routes.
- Admin routes: A GET endpoint ("/admin/feedbacks") facilitates access to feedback data for admin users.
- Feedback submission: A POST endpoint ("/submit-feedback") enables users to submit feedback effortlessly.

Frontend Implementation

The frontend component offers user-friendly interfaces:

- End user interface:
 - Accessible at <https://acme-feedbacks.publicvm.com/feedbacks>, allowing users to provide feedback conveniently.
- Admin interface:
 - Admins must authenticate at <https://acme-feedbacks.publicvm.com/admin/login>,
 - At <https://acme-feedbacks.publicvm.com/admin/feedbacks> admin can access to feedback data and sentiment analysis results.

The interfaces are built with Tailwind CSS for consistency, utility-first approach and mobile-friendly design.

Security

Authentication and authorization mechanisms with bcrypt for password hashing will ensure secure access to data and protect against unauthorized access. HTTPS SSL encryption on the server will protect the data in transit.

Database

The database, named « ACME » has two main collections: feedbacks and users. This structure allows for efficient storage and retrieval of feedback data and user information.

FEEDBACKS

Each document in this collection represents a single feedback entry submitted by a user through the feedback form. It implements following fields :

- **_id**: Unique identifier for the feedback entry.
- **name**: The name of the user submitting the feedback.
- **text**: The textual content of the feedback provided by the user, with a maximum length of 1000 characters.
- **sentiment**: The sentiment analysis result calculated for the feedback text.
- **createdAt**: Timestamp indicating when the feedback was submitted.

USERS

The users collection stores information about registered users of the application. Each document in the users collection represents a single user profile. It implements following fields:

- **_id**: Unique identifier for the user profile.
- **email**: The email address associated with the user's account.
- **role**: The role assigned to the user, such as "admin" or "regular user," defaulting to "user."
- **name**: The name of the user.
- **password**: The hashed representation of the user's password, ensuring secure storage.
- **createdAt**: Timestamp indicating when the user account was created.
- **updatedAt**: Timestamp indicating the last time the user profile was modified.

Deployment

The solution is deployed on the cloud on AWS Ec2 instance. Ec2 instances can be scaled up or down on demand, are reliable thanks to AWS high availability service, secured with (IAM) access management and network isolation. This deployment enables seamless integration with other AWS services (S3, VPC..) that would enhance the functionality and performance of the application.

Source control

The project is managed and version-controlled using Git, with the repository hosted on GitHub at the following URL : <https://github.com/rakhim06/Feedbacks>.