# Review and Rating Aggregator

# Abstract:

The **Review and Rating Aggregator** is designed to collect and process user ratings and comments through a user-friendly interface. The system allows users to submit ratings along with optional comments, and the data is processed on the server-side using a Flask backend. The project aims to demonstrate the integration of a frontend web interface with a backend server to handle user input and provide feedback.

# Introduction

The **Review and Rating Aggregator** provides a platform for users to submit ratings and comments for a given service or product. The application is built using a combination of HTML, CSS, and JavaScript for the frontend, and Flask for the backend. The communication between the frontend and backend is facilitated through AJAX requests, showcasing the use of modern web development technologies.

# Requirements

## Hardware Requirements

* A computer or server to host the Flask backend.
* Web browser for users to interact with the frontend.

## Software Requirements

* Python 3.x
* Flask
* HTML, CSS, JavaScript for frontend
* flask-cors extension for handling CORS
* Web browser

# System Architecture

The system follows client-server architecture:

## Client-Side (Frontend):

* HTML/CSS for the user interface.
* JavaScript for client-side scripting.
* AJAX requests to communicate with the Flask backend.

## Server-Side (Backend):

* Flask framework for handling HTTP requests.
* flask-cors extension for handling Cross-Origin Resource Sharing (CORS).
* JSON for data interchange between frontend and backend.

# Design

## Frontend Design

* The frontend is designed to be intuitive for users to submit ratings and comments. It includes an interactive star rating system and a form for entering comments.

## Backend Design

* The backend is implemented using Flask, which handles incoming POST requests from the frontend. The server-side logic processes the received data, and a JSON response is sent back to the client.

# Implementation

The implementation involves creating HTML/CSS templates for the frontend, integrating JavaScript for interactivity, setting up a Flask application for the backend, and incorporating flask-cors to handle CORS issues.

# Testing

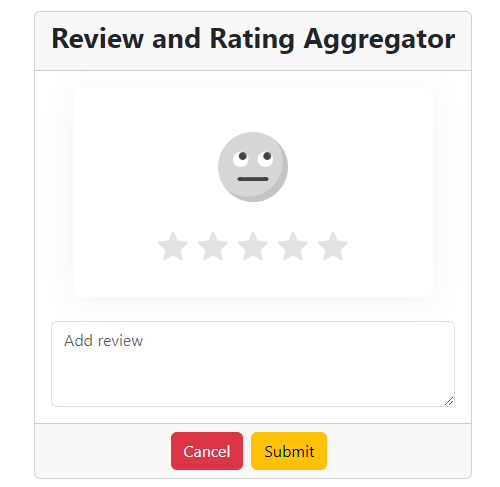
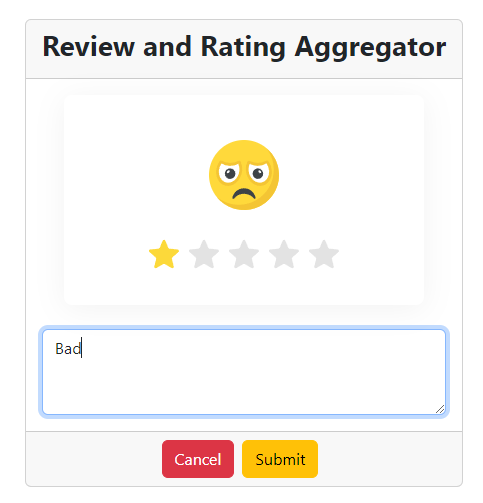
The application has undergone testing to ensure the following:

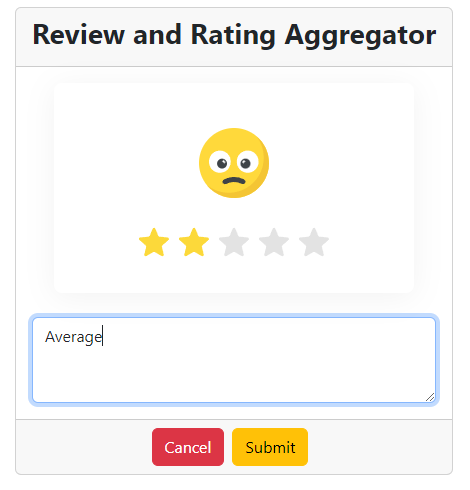
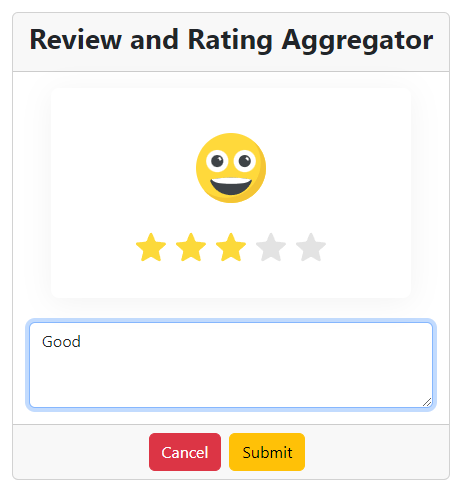
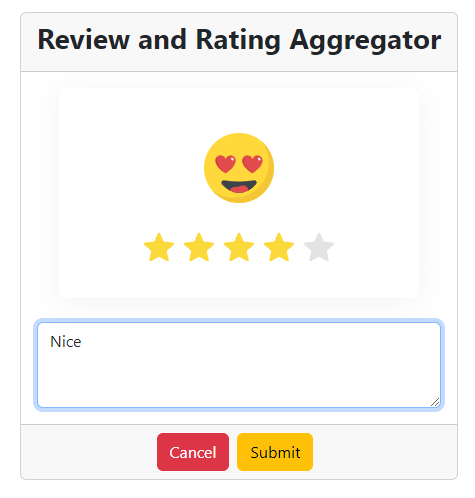
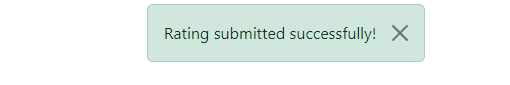
* Proper functioning of the rating system.
* Successful handling of user comments.
* Correct communication between the frontend and backend.
* Error handling for invalid inputs or server-side issues.

# Results and Discussion

The Review and Rating Aggregator has been successfully implemented and tested. Users can submit ratings and comments, and the data is processed on the server. The integration of Flask with the frontend allows for a seamless user experience.

# Screenshots

# Conclusion

In conclusion, the Review and Rating Aggregator showcases the integration of frontend and backend technologies to create a functional and user-friendly rating system. The project demonstrates the utilization of modern web development practices and can be extended for various applications that involve user feedback and ratings.