

DESIGN DOC

Database: Sqlite3

Backend: Python (Flask)

Frontend: HTML,JQuery(Ajax)

Database Design

Table : location

Column: id, name

Table: department

Column: id,loc_id,name

Foreign Key: loc_id ->id [location]

Table: category

Column: id,dept_id,name

Foreign Key: dept_id ->id[department]

Table: subcategory

Column: id,cat_id,name

Foreign Key: cat_id -> id[category]

Table: sku

Column: id,sku_id,subcategory_id

Foreign Key: subcategory_id -> id[subcategory]

Table: user

Column: email,pass_hash

BackEnd And FrontEnd Design

Authorization

Successful Authorization redirects user to another page where he can CRUD on Metadata and Search for SKUs

HTTPBasicAuth used for basic authentication (OAuth Not supported in current version).

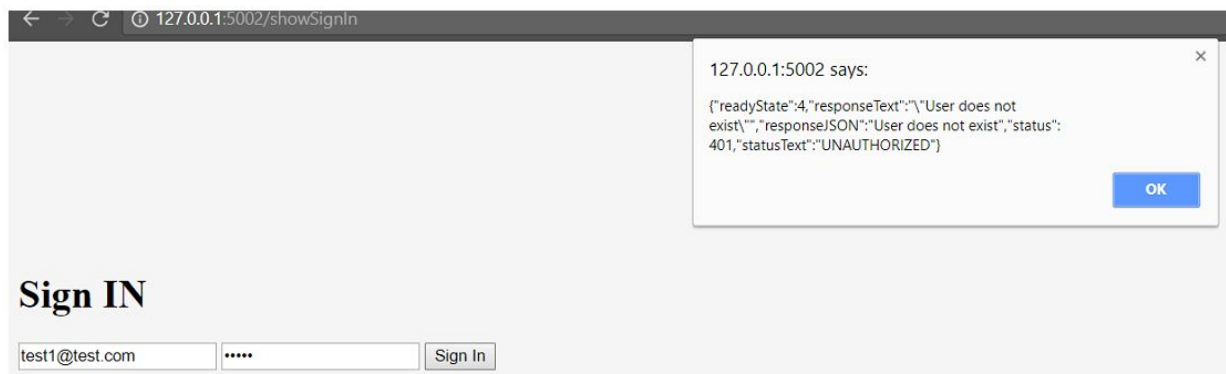
Error cases handled

- 1) Sign In: User Does Not Exist
- 2) Sign In: Incorrect password
- 3) Sign Up: User already Exist

Alert used to display error (Better error display suggested. Not Done yet)

No API access is allowed in case of failed/without Authentication

Hashed Password is used for authentication using Passlib.



