"HungryMe"

Description:

 HungryMe is an online food suggestion website that suggests to people what food they can think of when they have no idea whether food to eat or cook at the moment.

 There will be the food of the day, A random excellent dish suggested to the visitor. The food of the day will change every day so visitors can have the idea of a new food they could try the next day.
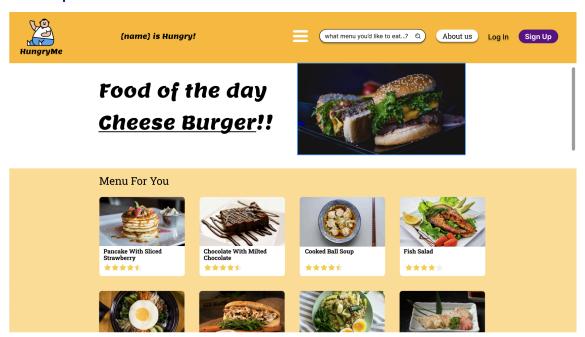
 Sometimes the suggested food contains the ingredients that visitors might not like. We provide the customizing feature to filter the conditions such as ingredients, allergies, rating and calories for the visitor to narrow the scope of food they would like to eat!

 Some visitors who are authenticated are allowed to rate and review the menu. So, he/she can share their opinion about that food then which will help others to consider trying the food

 HungryMe would like to have an area for visitors to share their ideas for the community. We provide the Community's Recipe which is the food recipe created by the visitor

 Users who write their menu. They can see and edit their menu in "My Cookbook".

Example UI:

PERSONALIZE YOUR EXPERIENCE

Do you have any food allergies?

WHEAT-FREE   DAIRY-FREE   PEANUT-FREE   TREE NUT-FREE   SULFITE-FREE   SOY-FREE   SESAME-FREE   SEAFOOD-FREE

Next

Menu For You

Pancake With Sliced Strawberry

Chocolate With Milted Chocolate

Cooked Ball Soup

Fish Salad

HISTORY

Bibimbap

Sandwich

Rice With Zucchini, Soft Boiled Egg, and Parsley

Fried Dumplings

Pancake With Sliced Strawberry

Chocolate With Milted Chocolate

Cooked Ball Soup

Fish Salad

Login

Login with Google

Or

username*

password*

☐ remember me   Forget password?

Login

not registered yet? Create account

HungryMe

Implementation:

Programming Languages: Python, HTML, CSS
Framework: Django
CSS Framework: Bootstrap
API source: https://rapidapi.com/apidojo/api/yummly2

Features:

Main Features:

- Show the Suggested food.
  - filtered by ingredients
  - filtered by allergy
  - filtered by user's rating
  - filtered by calories
- User authentication by OAuth.
- Users can rate and review the menus.
- Display the top rated dishes chart.
- Display a list of menus for you.
- Users can add their recipes.

Optional Features:

- Show the Suggested food of the day.
  - Random tasty dish that came up with the description.
- Show tags of the menu such as low calorie, main dish, appetizer.
- Show search history.
- Show a recently viewed menu.
- Users can add their favorite menu to their list of fav meals.
- Users can see his/her added menu.

Development Plan:

Iteration No.	Iter. dates	Major Work and Goal(s)	Milestones
1	10 Oct - 15 Oct.	1. Self study of using tools - Figma - Django - HTML, CSS - Bootstrap 2. Design example UX/UI	- Main user stories written, domain model approved by TA. All docs on project wiki.
2	17 Oct. - 22 Oct.	1. Homepage 2. Display a list of menus for you. 3. Anyone can search for the menu on navigation bar	- Home page can be visited. - List of menu shown on homepage. - Users can search for the menu by navigation bar.
3	24 Oct - 29 Oct.	1. Users can login, logout and sign Up. 2. Add a link to "Login" to the foods index page	- Users can sign up, login, logout.

4	31 Oct. - 5 Nov.	1. Login with Google(OAuth)
		2. Register with Google(OAuth)
		3. Improve the navigation and page layout.
		4. Externalize configuration data for security and portability.
		5. Show the Suggested food
		   - filtered by ingredients
		   - filtered by allergy
		   - filtered by user's rating
		   - filtered by calories
		6. Users can rate the menu and review the suggestions menu.
		7. Users can't rating again if he is already voted
		8. Everyone can see the rating score of each menu.
			- Suggestion feature works.
			- Authenticated users can rate and review the suggestions menu.
			- An authenticated user can rate only once.
			- Rating score visible for everyone.

5	7 Nov. - 12 Nov.	1. Build Top Rated-Dished 2. Build My Cookbook page 3. Build History page 4. Community's Recipe	- Pages can open from the External content button.

		5. Embed pages on the External content button on the navigation bar.	
6	14 Nov - 19 Nov.	1. Deploy web application. 2. Receive feedback to improve Web App.	- We don't have to runserver every time we want to use the Web App. - Get feedback from users.
7	21 Nov - 26 Nov.	1. Improve the user's experience. 2. Implement the optional features.	
8	28 Nov - 2 Dec.	1. Must succeed all main Feature and Add documentation.	- Add documentation to the github repository. - Succeed all main features.

Note: 30 Nov - 1 Dec Project Presentations
1 Dec - 6 Dec Fix remaining defect, polish site documentation

Major risks:

- **Time** -The different work time for each teammate.
- **API Database -** Our group has never used an API before.
- **Project Cost** - We might have to pay for exceeding the limited api request and server.
- **Data** - We can't control the data because we receive it from API.

Team Members:

Name	Student ID	Github ID
Kollawat Rupanya	6310545221	*zevenfox*
Kodchakan Prajob	6310545213	*jaybjackie*
Sittanat Palakawong Na Ayudthaya	6310545400	*jaofourthsOhappy*
Woraphan Sutthiboon	6410545584	*Unikorn996*

Software Process:

Our team will use Agile & Scrum with an iteration length of one week, starting on Monday and ending on Saturday. Team will meet every Monday and Thursday.

Retrospective will be held on Saturday. A summary of retrospective with a plan for process improvement will be recorded in the project wiki.

Process Component and tools:

- Configuration management: Github and Github Flow; Master branch for a deployable project, child branch for new features or fixing bugs.
- Documentation: GitHub/Google docs.
- Issue Tracking: Github for issue tracking.
- Communication: Discord
- Task management: Github Project
- Visibility: in the project README.md on Github, include links to the task board, documentation (esp. iteration plans), issue tracking (Github), and communication channel.
- Team Roles:
  - Kollawat : Product Owner, Developer
  - Kodchakan : Scrum master, Developer
  - Sittanat : Developer
  - Woraphan : Developer
- Quality Assurance: Use unittest for testing in each iteration
- Continuous Integration Server: Github Action
- Deployment: pythonanywhere / Heroku