

# Crowd Sourcing Website Project

## Summary of the project

### A. RepairMe - Design and purpose

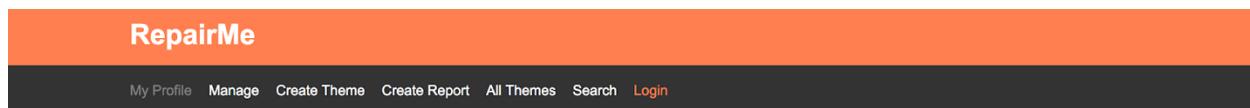
The idea behind the project was the development of a platform for the reporting of damaged public property (e.g. broken pavement parts, rusty fences, etc.). Using this web platform, it would be easier for local governments to quickly identify and resolve issues related to damaged public properties. As a result this crowdsourced platform would benefit the public using reports created by users, that hoped that the identified damages would be “repaired”, hence the name of the platform.

website: <https://apad-pk8943-python.appspot.com/>  
github link: [REDACTED]

### B. Structure

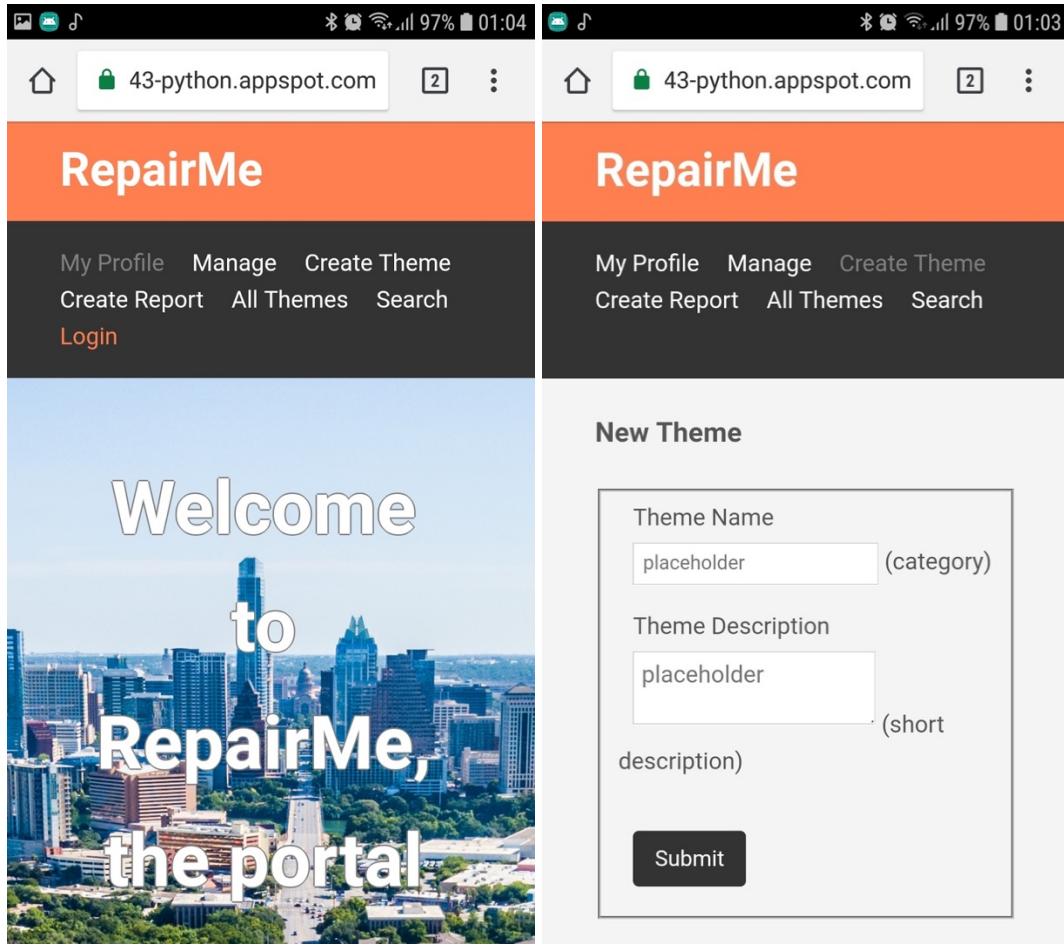
#### General structure and design

The user interaction with the website is based on the main navigation bar that includes all the different features and pages of the site that are connected together with each other and allow for easy navigation and overview of the website. Visitors of the website have access to the complete set of features of the website, even without logging in, with exception of the personal profile page and the page that manages the reports created by the user (more on that below).



