Beehive

Potential Mentors:
Pradeeban Kathiravelu
David Moxley

Length: 350h

Contact Info

Name: Mingjun Li

Email: limj76@mail2.sysu.edu.cn
yanhuojunjun@gmail.com

GitHub Page:
https://github.com/yanhuojunjun

Location & Time Zone:
Guangzhou, China, UTC+8:00

Introduction

Many individuals are enduring the trials of life, including vagrants, people recovering from addiction, people with AIDS, prisoners, and so on. Unfortunately, these individuals are often overlooked and marginalized. Furthermore, outsider art, which reflects the diverse life experiences of these marginalized groups, faces significant discrimination due to the typically limited artistic training of outsider artists. Given these factors, research on marginalized folks and the impact of negative stereotypes on their daily life is highly necessary to undertake.

Beehive is an open-source prototype that can offer assistance in conducting this type of research. Its objective is to develop a digital method for translating outsider art into a digital database, storing images in easily retrievable formats. In this way, Beehive provides the researchers in the University of Alaska System with an efficient platform to understand the lives of marginalized individuals and conduct their research effectively.

Beyond serving researchers, Beehive also offers marginalized individuals and outsider artists a platform to share their life stories. Consequently, it can raise awareness about the struggles of marginalized individuals, and advance the delivery of human services to them, not only in Alaska but also globally.

Deliverables

Beehive will be represented in the form of a website. By the end of GSoC 2024, I would like to develop the Beehive website with the following parts:

1. Landing Page

This will be the starting point of Beehive. It will have the following features:

- Support for user authentication using Google SSO, which ensures security measures are in place to prevent any unauthorized access to the website or tampering with the database.
- Users can sign up or log in using their Google account, and then users' information will be used to enable additional functionalities.
- Based on users' identity levels, users could be directed to either the admin interface or the user interface after logging in.

2. Admin Interface

Researchers from the University of Alaska System will be directed to the admin interface after logging in, which will provide the following functionalities:

- Perform addition, deletion, and modification operations on data in any table of the database.
- Modify relevant configurations by updating specific data in the database.
- Search for data (primarily images) based on different attributes. For example, through the admin interface, a researcher can access specific types of images (such as images related to vagrants) and retrieve relevant statistical data.

The functionalities above would effectively support the conduct of relevant research.

3. User Interface

Marginalized individuals and outsider artists will be directed to the user interface after logging in, which will provide the following functionalities:

- Share photos related to marginalized individuals on the platform. Titles, types, locations and descriptions can be uploaded along with the photos.
- Search and browse all the photos on Beehive, having insight into the lives of various marginalized individuals.
- Make comments on photos and share their own opinions or stories with each other. The comments between marginalized individuals may also be used in the research.

- Provide a MySpace system that allows users to modify their information or their photos' information, as well as review their favorite photos.

Implementation

I would like to develop Beehive using the Django framework. In the framework:

- Backend work will be implemented with Python.
- Frontend work will be implemented with HTTP.
- A MySQL database will be connected to store information.

The overall architecture of Beehive can be depicted in the figure below:

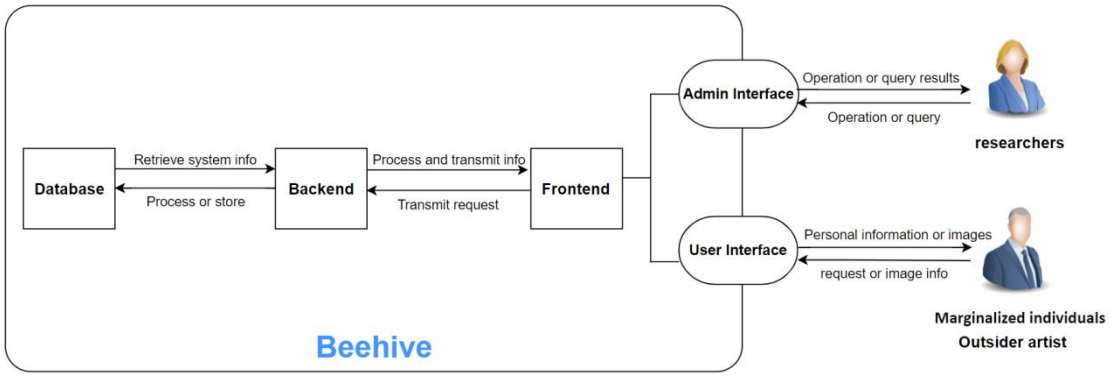


Fig: Architecture diagram of the proposed framework

More specifically, to fulfill the deliverables and goals above, I would like to complete the following work by the end of GSoC 2024:

1. Database design

- According to the requirements, design the database schema to store information related to users, images, comments, and other relevant data.
- Determine the relationships between different tables and define appropriate indexes and constraints.
- Connect the designed database to the Django framework

2. User authentication

- Implement user authentication using Google SSO
- Develop login and signup functionalities
- Integrate Google authentication in the Django project

3. Backend of Beehive

As described above, the backend work of Beehive could be divided into three parts: landing page, admin interface and user interface.

3.1 Landing page

Some of the most prominent API are given below:

- GET / login :
  return the rendered login page
- POST / login / {login info}:
  submit the login info and redirected to different interfaces
- GET / signup:
  return the rendered signup page
- POST/ signup / {signup info}:
  submit the signup info and get an account

3.2 Admin interface

Some of the most prominent API are given below: (take the image table as an example)

- GET / admin_image / {image_list}:
  return the image list page
- POST / admin_image / {image_list, query}:
  submit queries on images and redirected to the result page
- GET / admin_image_add:
  return a page to add an image
- POST / admin_image_add / {image info}:
  submit image info and add a new image into the database
- GET / admin_image_update / {image_id}:
  return the page to update the selected image
- POST / admin_image_update / {update info}:
  submit updated info and update the image
- GET / admin_image_delete / {delete_id}:
  delete the selected image

3.3 User interface

Some of the most prominent API are given below:

- GET / user_home / {user, images}:
  return a home page showing all images
- POST / user_home / {user, images, query}:
  submit queries on images and redirected to the result page
- GET / myspace / {user, images}:
  return a user's space page

- GET / share / {user_id}:
  return a page to share images
- POST / share / {image info, user_id}:
  submit image info and add a new image into the database
- GET / user_image / {user_id, image_id}:
  return a detail page of the selected image
- POST / user_image / {comment, user_id, image_id}:
  submit a comment for the chosen image

4. Frontend of Beehive

- Develop the interface of Beehive using HTML, CSS, and JavaScript.
- Design responsive and user-friendly web pages for sharing photos, browsing photos, making comments, searching for photos and managing database data.
- The landing page includes the following sections: log in page, sign up page
- Admin interface includes the following pages: browsing, adding, updating, searching pages for every table in the database
- User interface includes the following pages: home page, myspace page, share page, image detail page

5. Deploy and test

- Deploy the Beehive website in a production environment using a web server
- Conduct thorough testing to ensure the functionality, performance, and security of the website.

6. Related documentation

While developing the open-source Beehive website, I would keep in mind the importance of regular maintenance and periodic expansion. So I would document various aspects of the project in the Beehive GitHub repository, including the following sections:

- Contributor's guide: Instructions and guidelines for contributors on how to clone, run, and extend Beehive. This guide will help new developers understand the project's architecture and contribute effectively.
- User manual: Comprehensive instructions for researchers and other users on how to use Beehive.

