Google Summer of Code

Proposal - Google Summer of Code 2022

Pushing the Hunger Games annotation engine to the next level

By
Sumana Basu

Table of Contents

Sumana Basu
Email address: sumana.basu2001@gmail.com
Open Food Facts ID: sumana2001
Slack ID: U036KP3GP8V
GitHub ID: sumana2001
LinkedIn: sumana-basu

Pushing the Hunger Games annotation engine to the next level

Mentor:

Alexandre Fauquette

Overview:

The stakeholders of Open Food Facts consist of food consumers, food producers, government organizations, scientists and researchers. They all have different purposes for using OFF but the game-changer is data. All the data collected can be used for research, deciding the best health policies, and even to make better-informed food choices for people all around the globe. The Hunger Games leverage the results of the OCR used in OFF to validate more data and annotate new things to help train machine-learning models. The aim of this proposal is to add many more features and improve the UI as well as UX to retain contributors. Retaining contributors would ultimately, increase the labelling of data, resulting in better clusterisation and hence a more efficient model in terms of accuracy.

To get data and statistics about the contributions made by the volunteers, we need to assure that they have proper credentials and are logged in to OFF. But users tend to bounce back from the website if they have to directly create an account without knowing what the software does. The Hunger Games is mildly addictive and fun to play, so I propose to prompt users after say 10 questions to log in or sign up.

Moreover, I plan to integrate the OFF-events leaderboard, badges, and more into the hunger games application which will create a metaverse environment for OFF. I will also add levels that will give the users a strong sense of satisfaction. To improve the usability of mobile devices, modify the hunger games into a Progressive Web App (PWA). Developing a PWA would reduce network calls in total hence, reduce the bounce back and idle period for the contributors before their period. With the help of PWA, notifications can be sent at ease and let community members know about the initiatives of OFF.

In addition to that, I plan to add new features like profile pages for contributors to showcase achievements, giving options for undoing the user's actions, reporting and providing feedback to OFF, more autocompletion and filters, and so on. All these features will provide a pathway to having more reliable data about all food products.

The 4 Milestones:

1. UI Revamp:

a) Revamp the navbar to make it more user-friendly adding more items to it and removing the less used ones.
b) Redesigning the Welcome pop-up to explain each feature in the Navbar in a more interactive
c) Create cards to select the game the user wants to play.
d) Add tooltips and toast messages.
e) Add a Favicon and link on the logo that redirects to the main website.
f) Create a path from the main website to the hunger games website to attract more contributors.
g) Improve documentation of both the Github Repository and the Wiki Page for enhancing contribution workflow.
h) Improve the content of the website to make it more friendly and more informative.
i) Prompt users to contribute more and download the app.

2. Gamification:

a) A leaderboard based on the number of contributions which can also be filtered country-wise.
b) A level-based system that prompts users to reach the next level by showing how many more contributions the user needs.
c) Badges like first contributor, first scanner, fact-checker and so on which can also be shared on social media.
d) A personalized profile page showcasing all their profile details and achievements.

3. UX Improvements:

a) Make the application mobile responsive
b) Convert the application into a PWA
c) Create an option to undo the user's last action
d) Add a report/flag button for images
e) Add auto-completion for Logos and Label game to get more accurate data
f) Add filter for store
g) Add a category mode for products which don't have a category at all
h) Save preferred filters of the user
i) Add a 404 page

4. Authentication:

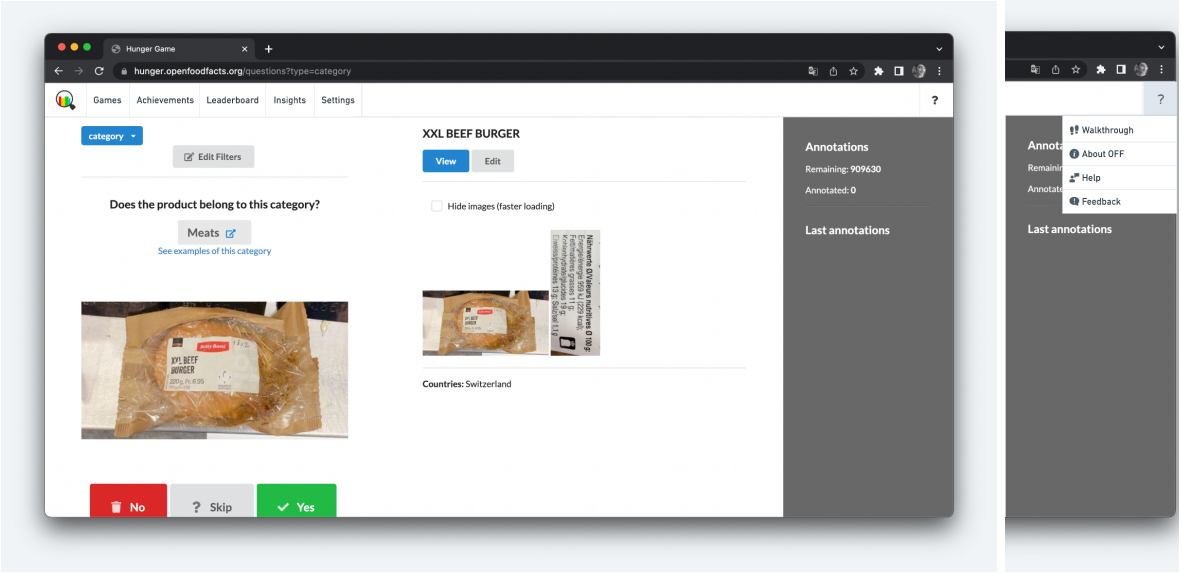
a) Check whether the user is logged in or not
b) Let users try a few questions without signing in and then prompt the user to login/sign up
c) Login system within the hunger games website

How I plan to achieve the milestones:

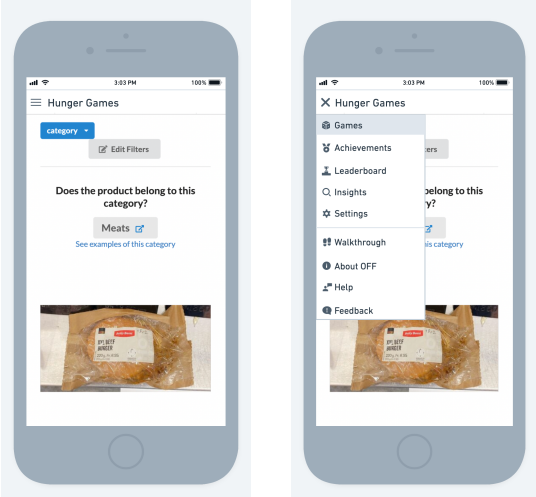
1. UI Revamp:

a) **Navbar:** As we will be integrating a lot of new features, adding new games etc., the navbar would become very long if we add all the games separately. I propose we make one view for Games where the user can select which game they wish to play. To provide the user with more help, I have added sections like open food facts, help, and feedback. This is how the new navbar will look like

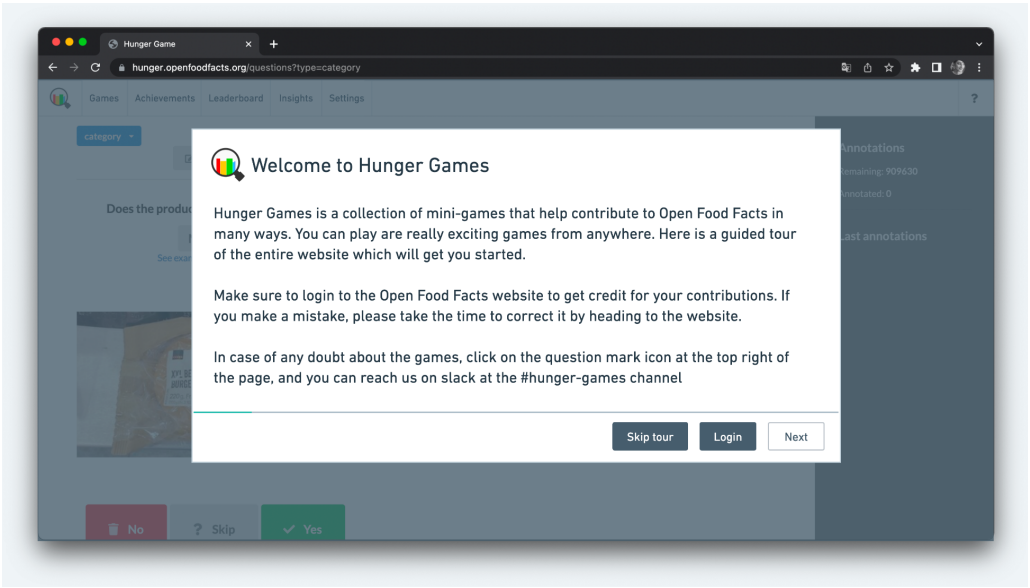
Laptop screens:

