Recipeezy

a project by:

Adhir Uchil

Alex Plazas

David Shackleford

Luis Mendoza

Project Description

Purpose

Recipeezy is a website which will **collect and organize recipes** from around the world. This will give people the opportunity to **share** their own personal or family recipes, while also **exploring** different foods and finding **new ideas** for their culinary adventure.

Major Features

- Recipe Lookup
  - The user will be able to look up recipes based on a wide range of input criteria
- Recipe Submission
  - The user will be able to submit their own recipe utilizing a simple user interface
- Voting System
  - The user will be allowed to vote on other people's recipes, which affect an accumulated "score" for the recipe and will affect the order that the recipe appears when doing a recipe lookup
- Dynamic Homepage
  - The user will be able to see the most upvoted recipes on their homepage to enhance the discovery process

Design Challenges

- Integrating cloud based services
  - Image storage - AWS
  - Database management - Heroku
- Connecting front-end and back-end together with use-cases
  - Voting system
  - Robust searching
  - Dealing with containers
- Team organization
  - Different time zones
  - Different levels of knowledge with design tools

Tools

Tools

Project Management

- Notion
- Github

Frontend

- HTML/CSS
- Bootstrap
- Javascript
  - Vue.js

Backend

- Python/Flask
  - SQLAlchemy
    - Postgres
- AWS S3
  - Boto3

Tools - Project Management

Notion

- Pros:
    - Highly customizable
    - Accessible from anywhere
    - Support for automations
- Cons:
    - Learning curve
    - Not 1-to-1 with other applications (Trello, Jira, etc.)
    - Limit to free trial (5 members)

Github

- Pros:
    - Standard in version management
    - Remotely collaborate with team
    - Display and collaborate with public
    - Understandable CLI
    - Easy to use UI
- Cons:
    - Some features behind paywall
    - Security (publishing database and aws info)

Tools - Frontend

HTML/CSS/JS

- Pros:
  - Web standard
  - Ubiquitous
  - Well supported
- Cons:
  - Compatibility between browsers

Bootstrap

- Pros:
  - Keeps classes organized
  - Meets best practices
- Cons:
  - Bulky
  - Removes “uniqueness”

Vue

- Pros:
  - Abstract the low-level JS
  - Simplifies the development
- Cons:
  - Takes time to learn
  - Relying on outside libraries that are prone to change

Tools - Backend

Flask/Python

- Pros:
  - Modular
  - Common language
  - Documentation
- Cons:
  - Scalability (request handling)
  - 3rd Party Reliance

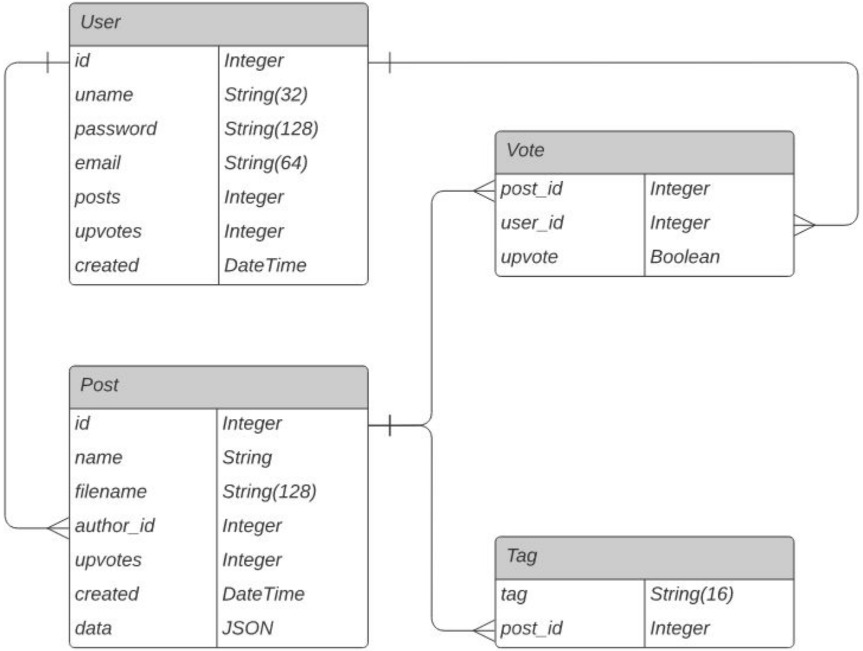
SQLAlchemy/Postgres

- Pros:
  - Heroku
  - Object based
- Cons:
  - Higher memory
  - Slower

AWS S3

- Pros:
  - Fast with static files
  - Free tier
  - Easy integration (boto)
  - Documentation
- Cons:
  - Amazon wants you to pay extra

Database Design



User	
id	Integer
uname	String(32)
password	String(128)
email	String(64)
posts	Integer
upvotes	Integer
created	DateTime

Vote	
post_id	Integer
user_id	Integer
upvote	Boolean

Post	
id	Integer
name	String
filename	String(128)
author_id	Integer
upvotes	Integer
created	DateTime
data	JSON

Tag	
tag	String(16)
post_id	Integer

Testing

- Unittest framework (python), in conjunction with flask class TestCase
- Tag db, Vote db, User db, and the Post db
- Testing insertion into database
    - Username, recipe creation
    - Votes for recipe post
    - Tag matching
- Testing image saving in static folder
- Testing routing confirmation

Documentation

- Documentation created using combination of auto-documenter and manual entry
- Sphinx library used to generate docstrings for class methods
- Generated docstrings manually edited to include detailed information
- Important code sections given in-line comments to describe functionality
- Ensured compliance with PEP 257 (Docstring Conventions) in final documentation across all parts of the codebase

Project Demonstration

Thank you!

From the Recipeezy Team

Supplementary

Flask Modules

- Flask-Login : Helps with secure login and saving username data
- Flask-WTForms : Helps with form entry data for front end
- Flask-SQLAlchemy : SQLAlchemy implementation for Flask