Pre-proposal Prototype

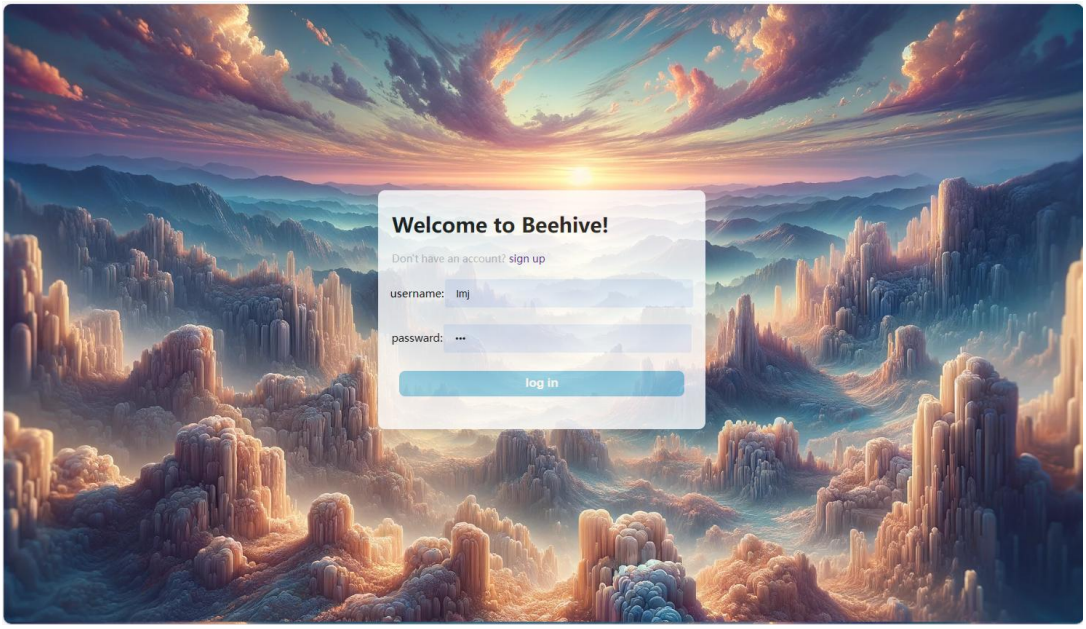
I have created a rough prototype before diving into the proposal just for mentors' reference on how I imagined Beehive looks like.

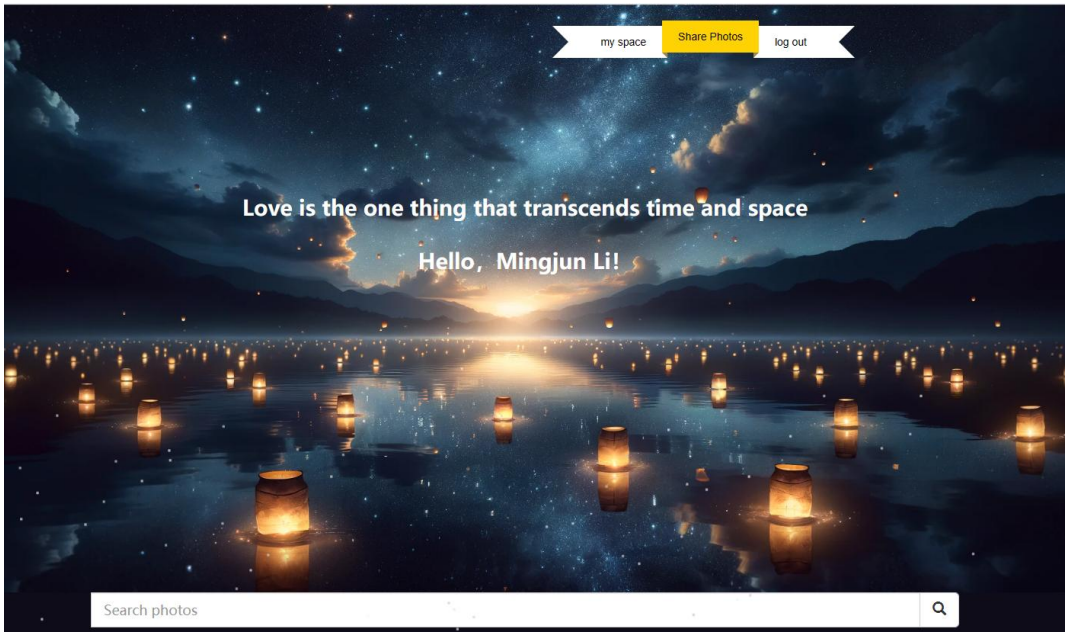
The prototype adheres to the overall architecture outlined earlier. It is developed using Django and is connected to a MySQL database on my local computer. I have developed both a simple backend and frontend for the prototype, ensuring it is fully responsive with a neat UI.