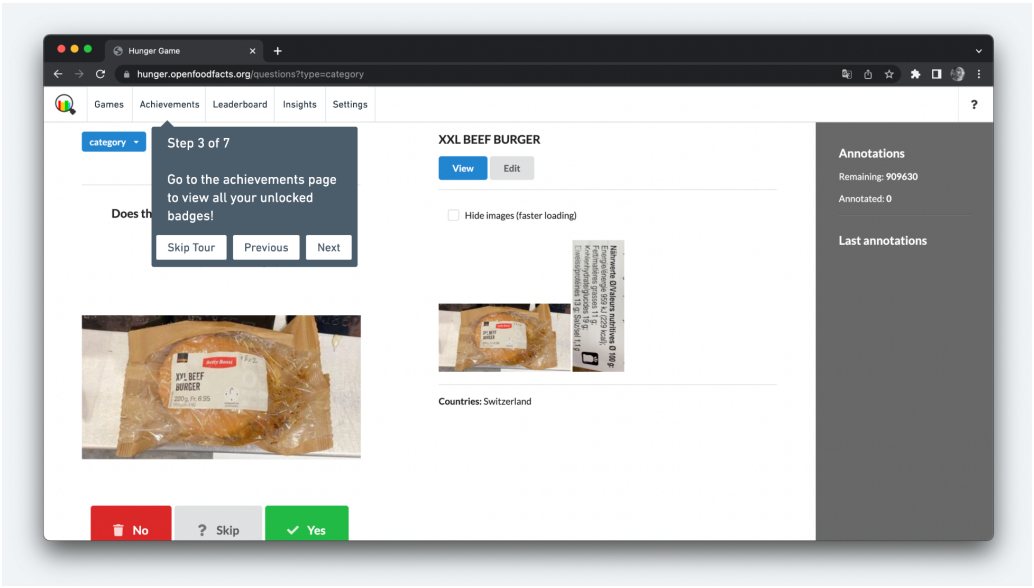


Mobile screens:



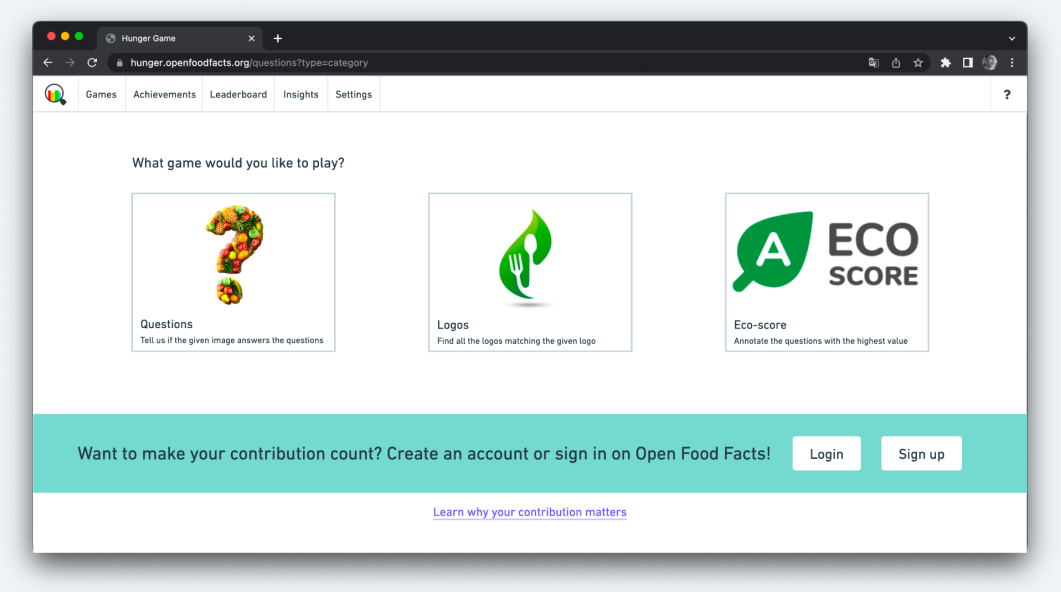
b) **Welcome Pop-up / Walkthrough:** Currently the welcome pop-up shows talks only about the two games - questions and logo game. To make it more user interactive, we can create a guided tour of the website using open-sourced npm packages like **Vue-Tour**. We will use tooltips to display each section of the website and of courses, there will be an option to skip or go through the walkthrough later.



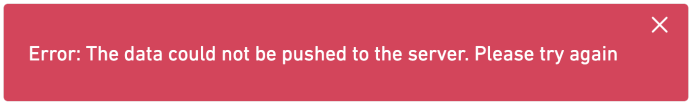


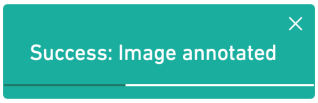
c) **Cards for games:** Since there are many games that are in the beta phase and will be soon in production, instead of crowding the navbar it's better to have a separate view from where users can select which game they wish to play. I have furthermore added a prompt to signup/log in to encourage more and more users to create an account in OFF. Lastly, we could add a small pop-up where we tell

users exactly why their contribution matters so much to motivate them to contribute further. Here's what the wireframe looks like:

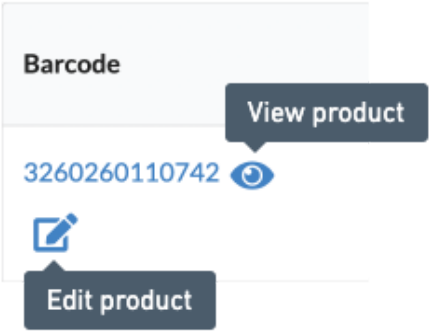


d) **Tooltips and Toast messages:** Once a user annotates something, or submits the value of the logos, no message is given so that the user can know if the data was received at the backend. To solve this problem we can use toast messages. For most messages, we can use timeouts to show the message only for a few seconds, and for the important ones like in case of a server error we could add a toast without a timeout, which must be dismissed by the user.





To provide more help to the user we can also add tooltips to the button. Tooltip/info tip or hint is a GUI element in which, when hovering over a component, a text box displays information about that element.



e) **Add a Favicon and Link on the logo:** The hunger games website is very hard to locate if users have multiple tabs open. To fix this issue we can add the OFF logo as a favicon. Moreover currently if a user reaches the hunger games website, there is no way they can reach the OFF main website from here. So the logo at the top-left corner of the navbar should have a link to the main website. This way we can increase awareness about OFF.

f) **Linking between OFF's main website and hunger games:** From the main website of OFF, it was very hard for me to find a way to hunger games. To attract more and more contributors, we should add a link from the main website which redirects users here. It can be a pop-up saying “wish to contribute and have some fun? Join us at hunger games.” We can also add a section about the hunger games on the **contribute** page of OFF.

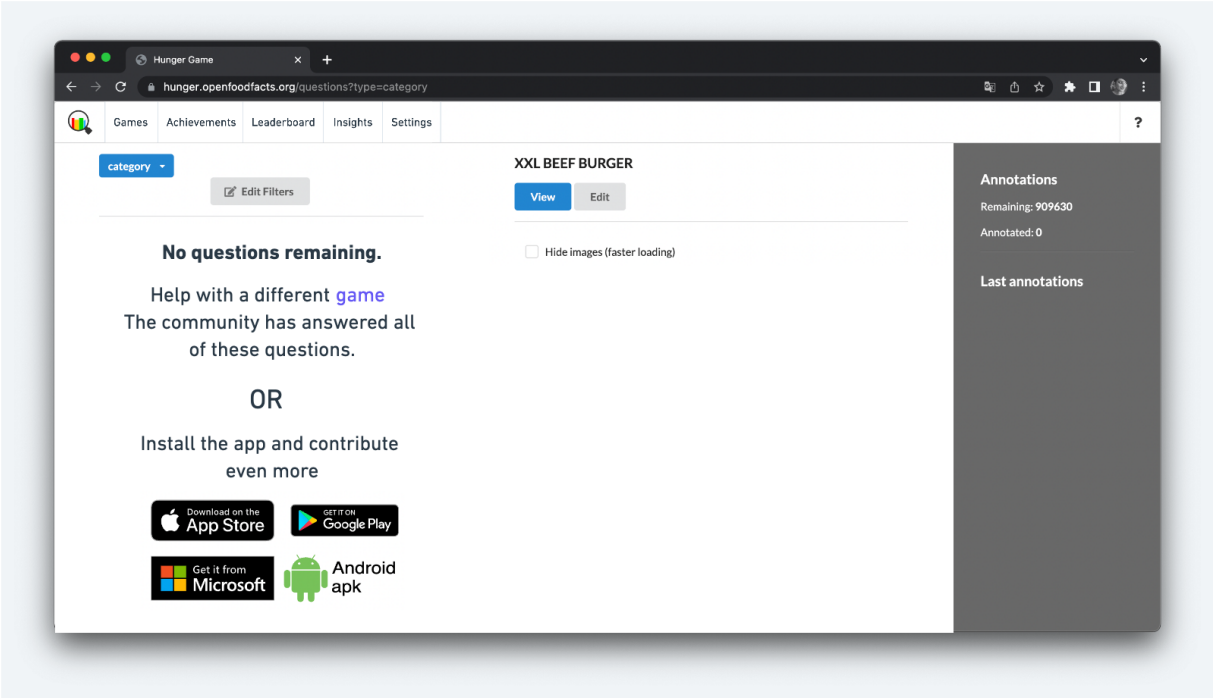
g) **Improve documentation:** In both the **GitHub** repository and the **Wiki** page, don’t have much information about what these games are. As a technical contributor who is on the Github page for the first time, it is difficult to understand what all games are there, what they are expected to do, what is the end result and how they can help. Also, the roadmap and development guide in French is not open and needs permission to access which considering it an open-source project should not be there.