*Picture 1: Navigation Bar*

The navigation bar indicates the page that the user is located in greying out the name of the page, while the login button is in coral to separate it from the rest. In addition, the style of the website is simplified and dynamic in order to support view from smaller screens.



Picture 2: Mobile friendly view

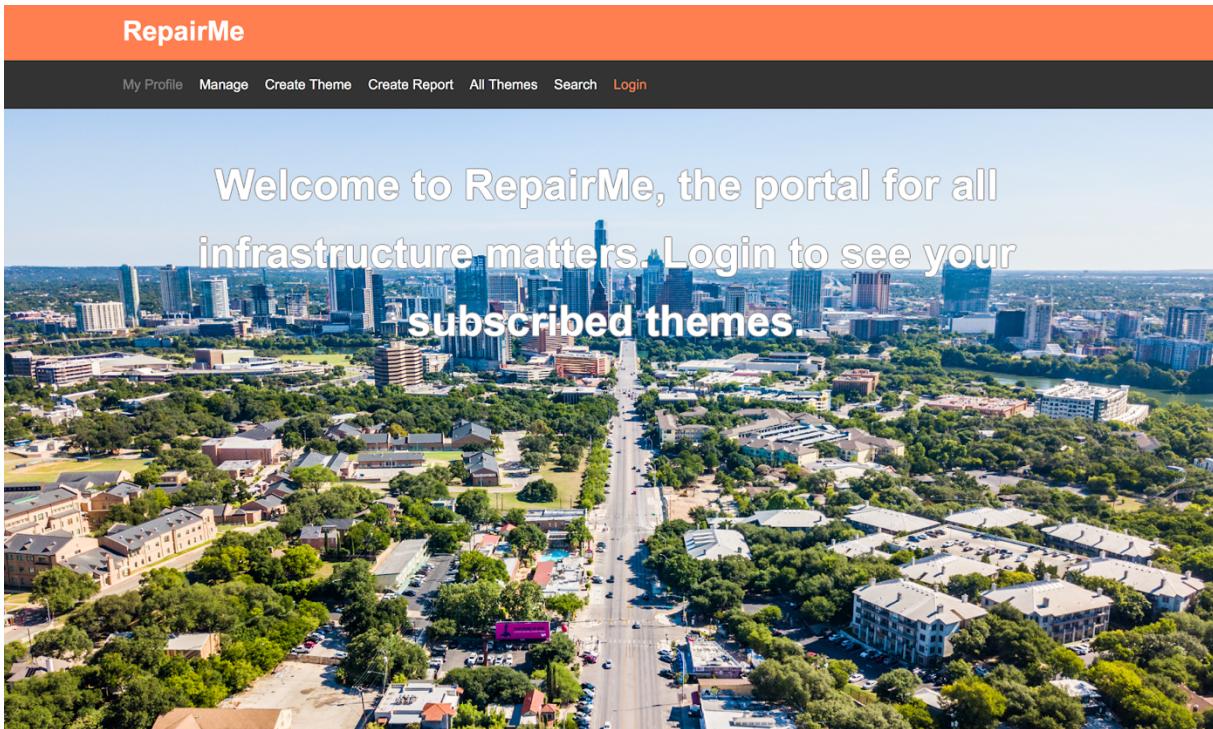
### Pages and Features

The RepairMe platform structure consists of the below pages, whose function and operation is analysed in detail below:

- i) My Profile
- ii) Manage
- iii) Create Theme
- iv) Create Report
- v) All Themes
- vi) Search

- i) My Profile

This page is a entry page for our users, welcoming them to the website, prompting users to login and see their reports and subscribed themes.