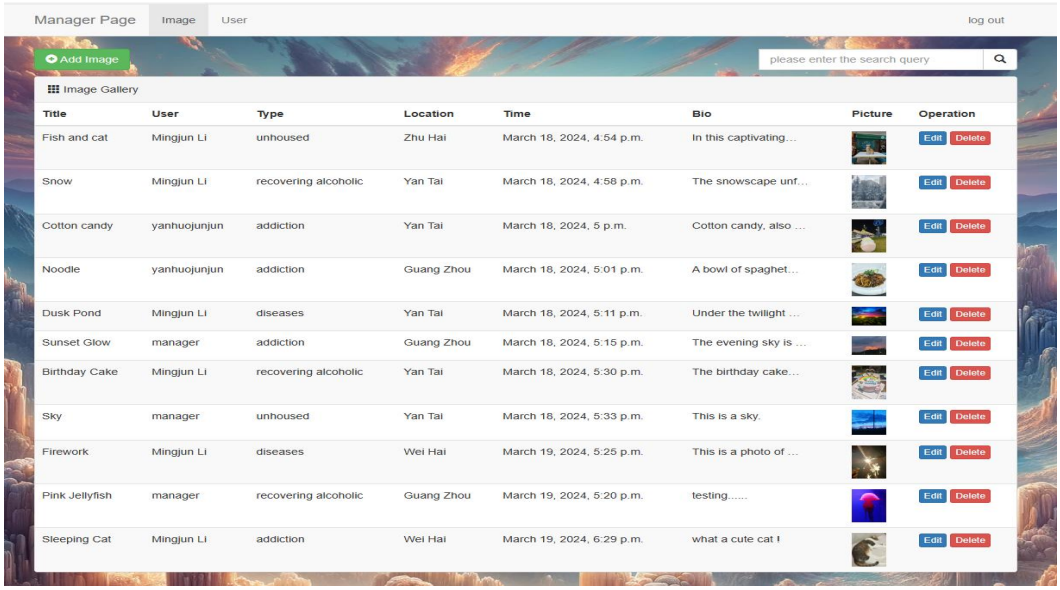
Currently, the prototype only includes two tables in the database(user table and image table). Additional tables could be created based on further database design, and user authentication will utilize Google SSO in the future. In the prototype, I have used sample photos from my phone to demonstrate its functionalities.

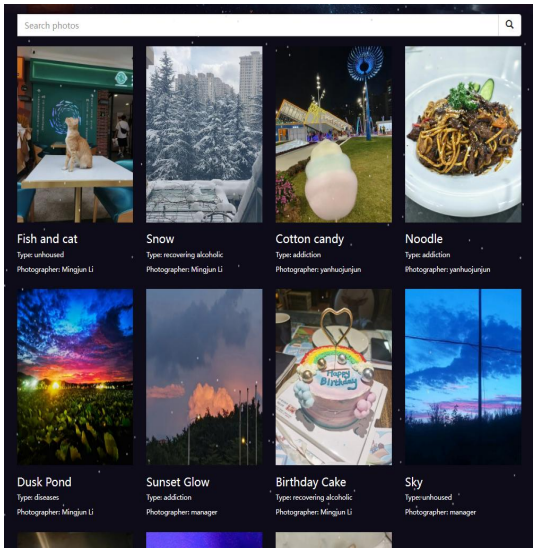
My prototype code as well as some demo videos can be found here: https://github.com/yanhuojunjun/Beehive

Here are some of website pages in my prototype:




my space
Share Photos
log out
Love is the one thing that transcends time and space
Hello，Mingjun Li!
Search photos


Manager Page
Image
User
log out
Add Image
please enter the search query
Image Gallery
Title User Type Location Time Bio Picture Operation
Fish and cat Mingjun Li unhoused Zhu Hai March 18, 2024, 4:54 p.m. In this captivating...
Snow Mingjun Li recovering alcoholic Yan Tai March 18, 2024, 4:58 p.m. The snowscape unf...
Cotton candy yanhuojunjun addiction Yan Tai March 18, 2024, 5 p.m. Cotton candy, also ...
Noodle yanhuojunjun addiction Guang Zhou March 18, 2024, 5:01 p.m. A bowl of spaghet...
Dusk Pond Mingjun Li diseases Yan Tai March 18, 2024, 5:11 p.m. Under the twilight ...
Sunset Glow manager addiction Guang Zhou March 18, 2024, 5:15 p.m. The evening sky is ...
Birthday Cake Mingjun Li recovering alcoholic Yan Tai March 18, 2024, 5:30 p.m. The birthday cake...
Sky manager unhoused Yan Tai March 18, 2024, 5:33 p.m. This is a sky.
Firework Mingjun Li diseases Wei Hai March 19, 2024, 5:25 p.m. This is a photo of ...
Pink Jellyfish manager recovering alcoholic Guang Zhou March 19, 2024, 5:20 p.m. testing.....
Sleeping Cat Mingjun Li addiction Wei Hai March 19, 2024, 6:29 p.m. what a cute cat !
Edit Delete