   We could first create different sections for different games in both the README.md as well the wiki page where we can explain each game and add GIFs or videos to show how the game can be correctly played. In addition to that, I feel it is important to mention Robotoff in the GitHub readme because all the functionality and API calls are from Robotoff.

h) **Improve the content of the website:** There are some places in the website where I feel the content can be modified to a more friendly and informative one. For example, not everyone knows about what an eco-score is. Similarly, we could add 1-2 lines about each game stating what you need to do at the top of their respective page. Lastly, when there are no more questions remaining, instead of printing that we could add a text which tells them ways to contribute further. For

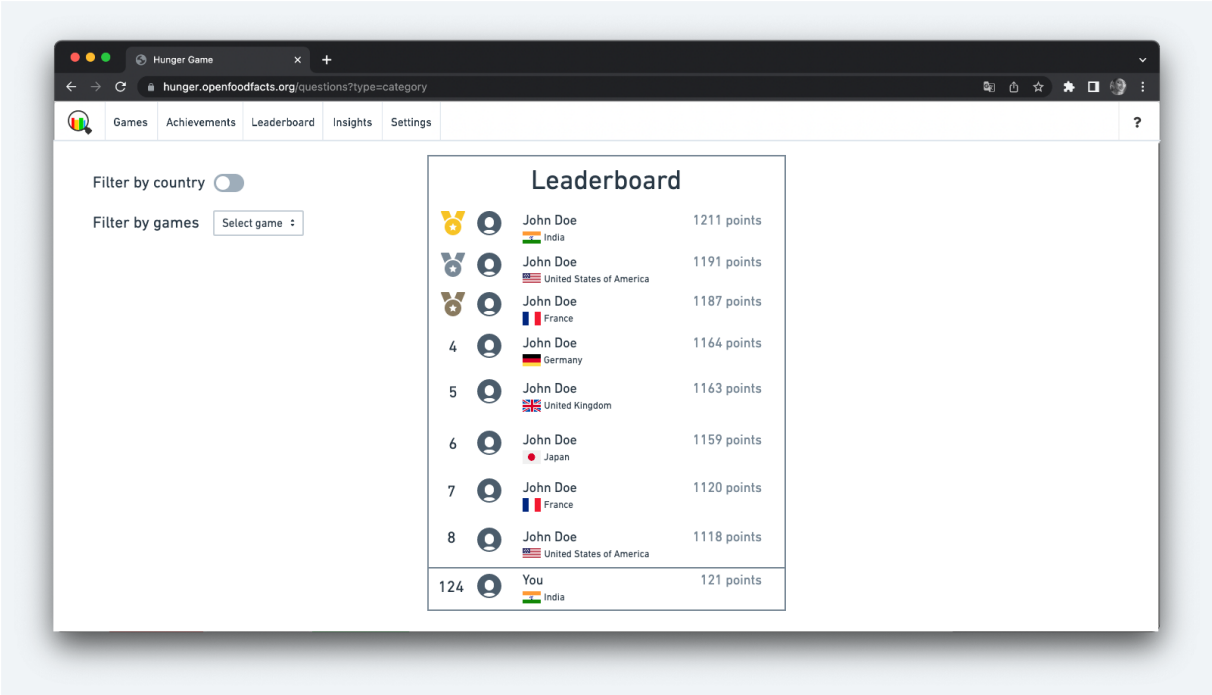
example: “Help with a different game. The community has answered all of these questions.”

i) **Prompt users to download the app:** We should prompt users to contribute more. One of the ways is to redirect our audience from hunger games to the app so that we can get more and more data. When there are no questions remaining, we could add a component that asks the user to install the app. We could also add it to the welcome popup.



2. Gamification: A major reason why games like “Candy Crush” are so addictive is that it is super easy to play and have increasing levels which unlock new worlds and take you higher on the leaderboard. We can make the same happen for hunger games.

a) **Leaderboard:** To create a healthy competition for the users and make sure they keep coming back to play and contribute, we can create a leaderboard that can also be filtered by country. Since we do not ask for the user’s country during signup, we could use geolocation of IPs to get their countries. Here’s a mockup of how the leaderboard would look



I will be using openfoodfacts-events to integrate the pre-built API to get the leaderboard. In openfoodfacts-events, an event is any action taken by the user like adding a product, editing a product, answering questions, matching logos, and so on.

The endpoint for all API calls to the Open Food Facts events to be used to gamify is

https://events.openfoodfacts.net/

Currently a get request to “**/events**” returns an output similar to this:

```
[
  {
    "event_type": "invite_shared",
    "timestamp": "2021-10-27T15:59:13.532437",
    "user_id": "ocervell",
    "barcode": null,
    "points": 10
  },
  {
    "event_type": "product_scanned",
    "timestamp": "2021-10-27T15:59:13.532437",
    "user_id": "ocervell",
    "barcode": "123456789",
    "points": 10
  }
]
```

But in Robotoff we register annotations “username”. To remove this discrepancy, I will add another key-value pair in the openfoodfacts-events’ events being “username: ocervell”

So the new response would look like this:

```
[
  {
    event_type: 'invite_shared',
    timestamp: '2021-10-27T15:59:13.532437',
    user_id: 'ocervell',
    barcode: null,
    points: 10,
    username: 'ocervell',
  },
  {
    event_type: 'product_scanned',
    timestamp: '2021-10-27T15:59:13.532437',
    user_id: 'ocervell',
    barcode: '123456789',
    points: 10,
    username: 'ocervell',
  },
]
```

Now I will add more event_types for different games. For uniformity they will be named as per the application they belong to. For example robotoff_logo, robotoff_questions

Looking at the points allocated to the existing events, this is the point system I would follow:

1. Questions game: 5 points
2. Logo game: 10 points
3. Eco score: 6 points (Extra 1 point for high value)

The endpoint and response of the leaderboard from the openfoodfacts-events API are:

https://events.openfoodfacts.net/leaderboard

The leaderboard can be filtered by game by sending a get request with parameters:

https://events.openfoodfacts.net/leaderboard?event_type=product_added

b) **Levels:** If the users get tired of playing the game, what will motivate them and keep them going? For this, I plan to add a level-based system. The levels will have a similar system to the leaderboard:
Every new account will start at Level 0. For each game, the points will be

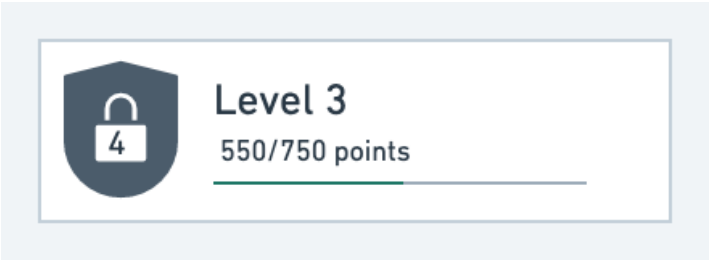
1. Questions game: 5 points
2. Logo game: 10 points
3. Eco score: 6 points (Extra 1 point for high value)

The levels will have increasing difficulty. So the number of contributions you need to make to pass each level will increase 2x. For example to cross each level the number of points you need to have are:

1. Level 1: 50 points
2. Level 2: 150 points (50*2 = 100)
3. Level 3: 350 points (100*2=200)
4. Level 4: 750 points (200*2=400)
   and so on

Each time you pass a level you get a badge saying “Level _ contributor”.