Picture 3: Welcome page

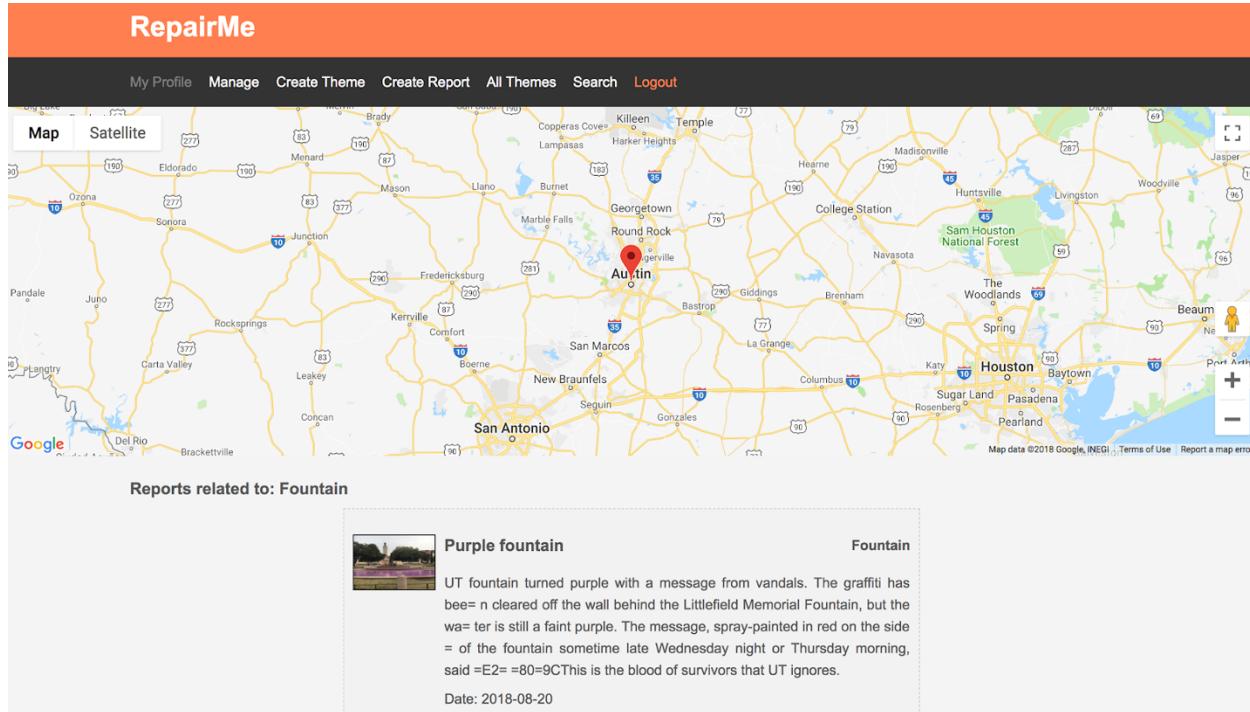
Upon successful submission of login information, the users are presented with the themes that they are subscribed to and clicking those they can access all the reports that have been submitted and are connected to each theme, while also they have access to their personal manage page, where they can see a summary of the reports they have created.

A screenshot of the RepairMe user dashboard. The top navigation bar is orange with the "RepairMe" logo. Below it is a dark grey bar with links: "My Profile", "Manage", "Create Theme", "Create Report", "All Themes", "Search", and "Logout". The main content area has a header "Subscribed Themes" and four cards: "Bench" (with "Read more" link), "Fence" (with "Read more" link), "Fountain" (with "Read more" link), and "Road" (with "Read more" link). The background of the dashboard is the same aerial cityscape as the welcome page.

Picture 4: User dashboard and subscribed themes

Upon logging in, the navigation bar is modified and the “login” button, changes to “logout”, while upon pressing it will allow for logging out of the user. While the user is logged in, all data entered are saved in the database under his specific user-email and can be retrieved for the various features and functions of the website.

Having access to their subscribed themes, users may select a theme of their choosing and review all the reports available and created by all users.