Search photos
Fish and cat
Type: unhoused
Photographer: Mingjun Li
Snow
Type: recovering alcoholic
Photographer: Mingjun Li
Cotton candy
Type: addiction
Photographer: yanhuojunjun
Noodle
Type: addiction
Photographer: yanhuojunjun
Dusk Pond
Type: diseases
Photographer: Mingjun Li
Sunset Glow
Type: addiction
Photographer: manager
Birthday Cake
Type: recovering alcoholic
Photographer: Mingjun Li
Sky
Type:unhoused
Photographer: manager

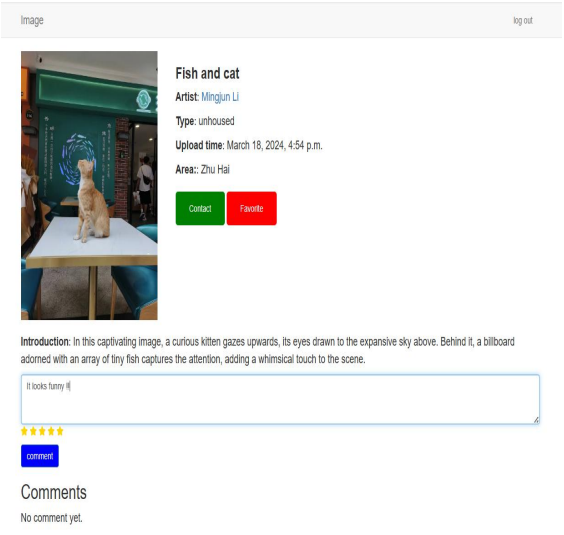


Image
log out
Fish and cat
Artist: Mingjun Li
Type: unhoused
Upload time: March 18, 2024, 4:54 p.m.
Area:: Zhu Hai
Contact
Favorite
Introduction: In this captivating image, a curious kitten gazes upwards, its eyes drawn to the expansive sky above. Behind it, a billboard adorned with an array of tiny fish captures the attention, adding a whimsical touch to the scene.
It looks funny !!
comment
Comments
No comment yet.

Schedule

No.	Duration	Objectives
Week 0	May 1 - May 11	➢ Community Bonding Period ➢ Get familiar with the team and mentors, and discuss more requirements and details about Beehive ➢ Find inspiration for website design
Week 1	May 12 - May 18	➢ Create the database design of Beehive ➢ Set up and deploy Django framework.
Week 2	May 19 - May 25	➢ Integrate Google authentication in Django project ➢ Implement user authentication using Google SSO
Week 3	May 26 - Jun 1	➢ Complete backend work for landing interface
Week 4	Jun 2 - Jun 8	➢ Design pages for logging in and signing up ➢ Complete frontend work for landing interface
Week 5	Jun 9 - Jun 15	➢ Design a search system for the admin interface ➢ Seek more efficient search methods for researchers
Week 6	Jun 16 - Jun 22	➢ Complete backend work for admin interface ➢ Implement efficient search on the database
Exams of school		
Week 7	Jul 6 - Jul 12	➢ Design pages for admin interface ➢ Complete frontend work for admin interface
Medium Evaluation		
Week 8	Jul 13 - Jul 20	➢ Design user interface functionalities ➢ Complete part of backend work for user interface
Week 9	Jul 21 - Jul 27	➢ Complete all backend work for user interface ➢ Design pages for user interface
Week 10	Jul 28 - Aug 3	➢ Complete frontend work for user interface
Week 11	Aug 4 - Aug 10	➢ Deploy Beehive in production environment ➢ Conduct thorough testing ➢ Write documentation
Week 12	Aug 11 - Aug 17	
Week 13	Aug 18 - Aug 24	➢ Make final adjustments ➢ Submit final work product
Final Evaluation		