Level 3 current points



Level 3 contributor badge

c) **Badges:** The endpoint for the badges in the already existing openfoodfacts-events API for a particular user is:

https://events.openfoodfacts.net/badges?user_id=ocervell

The response received is:


```
[
  {
    "user_id": "ocervell",
    "badge_name": "First invite",
    "level": 0
  },
  {
    "user_id": "ocervell",
    "badge_name": "First scan",
    "level": 0
  },
  {
    "user_id": "ocervell",
    "badge_name": "First product added",
    "level": 0
  },
  {
    "user_id": "ocervell",
    "badge_name": "Growth Hacker",
    "level": 1
  },
  {
    "user_id": "ocervell",
    "badge_name": "Furious Scanner",
    "level": 1
  },
  {
    "user_id": "ocervell",
    "badge_name": "Product Hunter",
    "level": 1
  },
]
```

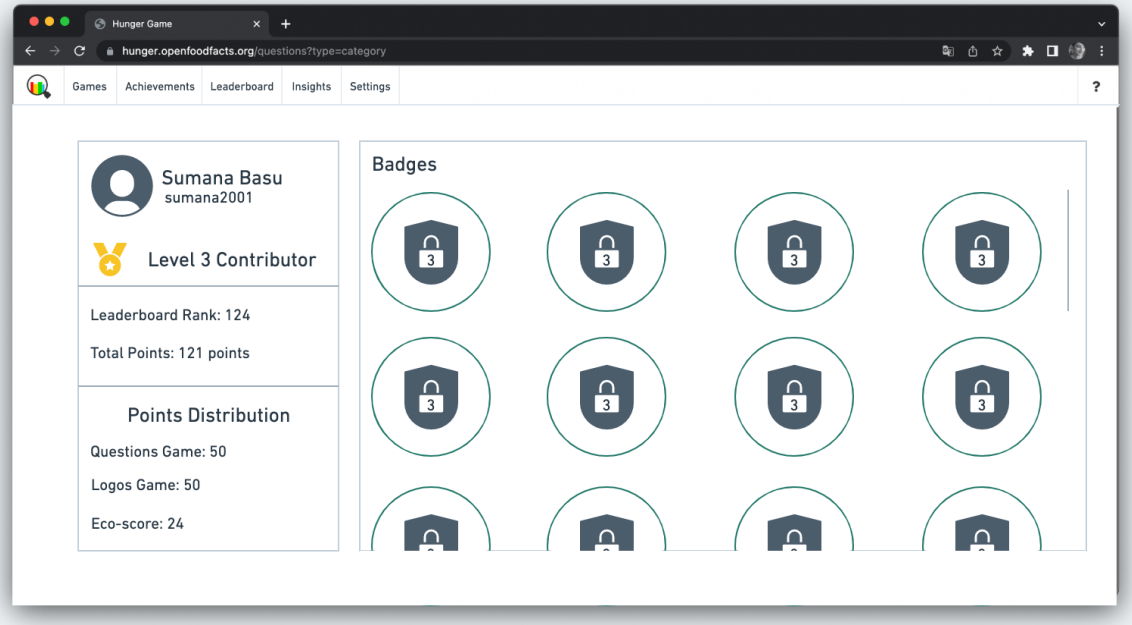

The badges will be showcased on the Achievements view sorted by level of difficulty given by the “level” key. We could add more badges to it focusing on the hunger games. Some of the badges could be

1. First contribution
2. Question Scorer
3. Logo Hurdler and so on..

d) **Achievements page:** To showcase the amazing work the contributor has done, they could have a public page showing all their badges, ranks, and points. The link can then be shared everywhere. The URL will be in the format

https://hunger.openfoodfacts.org/user_id/achievements

We will make API calls to openfoodfacts-events to get the leaderboard rank, total points, points of each game, and badges. We will also have to get the name of the user from **user.pl** in openfoodfacts-server.



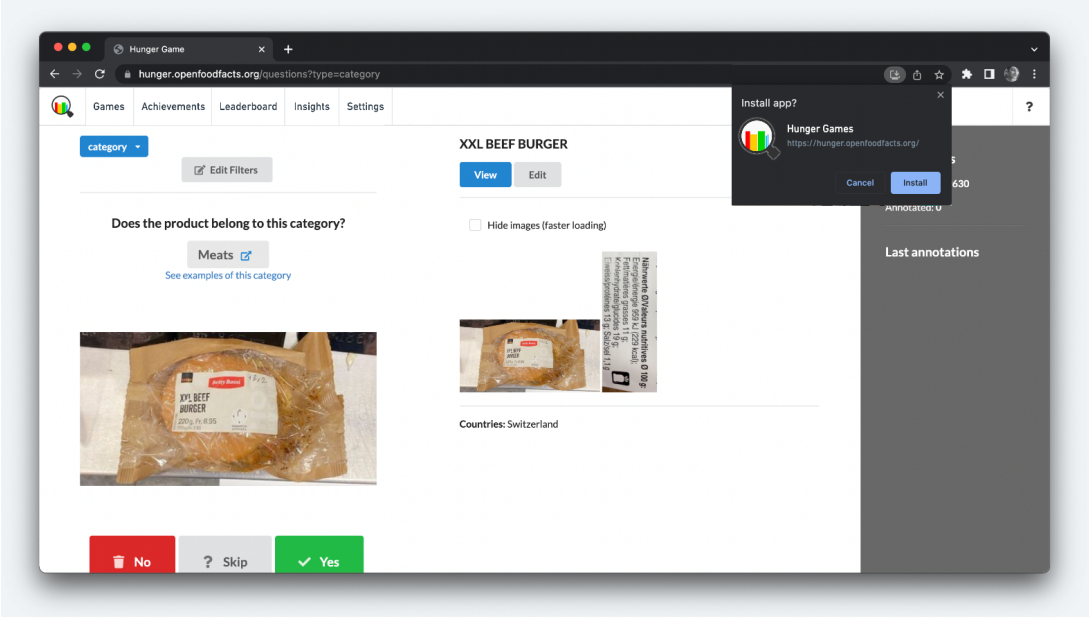
3. UX Improvements:

a) **Mobile responsive:** In the hunger games website, the major issue with why it is difficult to play on mobile phones is that you have to always scroll down to answer questions. I plan to tweak the UI so that the entire question with the YES/NO/SKIP buttons is visible on the same screen. Along with that, I noticed some places where the UI could be improved on mobile screens by adding extra spaces, enlarging the font, and so on. With the world turning towards a mobile-first approach this fix would highly increase the user-retention hence more contribution.

b) **PWA:** Progressive Web Apps are the solution most suitable for the mobile-first era. By converting hunger games into a PWA, we can make it usable from a mobile device so that users can use a 10-minute time gap while waiting in a queue or commuting. There are so many benefits of a PWA:

1. short loading time
2. good performance in poor network conditions
3. small size
4. app-like features (add to the home screen, offline mode, push notifications)
5. avoid app aggregators (Google Play, App Store, etc.)
6. instant updates and the list goes on and on

Here is what the download button would look like:



For converting the website into PWA, I'll have three main steps:

1. serviceWorker.js: A JavaScript service worker (to allow the site to load offline and store data locally)
2. manifest.json: A valid JSON manifest of the app's info, with the correct info filled in
3. Icons: A set of properly named icons, of multiple sizes

c) Undo: If by chance, the user makes a mistake and realizes it immediately, currently the user will have to go to the OFF website, find the exact product and edit the details. I plan to add an undo button for instant corrections. The undo button can be used to cancel the post request till it reaches the server and stores the changes.

For this purpose, I will first generate a cancelToken and store it.

```
import axios from 'axios'
const request = axios.CancelToken.source();
```

Then pass the cancelToken to the Axios request

```
axios.get('API_URL', { cancelToken: request.token })
```

Lastly, access the request you stored and call the .cancel() method to cancel it.

```
request.cancel("Undo");
```

d) Report/flag button: I will create a button to report images that are irrelevant like pictures with humans in it. Then I will call **notify_image_flag** from Robotoff's slack.py to check if the flagged image is actually sensitive or irrelevant. If the image is correctly flagged, an alert will be sent to the

ROBOTOFF_PRIVATE_IMAGE_ALERT_CHANNEL. The notify_image_flag calls – to check if the image flagged is sensitive:

```python
def _sensitive_image(flag_type: str, flagged_label: str) -> bool:
    """Determines whether the given flagged image should be considered as sensitive."""
    is_human: bool = flagged_label in {
        "face",
        "head",
        "selfie",
        "hair",
        "forehead",
        "chin",
        "cheek",
        "tooth",
        "eyebrow",
        "ear",
        "neck",
        "jaw",
        "nose",
        "facial expression",
        "glasses",
        "eyewear",
        "child",
        "baby",
        "human",
    }
    return (
        is_human and flag_type == "label_annotation"
    ) or flag_type == "safe_search_annotation"
```

e) **Auto-completion for Logos and Label game:** In the logos and label game, the user is supposed to select the same logos/labels and write the value of their brand/category/store, etc. Typing can lead to human errors like spelling mistakes, so I propose adding auto-completion to the game. This way as soon as the user types a letter, possibilities of the word would appear in a dropdown.