*Picture 5: List of reports related to theme “fountain”*

## ii) Manage

This page presents the reports created by the user, including all the information that was provided (i.e. title of the report, description, theme, date and search tags). Using the username / email provided by the user upon entry in the website, the page reaches out to the filters the database and returns the user submitted reports and themes connected to these reports. While the location is not presented in the reports, it is also recorded during the creation of the report and is saved in the database.

## RepairMe

[My Profile](#) [Manage](#) [Create Theme](#) [Create Report](#) [All Themes](#) [Search](#) [Logout](#)

### Your Reports

	<b>Thomas_laptop</b> Thomas_laptop Date: 2018-08-10 Tags: Thomas_laptop	<b>Thomas_laptop</b>
	<b>Purple fountain</b> Fountain UT fountain turned purple with a message from vandals. The graffiti has been cleared off the wall behind the Littlefield Memorial Fountain, but the water is still a faint purple. The message, spray-painted in red on the side of the fountain sometime late Wednesday night or Thursday morning, said =E2= =80=9CThis is the blood of survivors that UT ignores. Date: 2018-08-20 Tags: water	
	<b>Pavement at Guadalupe and 21st</b> Pavement The pavement in the corner of Guadalupe and 21st is broken and may cause an injury to all the students and their scouter that go to UT and pass that very spot every morning. Date: 2018-08-20 Tags: hole	

*Picture 6: List of reports submitted by the user*

As mentioned above, while all pages are accessible even without logging in, designed for casual use, when a visitor tries to access the manage page without having logged in, the website identifies that there is no account connected to this visitor and loads a message, instructing the visitor to login in order to access this page.

## RepairMe

[My Profile](#) [Manage](#) [Create Theme](#) [Create Report](#) [All Themes](#) [Search](#) [Login](#)

You need to be logged in to access this page



*Picture 7: Attention message for visitors that haven't logged in*

v) Create Theme

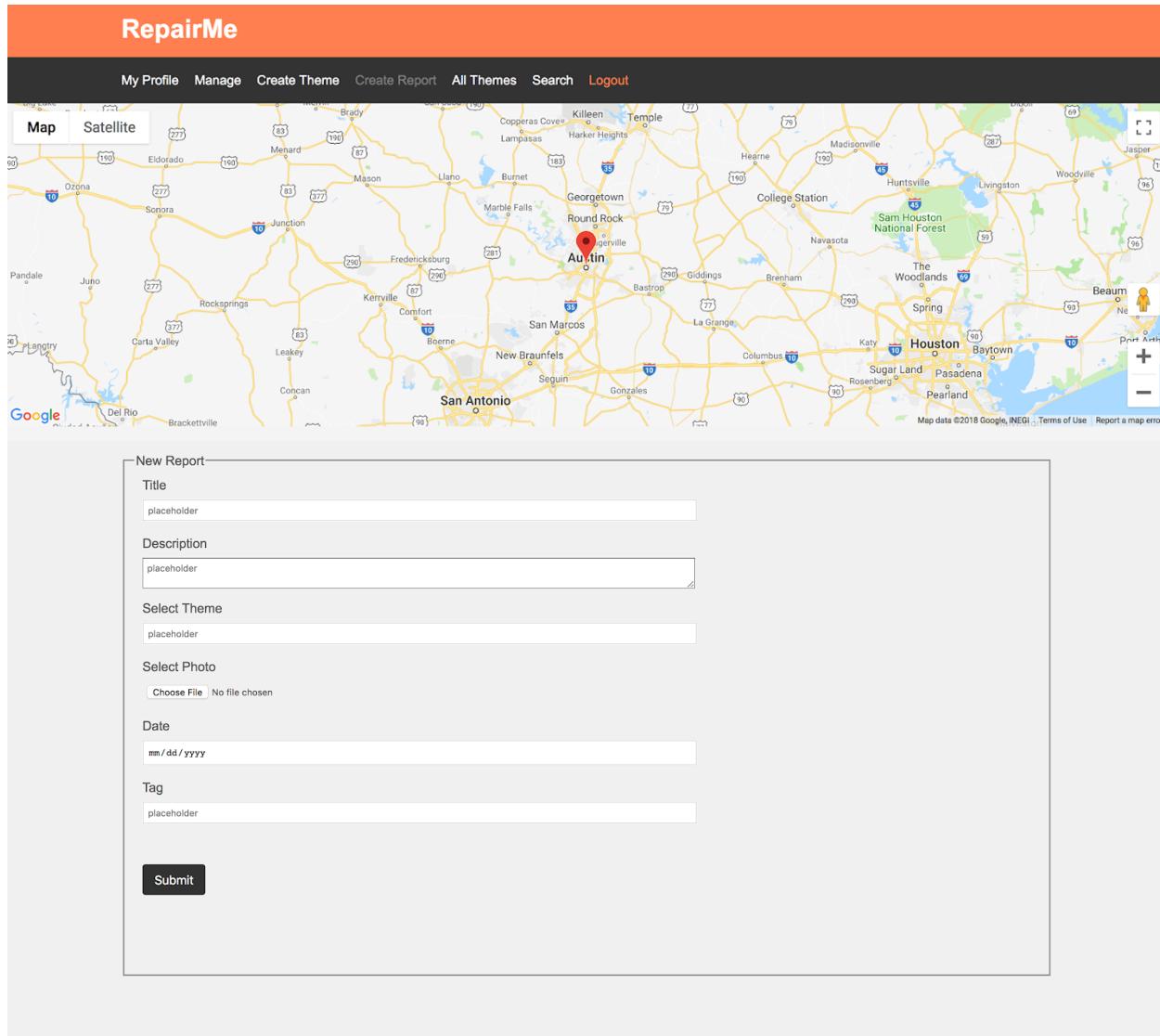
This page is devoted to the creation of new themes by the user. In order to create a new theme, the user enters the following data: theme name and theme description. These data are saved in a separate library that handles themes and are used throughout the website. Upon creation of a new theme, the user is redirected in the same page, while the data are saved in the database.

