VoteMash: A Voting Application

The application is built using following technologies.

1. **Google App Engine (Python, webapp2 framework, app engine's data store)**
   Google App Engine with python is used for building the entire application. The application uses webapp2 framework and utilizes the app engine's data store to store all the data corresponding to categories and items.
2. **Git**
   Git is used a a source control management tool. The code is regularly committed in the git repository along with the commit messages.
3. **Jinja templates**
   Jinja templates are used to render all the html files generated by the system. The templates makes it easy to keep the core logic of the system separate from its view which allows changing them independent of each other. All the information to be rendered, along with error messages (if any), are done through jinja templates.

The code is organized as follows.

1. **votemash.py**
   Handles all the initial requests to the major tasks namely edit, vote, search and result. The subsequent requests to these tasks are handled by their corresponding files as mentioned below.

2. **Edit.py**
   This file handles all the requests corresponding to editing task. The editing includes create/delete/rename category/item and import/export categories (along with their items). It also controls the flow between itself and rendered html files in the process. All communication happens through jinja templates.

3. **Vote.py**
   This file handles the request for voting the items. It also controls the flow between itself and rendered html files in the process. All communication happens through jinja templates.

4. **Result.py**
   This file is responsible for handling requests for seeing the leaderboard. It also controls the flow between itself and rendered html files in the process. All communication happens through jinja templates.

5. **Search.py**
   All the processing for search happens through this file. It takes the keywords to search and renders the search result through jinja templates.

6. **Data.py**
   This file implements the basic functionalities those are used by above files. Some of the functionalities include 'checking for empty variable', 'generating keys', 'processing dom trees during importing/exporting data', etc.

7. **html files**

   Each html file handles one of the data rendering task for each functionality. All the data for rendering is obtained through templates. Each html file is dedicated to render the output of each function.