To get the data i.e. list of brands or categories or labels or stores etc., I will be making get requests and using it to get the “name” field in each of the below endpoints:

1. https://world.openfoodfacts.org/brands.json
2. https://world.openfoodfacts.org/cgi/nutrients.pl

3. https://world.openfoodfacts.org/stores.json
4. https://world.openfoodfacts.org/packagings.json
5. https://world.openfoodfacts.org/categories.json
6. https://us.openfoodfacts.org/labels?json=true

Then I will pass the data in an array to the autocomplete component. Here's how I have implemented the autocomplete functionality:

```
name: 'SearchAutocomplete',
props: {
  items: {
    type: Array,
    required: false,
    default: () => [],
  },
  isAsync: {
    type: Boolean,
    required: false,
    default: false,
  },
},
data() {
  return {
    isOpen: false,
    results: [],
    search: '',
    isLoading: false,
    arrowCounter: -1,
  };
},
watch: {
  items: function (value, oldValue) {
    if (value.length !== oldValue.length) {
      this.results = value;
      this.isLoading = false;
    }
  },
},
mounted() {
   document.addEventListener('click', this.handleClickOutside)
},
destroyed() {
   document.removeEventListener('click', this.handleClickOutside)
},
```

```
methods: {
   setResult(result) {
     this.search = result;
     this.isOpen = false;
   },
   filterResults() {
     this.results = this.items.filter((item) => {
       return item.toLowerCase().indexOf(this.search.toLowerCase()) > -1;
     });
   },
   onChange() {
     this.$emit('input', this.search);
     if (this.isAsync) {
       this.isLoading = true;
     } else {
       this.filterResults();
       this.isOpen = true;
     }
   },
   handleClickOutside(event) {
     if (!this.$el.contains(event.target)) {
       this.isOpen = false;
       this.arrowCounter = -1;
     }
   },
   onArrowDown() {
     if (this.arrowCounter < this.results.length) {
       this.arrowCounter = this.arrowCounter + 1;
     }
   },
   onArrowUp() {
     if (this.arrowCounter > 0) {
       this.arrowCounter = this.arrowCounter - 1;
     }
   },
   onEnter() {
     this.search = this.results[this.arrowCounter];
     this.isOpen = false;
     this.arrowCounter = -1;
   },
},
```

Now when the user types anything in the “Value” text field, a dropdown will appear, which will show all the possible results.

f) **Filter for store:** We already have a filter for the country, brand, and proposed value. We can also add a filter for stores for users to choose the stores and products they know and are confident about which will result in more accurate annotations.

For this, we will be using the same **ProductInsight** model which has a field called “value”. Value stores the value of the insight, for example, the brand name for a product or the numeric value of the weight of the product. Some insight types will populate both this field and the 'value_tag' field above, some only this field. This field is set for the following insight types: brand, expiration_date, packager_code, packaging, product_weight, and **store**.

```
value = peewee.TextField(null=True, index=True)
```

From the frontend we’ll pass the store as a parameter in the **questions** function:

```
questions(sortBy, insightTypes, valueTag, brands, country, store, count = 10, page) {
  const lang = getLang();
  return axios.get(
    `${ROBOTOFF_API_URL}/questions/${sortBy}`, {
    params: removeEmptyKeys({
      page, count, lang, insight_types: insightTypes, value_tag: valueTag, brands,
country, store
    })
   }
  ).then(result => {
    let questions = result.data.questions;
    result.data.questions = questions.filter(question => question.source_image_url);
    return result;
  })
}
```

Now we can get that back as a parameter in Robotoff in **InsightCollection**:

```
store: Optional[str] = req.get_param("store")
```

Then we include it in **get_insights** in InsightCollection:

```
get_insights_ = functools.partial(
  get_insights,
  keep_types=keep_types,
  country=country,
  server_domain=server_domain,
  value_tag=value_tag,
  brands=brands,
  annotated=annotated,
  annotation=annotation,
  barcode=barcode,
  store=store,
)
```

Finally, we add a where clause to be added to the **get_insights** function:

```
where_clauses = [ProductInsight.server_domain == server_domain]

if annotated is not None:
  where_clauses.append(ProductInsight.annotation.is_null(not annotated))

if where_clauses:
  query = query.where(*where_clauses)
```

g) **Category mode for products that don't have a category at all:** There are many products that have been 0% completed. We could add another game/view called “need help” or “to be completed” where we fetch only the products which are incomplete. For this we will use the following endpoint:

https://world.openfoodfacts.org/state/ categories-to-be-completed.json?fields=code&sort_by=popularity

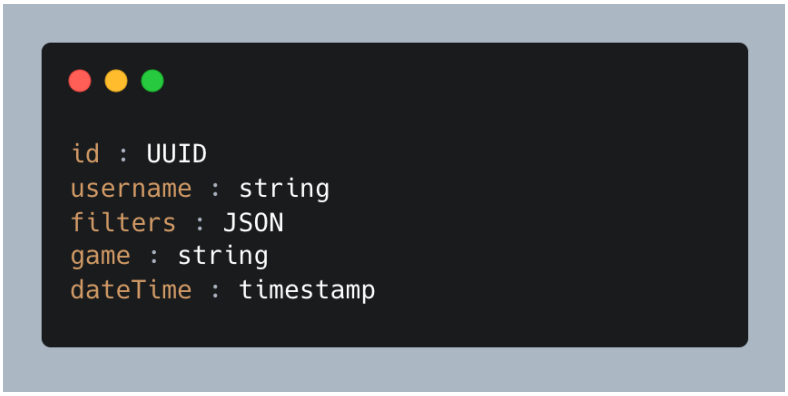
From the bar code we receive a response, we can test if the Robotoff has some insight by sending a request to the following endpoint:

https://robotoff.openfoodfacts.org/api/v1/insights/{id}

It will have the same UI as the questions game and will have the exact same function except that all the products are shown be the ones which currently need a lot of work.

h) **Saving user-preferred filters:** Users who are regular contributors might get annoyed adding their preferred filters on the games every single time. What can we do? We can save them! Now, every time the user will log in, their filter will already be added to the game.

For this, I will create a new table that stores the user's preferred filters. The table will be called "user_saved_filters" with the following structure:

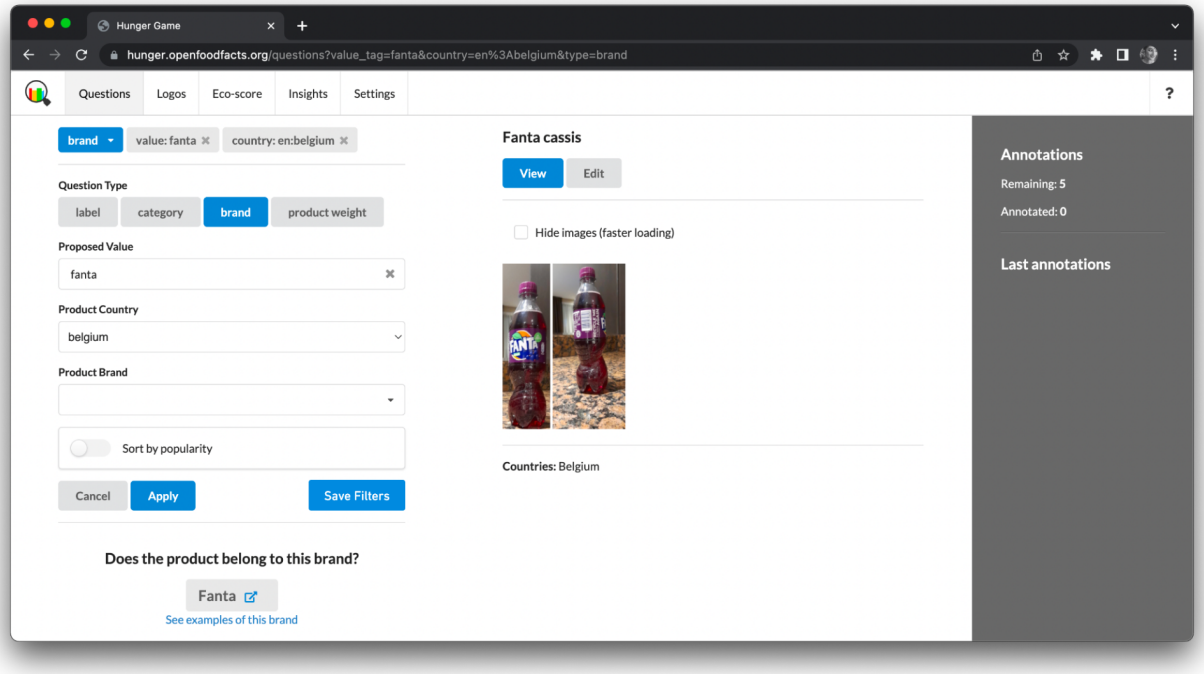


```
id : UUID
username : string
filters : JSON
game : string
dateTime : timestamp
```