The screenshot shows a web application interface for creating a new theme. At the top, there is a header bar with the title "RepairMe". Below the header, a navigation menu includes links for "My Profile", "Manage", "Create Theme", "Create Report", "All Themes", "Search", and "Logout". The main content area is titled "New Theme". It contains two input fields: one for "Theme Name" and another for "Theme Description". Both fields have placeholder text ("placeholder") and include small text indicating their purpose: "(category)" for the name and "(short description)" for the description. A "Submit" button is located at the bottom left of the form area.

*Picture 8: Theme creation page*

vi) Create Report

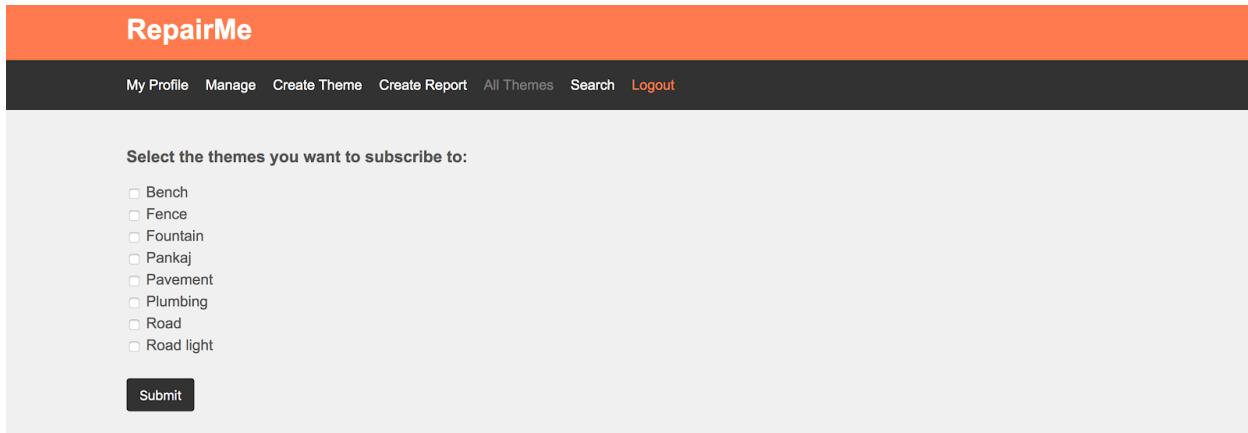
This page is devoted to the creation of new reports by the user. In order to create a new report, the user enters the following data: theme, report name, description, date and related image. While the database is created to allow the correct allocation of all the above information and ensure their use and retrieval, a report may be created by the user having as well empty fields. The report created is connected to the user that is logged in in the website, using the email of the user. In addition to the above, the location (i.e. longitude and latitude) of the user is automatically identified and incorporated in the report using google maps API, while a map is displayed centered in Austin and including a relevant marker at this specific location. In addition the theme selection of the report, has incorporated suggestions of all the existing themes, using the websites database.