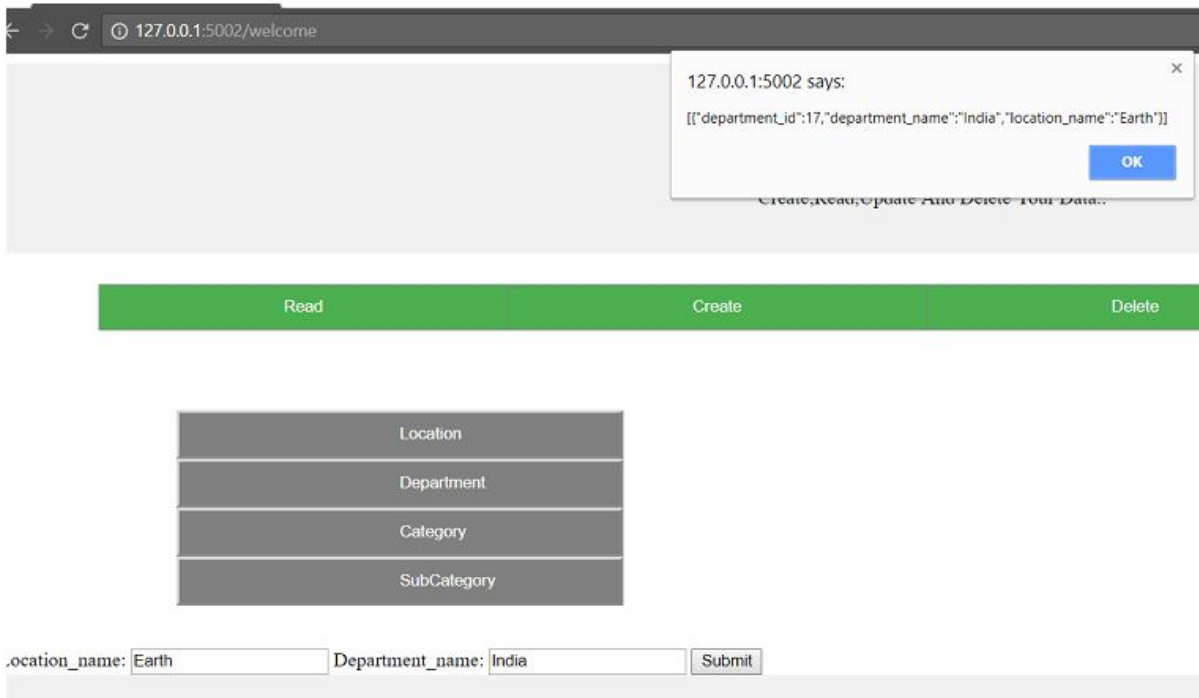
Sign Up

CRUD Methods support for Metadata.

CREATE:

- # User provided Metadata value (and Parent if applicable) is passed to API via jquery.
- # User is notified of Success event (Failure in case of error)
- # Error cases handled:
 - 1) Unauthorized
 - 2) Duplicate Entry
 - 3) Failed creation in case of Parent does not exist (e.g some category been tried to create but associated department does not exist.)



READ:

Metadata values are provided to user from API in JSON format which then parsed Using JQuery in tabular form.

User is notified of Success event (Failure in case of error)

Error cases handled:

- 1) Unauthorized

Read	Create
------	--------

Location
Department
Category
SubCategory

1	Perimeter
2	Center
25	USA
26	India
27	New York
28	Japan
29	Nainital

UPDATE:

- # User provided Metadata old and new value is passed to API via jquery.
- # User is notified of Success event (Failure in case of error)
- # Error cases handled:
 - 1) Unauthorized
 - 2) Entry does not exist

127.0.0.1:5002/welcome

127.0.0.1:5002 says:
"Entry Updated"
OK

Create, Read, Update And Delete Your Data..

Read Create Delete

Location
Department
Category
SubCategory

Old value: New value:

DELETE:

- # User provided Metadata value (and Parent if applicable) is passed to API via jquery.
- # User is notified of Success event (Failure in case of error)
- # Error cases handled:
 - 1) Unauthorized
 - 2) Entry does not exist
 - 3) Failed deletion in case of Child exist (e.g some category linked to department, can't delete department)

127.0.0.1:5002/welcome

127.0.0.1:5002 says:
"Entry Deleted"
OK

Create, Read, Update And Delete Your Data..

Read Create Delete

Location
Department
Category
SubCategory

Category_name: SubCategory_name:

GET SKU:

- # User provided Metadata value is passed to API via jquery.
- # User is notified of Success event (Failure in case of error)
- # List of all SKUs are return back in Json format which is parsed to tabular format via JQuery and displayed to user.
- # Error cases handled:
 - 1) Unauthorized

SKU

Get SKU Of Your Data..

SKU

Location_id: Perimeter	Department_id: Bakery	Category_id: Bakery Bread	Subcategory_id: Bagels	Submit
SKUDESC1				
SKUDESC14				

Installation Guide

#Download Python for windows- <https://www.python.org/downloads/windows/>
[Windows x86-64 executable installer](#)

Go to Cmd -> python folder

Pip install virtualenv (Install virtualenv)

Virtualenv venv

C:\python\venv\Scripts>activate (To activate virtual environment)

(venv) C:\python > pip install flask flask-jsonpify flask-sqlalchemy flask-restful

(venv) C:\python>pip freeze

aniso8601==1.3.0

click==6.7

Flask==0.12.2

Flask-Jsonpify==1.5.0

Flask-RESTful==0.3.6

Flask-SQLAlchemy==2.3.2

itsdangerous==0.24

Jinja2==2.10

MarkupSafe==1.0

python-dateutil==2.6.1

pytz==2017.3

six==1.11.0

SQLAlchemy==1.1.15

Werkzeug==0.12.2

Install Sqlite tool.

[Sqlite-tools-win32-x86-3210000.zip](#)