An example of the record in the above table will be:

```
{
  "id" : "158542f3-3957-4859-ad66-196a502c1639",
  "username" : "sumana2001",
  "filters" : {
    "value_tag": "fanta",
    "country": "en:belgium",
    "type": "brand",
  },
  "dateTime" : "2022-03-15 21:05:15",
}
```

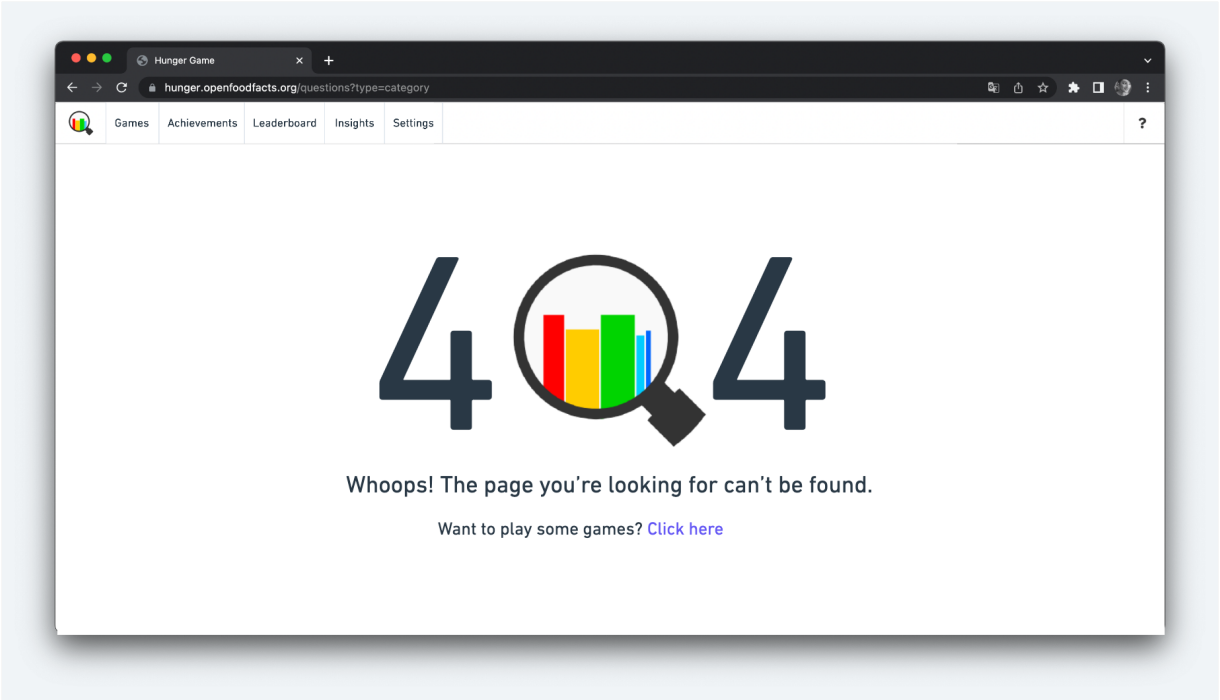
In the frontend, we will add a save button to save the filters that are applied by the user at that moment:



The saved filters will be passed as an optional parameter in the questions function and will be received in the **get_questions_resource_on_get** function in Robotoff.

There we will add an if condition to check if the user has saved filters and there are no current filters added to the game by the user. If that is the case, we will use those filters in the get_insights function. Else it will return the questions with the filters either manually added by the user or with no filters.

i) **Add a 404 page:** A 404 page is a landing page that tells your site viewers the requested page is unavailable or, in some cases, doesn't exist. At present, if a user reaches a page that does not exist, the hunger games website shows the blank page with the navbar. Instead, we can add an attractive view in such cases that tell the user where to go from there.



4. Authentication: I proposed I plan to add a lot of features like an achievements page, leaderboard, user-preferred filters, and so on, for which authentication is essential so that we know which user's data we are getting. Authentication will also help users get credit for their valuable contributions.

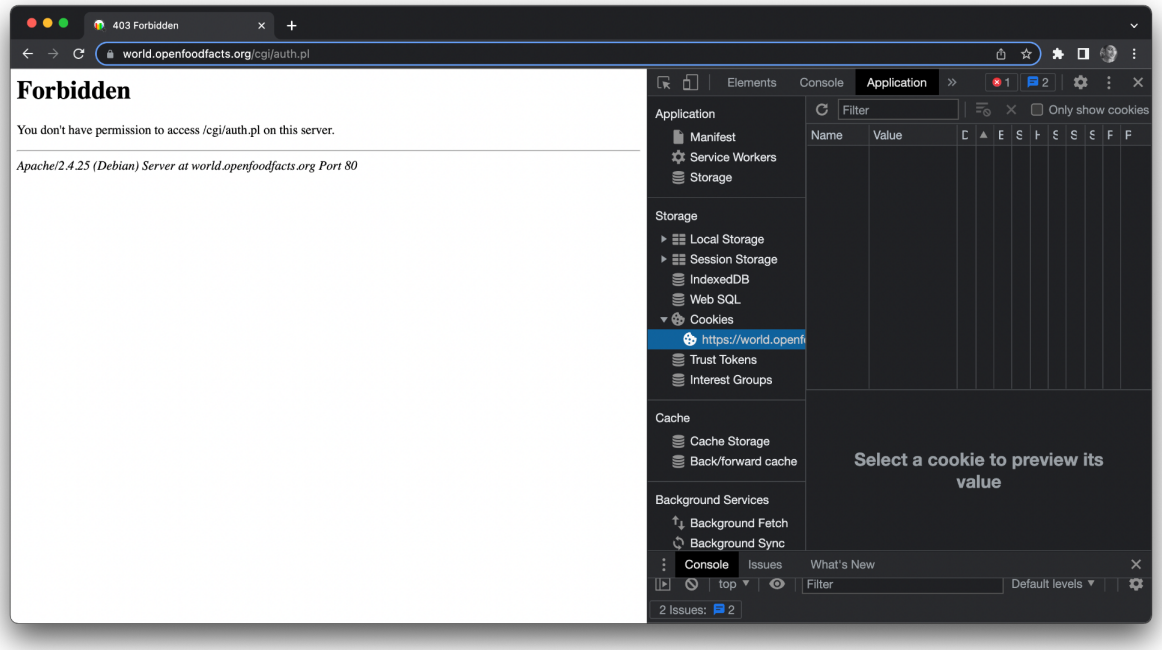
a) **Check if a user is logged in:** Many times in the application we will have to check if the user is logged in or not to show a particular view like the achievements. Else we will have to show them a prompt to signup/login.

To check whether the user is signed in, we will check the session cookie, which is done by the API written in Perl for openfoodfacts-server:

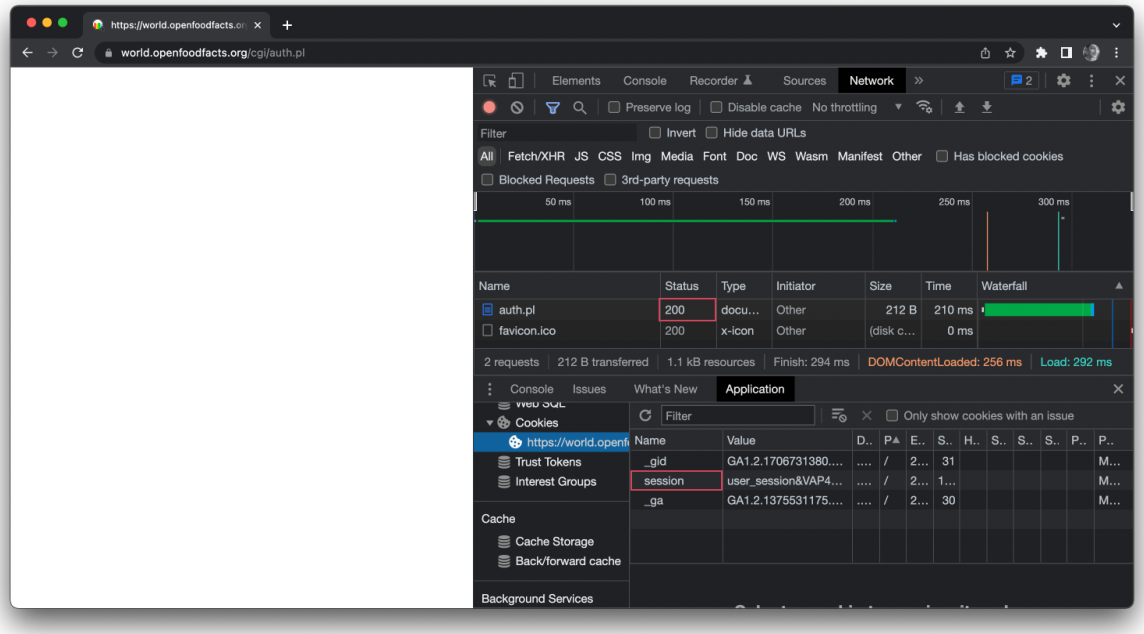
https://world.openfoodfacts.org/cgi/auth.pl

If the user is logged in, it will give a status code of “200”. Else it will give the response with status code “403”.