Picture 9: Report creation page

vii) All Themes

This page, is dedicated to the created themes and allows for subscription. The user selects the themes of his choice and submits his request to the website. After that, the website saves this type of theme in a dedicated user specific database for subscribed themes and adds the relevant shortcut button in the personalised profile screen.

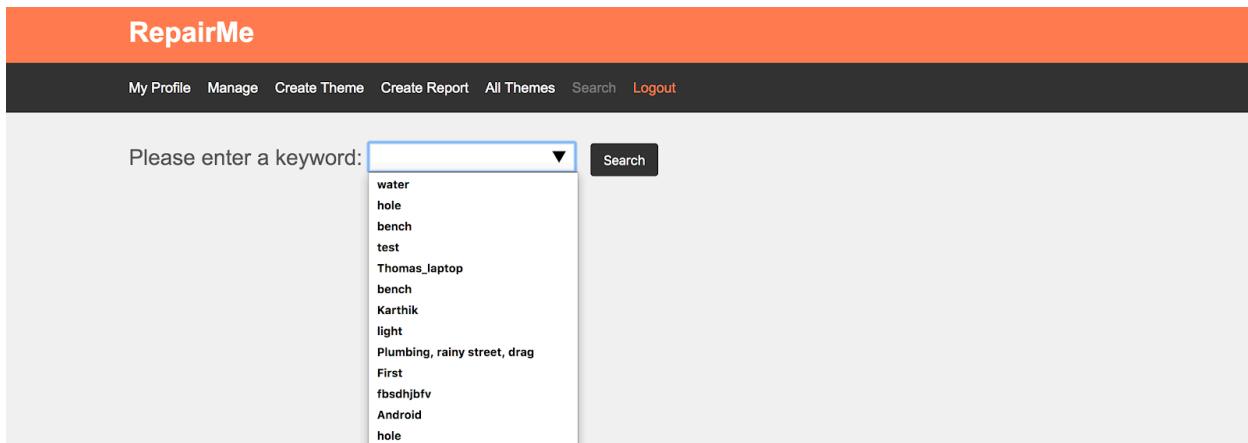


*Picture 10: Theme overview and subscription page*

Moreover, this page works as a page summarizing the data of the website and presenting the reported themes. In this case themes are items of interest (i.e. benches, fences, etc.) and are separated based on the data submitted by users (refer to the “create new theme page”). The user has direct access to any category of themes and by selecting it, he is redirected to a new webpage presenting data for all the reports related to the theme he/she selected.

#### viii) Search page

This page is used to search through the platform’s database for a specific report, based on the set tag of the report. In order to do so, based on a provided search word or phrase, it reaches the database and retrieves all the reports with matching “tags” so that the user can access them directly. When initiating the search, users are treated with search suggestions, based on the recorded tags in the website’s database.



*Picture 11: Search suggestions prior searching*

Upon selecting a tag and searching pressing the dedicated button, the website filters it’s available reports and fetches the ones that include the same tag as searched, presenting all report information (i.e. title, description, theme, date and tags). At the same time, the tag used for searching appears in the top of the page.

