Simple Voting System Test

System Description

The Simple Voting System allows users to vote on questions registered by the administrator. The administrator can register questions with titles, identifiers, and response options. Response options may include an image, a title, and a brief description.

Users can vote on registered questions and later the administrator can view the number of votes received for each question, along with the percentage of votes.

The administrator can configure whether to show or hide the total votes for each question after the user votes.

The administrator can also configure to disable the voting system altogether.

The system should also be integrated with Drupal, allowing votes to be displayed. Additionally, the system should be able to provide an API for authorized third-party applications to interact with the votes, allowing, for example, registered votes to be made available in an application with the entire Drupal flow experience.

Functional Requirements

1. The administrator must be able to register questions with unique titles and identifiers.
2. Each question can have multiple response options.
3. Response options may include an image, a title, and a brief description.
4. Users must be able to vote on a question by selecting one of the response options.
5. The system must record the votes received for each question and response option.
6. The administrator must be able to view the total number of votes received for each question, along with the percentage of votes.
7. The administrator can configure to disable the voting system altogether.
8. The system must be integrated with Drupal to display the votes.
9. The administrator must be able to configure whether the total votes for each question should be shown or hidden in Drupal after the user votes.
10. The system must provide an API for authorized third-party applications to interact with the registered votes.

Non-Functional Requirements

1. The system must be easy to use and intuitive for the administrator to register questions and response options.
2. The system must be secure, protecting vote data and preventing unauthorized manipulation.

Technical Requirements

1. Do not use community modules to solve the problem except for "restui".
2. Do not use node for entities.

3. The code must be delivered in a github repository, with a database dump and environment via lando.

Note that this test is focused exclusively on the development and operation of the system backend. Therefore, the layout, design, or appearance of the user interface (frontend) will not be considered in the evaluation. We are interested in aspects such as:

- Correct implementation of business logic
- Structure and organization of the code
- Implemented functionalities according to requirements
- Appropriate use of technologies and good backend development practices
- Code performance and efficiency.