

# Wiki Formatting Design Document

## Authors

Mario Macias - [memacias@google.com](mailto:memacias@google.com), Dustyn Loyda- [dloyda@google.com](mailto:dloyda@google.com), Alexandro Fernandez - [alexandrof@google.com](mailto:alexandrof@google.com)

## Background

How can we allow a user to give input or edit privileges onto a wiki page without having to worry over the sanitization of their input? Can this be expanded to completely remove their ability to use tags to avoid formatting discrepancies on the wiki page?

## Proposal

With an HTML text-area we can allow users to edit pages in our wiki without needing to concern ourselves with sanitizing their input.

## Implementation Details

The use of Python dictionaries to store States/Provinces as keys with a Python list of each state's cities as a value. In essence the structure would be a dictionary storing a set of lists. This helps maintain small time complexities due to the extraction of data from a dictionary but could be improved later to store another dictionary instead of a list as lists can have longer load times depending on the position of the desired city.

An HTML text area with the page contents is loaded onto a form which can be edited by the user, formatting is controlled by line order and is reviewed for completeness. This means if a line is missing the edit will not be allowed until the user fixes it or if the edit is discarded. This text area also allows the input from the user to be sandboxed as its contents are read as plain text with no use for a parsing language but rather a line-controlled input.

The input from the text area is then saved onto a text file which also is logged onto the page history. The contents are then reloaded and displayed back to the user. The user interface for cities is divided into a title section, a fact description section, and the discussion board section. If the user tries to change this format through the edit function an error is generated and displayed to the user as the wiki's formatting is the main line of defense against unwanted scripts.

By allowing links to be posted as plain text we can also dis-encourage users to click on links which could lead to malicious websites. Most users would now be forced to read a portion of the link before copying and pasting onto their address bar which could potentially save them from a phishing scheme of sorts.

The home page contains all cities divided by states. This means that every line should be represented by a state and listed on that line are all of the cities for said state. The cities are then formatted into "cards" which contain the state, city picture and a link to the city

page. This is just to keep the UI clean and easy to navigate.

The user can add a new city page from the home page which will be added to the state dictionary and rendered according to their input. A text file is then generated for the page which is what is used as an outline for rendering the city page.

The goal of this formatting implementation is to keep from having to sanitize every line of input given from the user. This allows our wiki to maintain uniformity whilst allowing users to post about their interesting cities or edit someone else's page about a city.

## Security Considerations

The use of the plain text input allows us to ignore tags that could open malicious scripts or malicious links. Something else to keep in mind is that users can generate their own page, but through the page rendering format the user must create the page to match the formatting of the other pages, hence whether they edit a page or create a new one the same standard for rendering a page is upheld and the integrity of the wiki is maintained.

## Alternatives Considered

- Alternative markup language
  - The use of markdown was considered but then we would need to sanitize user input which could cause more time delays when trying to render an edited page, hence this idea was scrapped, and we continued to use HTML.
- Alternative data structure
  - The use of 2-D lists was discussed but given the amount of time a larger wiki could take, we opted to use the dictionary/list combination as this could save us much of the state search. In the future a nested dictionary is almost surely going to be the used data structure as that would help with city search.

## Resources

- <https://www.w3schools.com/TAGS/default.ASP>
- [https://en.wikipedia.org/wiki/Richard\\_Ramirez](https://en.wikipedia.org/wiki/Richard_Ramirez) (example of dark history of a city)
- <https://developer.mozilla.org>
- <https://www.geeksforgeeks.org/hash-map-in-python/>
- <https://www.geeksforgeeks.org/python-hash-method>