## RepairMe

[My Profile](#) [Manage](#) [Create Theme](#) [Create Report](#) [All Themes](#) [Search](#) [Logout](#)

Please enter a keyword:

Search results for : water

	<b>Purple fountain</b>	Fountain
UT fountain turned purple with a message from vandals. The graffiti has been cleared off the wall behind the Littlefield Memorial Fountain, but the water is still a faint purple. The message, spray-painted in red on the side of the fountain sometime late Wednesday night or Thursday morning, said =E2= =80=9CThis is the blood of survivors that UT ignores.		
	<b>Rusty and broken fountain</b>	Fountain
You call this a fountain? Not even birds go near that disgusting thing.. Fix it you useless bureaucrats!		

*Picture 12: Search results for the tag “water”*

In case a user chooses to search a tag not included in the suggested list, while he has the option to do so, no results will return and an image will pop-up indicating the same.

## RepairMe

[My Profile](#) [Manage](#) [Create Theme](#) [Create Report](#) [All Themes](#) [Search](#) [Logout](#)

Please enter a keyword:

Search results for : wrong-tag



*Picture 13: Search results for the tag “wrong-tag”*

## C. Tools

The below table presents the programming tools used in this project for reference purposes.

Front end programming	HTML, CSS
Template language	Jinja2
Back end programming language	Python
Database service	Google Cloud Storage
Web framework	webapp2
Android Application	Kotlin

## Lessons learnt

### A. Takeaways from the project

#### 1/. Back end web development with python

The project was a very good opportunity to implement Python programming skills, combining them with different technologies (jinja2, Google Cloud Storage) and working hard on the effective interconnection of all of them. Debugging was a much complex procedure, inspecting multiple layers of operation and different connection points, following the flow of data entered from the user, being successfully and correctly transferred in the database and later also being retrieved again in the html and presented in the UI, resulting in a seamless operation of the website and flow of data.

#### 2/. Use of Google Cloud Storage and jinja2

This project helped into grasping a better understanding of the operation principles and functionality of the Google Cloud Storage as well as Jinja2. In order to develop the platform, gathering online information and material I identified tools and methods that could be used in the project, moving steadily one step at a time.

#### 3/. Implementation of the designed framework

While having a great start on the project, defining the end goal and the framework of the website, I faced many difficulties in the implementation stage, recognizing design problems that were even based on restrictions of the technologies that were used. As a result, I followed a twisted path, coming up with alternative solutions and redesigning the python program; and while the end result matched the initial design, the technology behind it was different compared to the initial prototype.

#### 4/. Debugging and testing

Developing a complete project including front and back end development, cleaning and improving the code throughout the process, the importance of standardized testing and debugging became clear. While mistakes are expected in such a process, no mistake needs to be made twice.

## **B. Things that just worked**

### **1/. Initial idea**

The first session was successful and very productive. Having the main idea related to the project, themes and reports at a very early stage, was a great starting point for the project and the overall process.

### **2/. Designing the website framework**

Using basic design tools, sketching our mocks on paper, the overall framework of the platform was quickly defined while initial decisions were made in regards to the end goal and actual interactions of the database and html.

### **3/. Python**

Feeling comfortable with Python, I had no difficulties in coding having good understanding of the code and being able to reference online resources. While there were difficulties related to the incorporation of jinja2, being familiar with Python programming was key throughout the research and debugging processes.

### **5/. HTML**

It was important to create a basic structure of the website front end side in order to test and implement the interaction of the platform and transferring of data. This crucial part of the project was completed swiftly and effectively, and I was able to work from there continuously improving the UI and working in parallel in the interconnection with the database.

## **C. Pain Areas**

Even though the project as a whole was a great learning experience that forced me to strengthen my technical skills and implement them in an actual working website, I struggled daily and faced numerous bugs and problems ensuring the interaction of different tools and every feature that was added to the website. Below is a small list of items that successfully overcome during the project:

- Finding how to instantiate the handlers in web application
- Trying to traverse from one page to another in webpage
- Creating same handler for all report themes
- Eliminating the possibility of multiple theme creation
- Uploading images using blobstore
- Fetching images from blobstore

Despite all the hurdles and day to day continuous tackling of issues I was able to figure out solutions and managed the completion of all the basic functionalities of a sustainable and modern crowd sourcing website.