In summary, I divide the project into five phases, each marked with a different color in the table

- **Phase One:** Week 0-1, requirements acquisition and framework for the project
- **Phase Two:** Week 2-4, user authentication and landing page of Beehive
- **Phase Three:** Week 5-7, admin interface of Beehive
- **Phase Four:** Week 8-10, user interface of Beehive
- **Phase Five:** Week 11-13, testing and documentary

GSoC work hours (UTC+8:00)

As a large project, Beehive is expected to require a total of 350 hours. I am capable of dedicating 35 to 40 hours per week to its completion.

I am flexible with my working hours and can immerse myself in the work for extended periods. I could work either during the day or at night, and I am willing to attend weekly meetings in the morning during the coding period (2 PM ~ 5 PM in AKST) to facilitate real-time communication with my mentors.

Communication preferences - I am comfortable with either Zoom or Skype, and I am willing to adjust to whatever platform suits the mentors best.

Other commitments

As indicated in the table above, I would have an exam during GSoC. To compensate for my unavailability due to exams, I plan to start my work two weeks earlier during the community bonding period, and I am willing to work extra hours every day after the exams to ensure the project stays on track.

During the GSoC 2024, I would strictly follow the schedule above and advance the project efficiently. I promise that GSoC will be my only commitment this summer and I will dedicate my full attention to it.

More About Me

1. Educational Background

I am currently a third-year undergraduate majoring in Computer Science at Sun Yat-Sen University. I ranked in the top 0.5% on the national college entrance examination, and currently maintain a GPA of 4.0/4.0, thus, I have been selected for the Top-notch Student Program in our country (select excellent CS students among the top universities in our country). Besides schoolwork, I have participated in

various software development projects out of interest, such as website and game projects.

2. Motivation

During my time at school, I developed a keen interest in database and website development. I have studied technologies such as database application development, website design, and HTTP during my courses. As graduation approaches, I have decided to pursue a master's degree in America, and work as a software engineer in the future.

Beyond professional aspirations, I am motivated to contribute to Beehive for a more personal reason. Recently, our professor shared a story of an open-source application named GeoPyTool. The developer of the application, who was a geologist with no programming background, wrote the sentences at the beginning of the readme file: "Few years ago, I had tumor and some other lifelong disease. Then I lost the ability to engage in geological field work. I hate to be a useless person and hope to contribute to other people and the world...", so he devoted the rest of his life to completing GeoPyTool.

I feel ashamed when I read this, as for years, I coded solely for my own benefit. Sometimes I would see vagrants lying along the street, but all I did was just feel sympathy for them. I didn't take any real action, just like many others. That's why I do want to do something for marginalized individuals, and do something for the world by contributing to open-source communities. So GSoC could be a good starting point for me, and the Beehive project of Alaska is my top choice.

3. Skill Set

Relevant skill set required for developing Beehive:
MySQL, Django, Python, HTML, CSS, and JavaScript

With these skills, I believe I can excellently complete the Beehive project.
Additionally, here are some projects that could demonstrate my relevant skill set:

- **Movie Review Website:**

https://github.com/yanhuojunjun/Movie-Review-Website

• Developed a movie review website based on Django and openGauss (an open-source database), including frontend as well as backend work
• Completed sophisticated functions of the review website

• Received full score in the database course

- **Fireworks Simulator Game:**

https://github.com/yanhuojunjun/Firework-Simulator-Game

• Implemented a sophisticated particle system and developed a fireworks simulator game based on the particle system, including frontend and backend work

• Designed a neat UI for users and gained users' favor

• Led a team of two members, facilitating effective communication and project management through github.

- **Distributed key-value System:**

https://github.com/yanhuojunjun/Distributed-Key-Value-System

• Implemented a distributed system connected to multiple databases

• Developed a search system for it, enabling efficient key-value search across the databases.

4. Relevant Contribution

A prototype of Beehive I've made can be found here:
https://github.com/yanhuojunjun/Beehive