Not logged in:

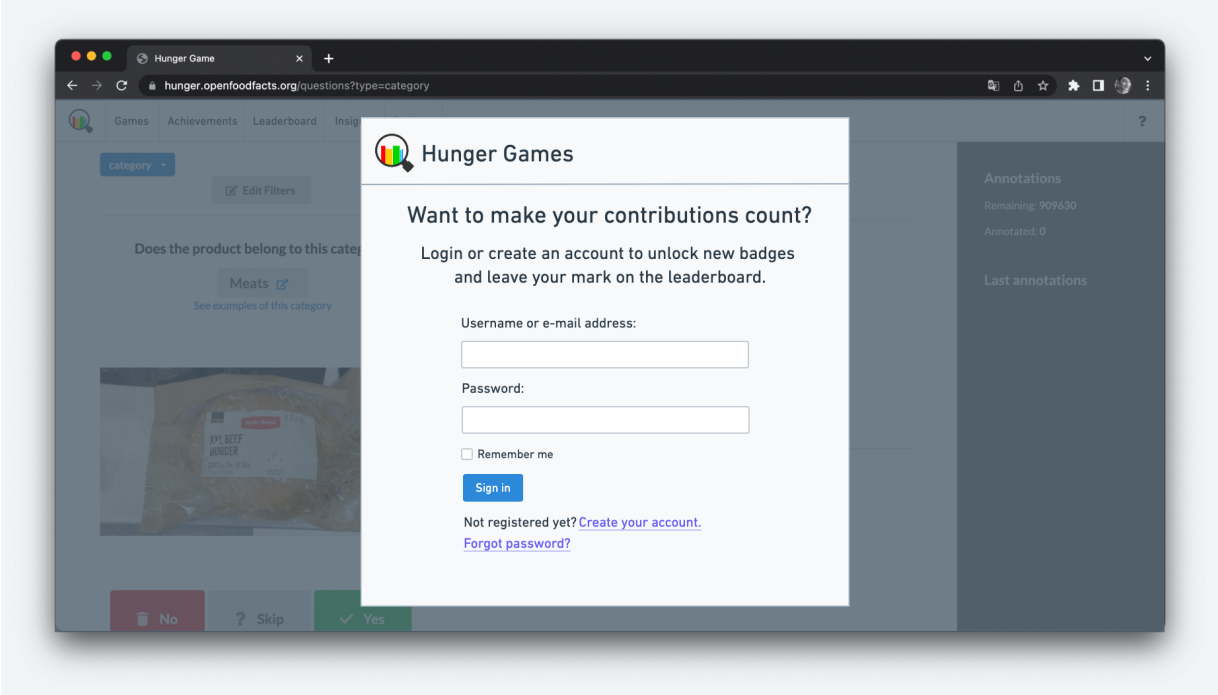


Logged in:



b) **Try a few questions and then prompt:** New users who don't know much about the games might want to get a live experience first, before going through the entire process of creating an account and signing up. For such users, we can let them play a few questions, say 10, and then send a prompt to sign up/login, in to make their contributions count.

It can be done by simply creating a counter which will count to a particular number, in this case, 10, and every time the user annotates, the counter goes down. As soon as the counter reaches zero, we display a pop-up of authentication.



c) **Login system within the hunger games website:** Currently, users have to go to the Open Food Facts website and log in and then make sure they don't clear their cookies in hunger games to stay logged in hunger games. Instead, we can simply have the login system inside the hunger games website by adding popups using an iframe. But before this can be added there is a small **issue** regarding CORS for hunger games. We will be using the same UI as the Open Food Facts main website. The APIs to be used are from openfoodfacts-server:

1. Sign-in: **https://world.openfoodfacts.org/cgi/session.pl**
2. Sign-up: **https://world.openfoodfacts.org/cgi/user.pl**

Plan of Action: Timeline

19th April - 20th May

a) Getting familiar with the code
b) Fixing bugs
c) Resolve **#394**
d) Resolve **#369**
e) Resolve **#42**
f) Resolving bugs that arise while resolving the above issues
g) Add a Favicon and link on the logo that redirects to the main website
h) Prompt users to contribute more and download the app
i) Create a path from the main website to the hunger games website to attract more contributors
j) Build a more concrete understanding of OFF's work and the hunger-games codebase
k) Discuss other project approaches with a mentor

20th May - 12th June

a) Community Bonding
b) Integrate auth.pl to check if the user is authenticated
c) Login system within the hunger games website
d) Let users try a few questions without signing in and then prompt the user to login/sign up
e) Make the application mobile responsive
f) Convert the application into a PWA
g) Improve documentation of both the Github Repository and the Wiki Page for enhancing contribution workflow
h) Improve the content of the website to make it more friendly and more informative

13th June - 29th July

a) A leaderboard based on the number of contributions which can also be filtered country-wise
b) Badges like first contributor, first scanner, fact-checker and so on which can also be shared on social media
c) A level-based system that prompts users to reach the next level by showing how many more contributions the user needs
d) A personalized profile page showcasing all their profile details and achievements.
e) Create cards to select the game the user wants to play.
f) Revamp the navbar to make it more user-friendly adding more items to it and removing the less used ones.
g) Redesigning the Welcome pop-up to explain each feature in the Navbar in a more interactive

30th July - 12th September

a) Resolve all the bugs created during the implementation of the 4 milestones
b) Create an option to undo the user's last action
c) Add a report/flag button for images
d) Add auto-completion for Logos and Label game to get more accurate data
e) Save preferred filters of the user
f) Add tooltips and toast messages
g) Add a 404-page
h) Buffer time to resolve issues or feature decided before
i) Write the report for final evaluation

Apart from the vital contributions: Extras

Now this will be a fun time and should not be considered as a part of the Goal.

- Contribute to other ongoing projects
- Complete beta-mini games
- Fix Bugs and build some amazing features
- Try to improve the project readme in general
- Write blogs to showcase the project we are building and what it aims for
- Make a video to showcase how can everyone get benefited from it

- Run lighthouse and fix the corresponding issues to fire up the application and improve the application in terms of Performance, Accessibility, Best Practices and SEO

The timeline might change according to planning or difficulty of the feature implementation or bug resolution. The prime focus would be to keep everything under the assigned timeline and add the extras if time permits. If in case I am not able to finish a particular task during the assigned timeline, I will definitely continue working on it after the GSoC period ends.

What applications/libraries of Open Food Facts will the proposed work modify or create:

This project will be modifying and adding new features to hunger games. It will also include integration with **openfoodfacts-events**. In addition to that, since Hunger Games calls all its functionalities from the **Robotoff API**, the project will also need some modifications in the same. Lastly, to add links to the main website and fix the CORS issue for authentication, **openfoodfacts-server** will also be modified.

What city and country will I reside in during the summer:

I will reside in Bengaluru, Karnataka, India during the entire summer.

