**Design Document for the Feedback Web Application**

**Team Cloude**

Morufat Aminu

Ernest Obehi

Saheed Amusa

Stanley Omorogbe

Tunde Bankole

**Project Overview**

**Project Name:** Feedback Form Web Application

**Description:** A simple web application for collecting user feedback.

**Goals:**

* Allow users to submit feedback.
* Store feedback in a database.
* Display a confirmation message upon successful submission.
* Use OKTA for authentication.
* Monitor application, server, and network using ELK
* Load balance using HAProxy
* Secure the application with OKTA authentication.

**Functional Requirements**

1. Users must authenticate using OKTA before accessing the feedback form
2. Users can submit feedback using a form.
3. Feedback includes first name, last name, rating, and comment.
4. Store feedback in a database.
5. Display a confirmation message upon successful submission: "Thank you, your feedback has been submitted successfully."
6. Display the total number of feedback received on the backend.

**Non-Functional Requirements**

1. The form should be user-friendly and responsive.
2. Data should be securely stored in the database.
3. The application should handle concurrent submissions efficiently.
4. The application should ensure secure authentication and authorization.

**Architecture Design**

**Components:**

* **Frontend:** HTML/CSS for the feedback form.
* **Backend:** PHP for handling form submission and database interaction.
* **Database:** MySQL to store feedback.
* **Authentication:** OKTA for user authentication.
* **Load Balancer:** HAProxy for distributing incoming traffic across multiple web servers.
* **Monitoring:** ELK stack (Elasticsearch, Logstash, Kibana) for monitoring application, server, and network performance.

**Interactions:**

* User accesses the feedback form and is prompted to authenticate via OKTA.
* Upon successful authentication, the feedback form is displayed.
* User submits the feedback form.
* PHP script processes the submission and inserts data into the database.
* PHP script displays a confirmation message to the user: "Thank you, your feedback has been submitted successfully."
* Feedback data is logged and monitored using the ELK stack.

**Database Design**

**Tables:**

- feedback

* id (INT, Primary Key)
* first\_name (VARCHAR)
* last\_name (VARCHAR)
* rating (INT)
* comment (TEXT)
* entered\_date (TIMESTAMP)

-SQL

CREATE TABLE feedback (

id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

rating INT NOT NULL,

comment TEXT NOT NULL,

entered\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

- Schema  
‘id’: An integer that auto-increments with each new record and serves as the primary key.

‘first\_name’: A varchar field with a maximum length of 50 characters, cannot be null.

‘last\_name’: A varchar field with a maximum length of 50 characters, cannot be null.

‘rating’: An integer field, cannot be null.

comment: A text field for user comments, cannot be null.

‘entered\_date’: A timestamp field that defaults to the current timestamp at the time of record creation.

**Entity-Relationship Diagram**

+------------------+

| feedback |

+------------------+

| id |

| first\_name |

| last\_name |

| rating |

| comment |

| entered\_date |

+------------------+

**User Interface Design**

**Wireframes:**

- Feedback Form:

* Input fields for first name, last name.
* Radio buttons for rating.
* Textarea for comments.
* Submit button.

- Confirmation Message

* Text: “Thank you, your feedback has been submitted successfully.”

**User Interface Form:**

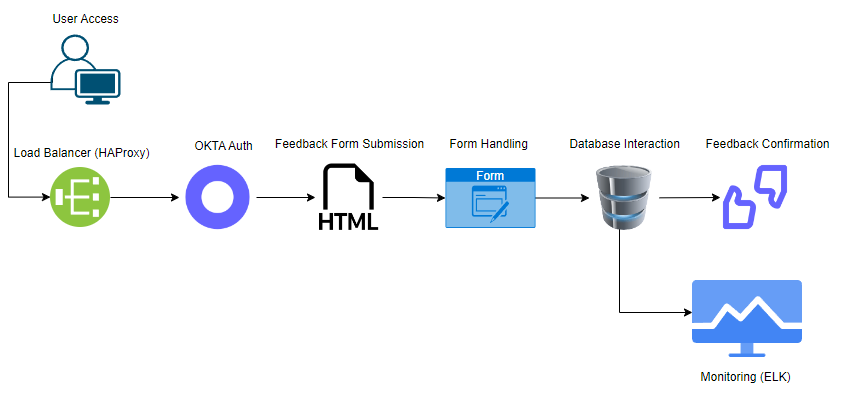
**A blue and white form with white text

Description automatically generated**

**Workflow**

1. **User submits feedback:** User accesses the form and is prompted to authenticate via OKTA.
2. **OKTA authentication:** User logs in using OKTA credentials.
3. **Feedback form display:** The feedback form is displayed upon successful authentication.
4. **Form submission:**
5. **Form handling:** The PHP script ‘feedback.php’ processes the form data.
6. **Database insertion:** The script inserts the feedback into the feedback table.
7. **Confirmation message display:** The script displays "Thank you, your feedback has been submitted successfully."

**Diagram**

****

**Security Considerations**

1. **Input Validation:** Ensure all input fields are validated to prevent SQL injection and XSS attacks.
2. **Database Security:** Secure the database with a strong password and disable remote root access.
3. **HTTPS:** Use HTTPS to encrypt data transmitted between the client and server.
4. **Authentication:** Use OKTA for secure user authentication.

**Implementation Plan**

1. **Week 1:**
2. **Week 2:**
3. **Week 3:**