What are my expectations from my mentor:

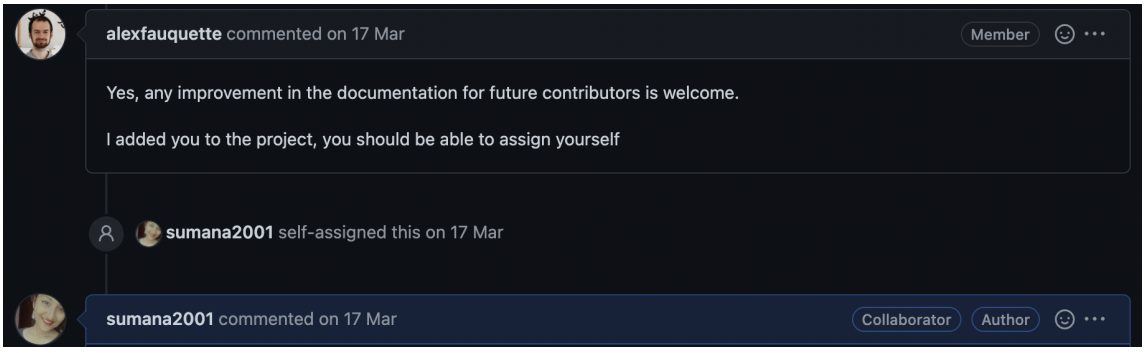
Guide me to understand the existing codebase of the Hunger Games website, terminologies, and other APIs of Open Food Facts wherever I am incapable of doing so on my own. Suggest resources to have a clearer view of how things are done ideally. Help me come to a decision when I have more than one way of doing things and tell me why that is the best decision. Ultimately, take time to review my work and provide their timely insight while helping me in optimizing the code with respect to Time, Space, and CPU utilization complexities.

Why did I choose this project:

I was seeking to do a challenging project this summer, the Google Summer of Code 2022 provides the perfect opportunity. I have contributed to other open-source organizations however after reviewing project ideas, I decided that I should focus on the OFF projects. Once this decision was made and after reaching out to Alexandre Fauquette sir, it proved to be a good fit for reaching my project and learning goals.

OFF's motive is to serve food consumers, food producers, government organizations, scientists and researchers, and other stakeholders, with a better understanding of nutritious and environment-friendly food**,** which I want to join. I always wanted to resolve the problems in the food and beverage industry. Personally, due to poor judgment of apps available on the food items available, I was admitted due to an acute. Being someone who has suffered from the reason OFF is solving, I am particularly interested in taking forward the annotation engine forward. I believe, a good model can only be made with data, but working training a model without labelled data would make the problem worse. With OFF's database consisting of 2,275,399 products each containing so many images, labelling them individually by 4 permanent employees is particularly, thus my proposal helps contributors who are contributing to our database.

Till now, I have made some open-source contributions to organizations but never actually thought of getting into GSoC as a participant. Being a part of the OFF community is not only till GSoC but in the future as well, I will continue contributing to this community. Being very well versed with the application stack OFF uses while making some code contributions to the existing codebase gave me a head start in the learning curve. It also allows for a more immediate impact on the overall OFF Project mission. Furthermore, recently I got access as a collaborator for the project I am applying which ultimately gave me a sense of belonging to the project and the community itself.



What are my past experiences with the open-source world as a user and as a contributor:

I started contributing to opensource a few months back to large organizations. Before that, for more than 1.5 years I maintained projects for my campus club, Webwiz, and another organization Betaoverflow to motivate beginners to contribute to open-source and personal projects on Github. Since I started quite late contributing to Open Food Facts, I have a few merged PRs. I worked on other organizations and personal organization and have mentored more than 1000 people in software development being a mentor in several programs which promotes open-source, writing blogs, delivering talks, and founding and running a community!

My contributions to Open Food Facts' Hunger Games:

Issue	Pull Request	Status
Issue #397	https://github.com/openfoodfacts/openfoodfacts-hungergames/pull/399	Merged
Issue #401	https://github.com/openfoodfacts/openfoodfacts-hungergames/pull/402	Merged
Issue #405	https://github.com/openfoodfacts/openfoodfacts-hungergames/pull/407	Merged
Issue #394	https://github.com/openfoodfacts/openfoodfacts-hungergames/pull/416	Working

Some of the other open-source contributions and projects:

Organization	Link
Major League Hacking	https://github.com/MLH-Fellowship/prep-project-4.1.3/pull/31
Major League Hacking	https://github.com/MLH-Fellowship/prep-project-4.1.3/pull/17
Major League Hacking	https://github.com/MLH-Fellowship/prep-project-4.1.3/pull/9

Major League Hacking	https://github.com/MLH-Fellowship/pod-4.1.3-portfolio/pull/16
Major League Hacking	https://github.com/MLH-Fellowship/pod-4.1.3-portfolio/pull/22
Major League Hacking	https://github.com/MLH-Fellowship/pod-4.1.3-portfolio/pull/26
Major League Hacking	https://github.com/MLH-Fellowship/pod-4.1.3-portfolio/pull/34
Major League Hacking	https://github.com/MLH-Fellowship/pod-4.1.3-portfolio/pull/40
Betaoverflow	https://github.com/betaoverflow/drona
Betaoverflow	https://github.com/betaoverflow/project-orsi

What are my other commitments?

My university end semester exams are scheduled from 22nd April 2022 to 2nd May 2022. So **I have no commitments in the summer other than GSoC.** I will be available for at least 40 hours a week through online platforms and am ready to extend whenever needed. I would be working full-time for GSoC. If I do get any commitments or offers, I would accept them only after consulting the mentors.

What I plan after the GSoC period is over:

Definitely, I would like to keep contributing to OFF even after GSoC and will be available and enthusiastic to keep developing the platform. Even if I am not selected this year, I would like to help this project by resolving issues, suggesting new ideas, and participating in discussions along with making code contributions. I usually help out people with code and will love to mentor some young coders for OFF projects. I would like to keep contributing to the Open Food Facts organization and do whatever I can to help.

About me:

I am Sumana Basu a pre-final year undergrad from the National Institute of Technology, Rourkela, India pursuing a Bachelor of Technology in Computer Science and Engineering. I love building web applications that stay on the internet and can be accessed by everyone. I am a community-driven person and highly passionate about open source development. In my sophomore year, I founded Webwiz, a technical society in the university. I am a hackathon addict and currently focus on writing production-level code and have a knack for Design and DevOps. In the last 6 months, I have won 10+ international hackathons and even got funding from IBM for one of my prototypes to develop further. I have experience working with several organizations starting from leading a campus club to one of the largest companies in the world. I am an MLH Fellow which has a selection rate of less than 4% out of hundreds of thousands of applications. The fellowship is powered by Meta, AWS, GitHub, Intuit, Adobe, and other industry players. Being a full-time Technical Program Associate at MLH I've helped build one of the largest developer communities in the world. Our community is where the next generation of technologists and founders learn the skills they need to bring their ideas to life and build their professional networks. I was one of the first few students who secured an on-campus internship at SAP Labs.

My experience:

- **Synthesis** Software Developer
  Implemented a data abstraction layer between the MySQL database and the underlying microservices connected to it. It was built with GraphQL, Go, Gqlgen, and Gin to improve the security, error prevention, and uptime of the service mesh. I was part of Synthesis in MLH Fellowship with Meta, Github, Amazon, and Adobe.

- **Major League Hacking** Fellowship
  An intensive program collaborated with 15+ students from 4 different countries. Built out a portfolio of projects, and experimented with new technologies (React, Jekyll, Rust, Python) during a short, impactful hackathon sprint. The fellowship was backed by Meta, GitHub, Adobe, AWS, Intuit, and more. Lead the frontend team with delegating tasks, conducting daily stand-ups in order to build a highly performant & responsive application

- **Prayatan - Ek Awaaz** Tech Lead
  Developing a full-stack web application for an NGO based in Uttarakhand as their portfolio to significantly increase their reach and traffic. Along with that created a website for covid-related donations.

- **Udichi Foundation Pvt Ltd** Freelance
  Developed and designed dynamic and interactive web applications that ensured high traffic, page views, and user experience, resulting in a 40% increase in revenue.

- **Shashikant Rural Development Foundation** Freelance
  Developed a full-stack web application with the features of payment gateway integration for donations and log in. Also optimized the database queries.

- **Operational Research Society of India** Software Developer
  Designed and revamped user-friendly React web application. Fixed bugs from an existing website and implemented enhancements that significantly improved its functionality as an organization

- **Webwiz** Co-Founder
  In the summer of 2020, I founded Webwiz a campus club of NIT Rourkela. I lead a multicultural group of 100 developers, designers, and content writers. Researched problems and developed solutions for the students and faculty of my institute. Held mock technical and coding interviews to help a fellow member get internships. Currently, the core team has internship offers from Google, Amazon, Barclays, Flipkart, SAP Labs, Qualcomm, and more.

- **Hackodisha** Lead Organizer
  Founded and organized the largest hackathon in Eastern India. Hackodisha powered by Postman had over 1600 hackers, 40+ sponsors, and more than 1,00,000 social media reach making it one of the largest hackathons of the subcontinents. Being the Lead organizer I managed and kept all the working teams in sync. Meanwhile, took the web application from prototype to production.