Generalising Traditional Wiki Systems to New Media Page Formats

A Project Report

Presented to Dr. Chris Pollett

Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the

Class of Spring-2021: CS 297

By

Aniruddha Dinesh Mallya

May, 2021

ABSTRACT

Wiki systems are a kind of website application that allows collaborative editing of its content and structure by its users. Such systems make it quick and easy to write information on web pages, facilitate communication and discussion. The goal of this project is introducing new different modern media formats like URL shorteners and pastebin to traditional wiki systems like Yioop to improve collaboration.

Yioop is an open source, PHP search engine that can be configured to allow users to create discussion groups, blogs, wikis etc. In this semester we divided the initial project phase into four deliverables. Initially, we created a foundation for developing a chat system in Yioop for its users. Next, we designed the logic for creating a URL shortener to be able redirect to different web-pages quickly. Later on we created a pastebin service for users and laid the groundwork for different views in Yioop.

Index Terms: Wiki systems, Yioop, Web-pages, URL shortener, Pastebin, Direct-Messaging, Chat protocols, Wiki Pages

Table of Contents

INTRODUCTION 1

CREATE A PRE-DEFINED GROUP FOR A DIRECT MESSAGING SYSTEM 3

CREATE A URL SHORTENER 5

CREATE A PASTEBIN LIKE MEDIA PAGE FORMAT 7

ADD VIEW ENHANCEMENTS FOR DIFFERENT MEDIA TYPES 9

CONCLUSION 11

REFERENCES 12

INTRODUCTION

Wiki is a web-based software that allows all viewers of a page to change the content by editing the page online in a browser. This makes the wiki a simple and easy-to-use platform for cooperative work on texts and hypertexts. To work with wiki it was first important to understand the general workflow of wiki pages on Yioop. The wiki script is a PHP script that reads the raw page data from an SQLite database. This raw data is checked line for line, and the specific format commands contained in it are replaced by the corresponding HTML codes. Subsequently, the page thus created is integrated in the layout template. Every wiki page has its own, distinct name indicating the subject of the page. In addition, there is a navigation menu and a few page-specific links of the displayed page. The most important of these links is the “Edit” link. This sends another query to the server after which the same page is loaded again, only this time the contents are not converted to HTML format, but rather displayed in “raw form” in a large text field in an HTML form. The user can edit the text in this form and send a new version, which immediately replaces the old version in the database. When the page is accessed again, the new version is displayed[5].

We know wiki systems in discussion boards need some kind of a group system to share and exchange discussion items and different systems are different in how they do it. This project explores how documents can be updated and shared, for e.g. **Wiki.js**, it allows content migration using WikiText editor, tabular editor, and a REST/GraphQL optimized editor. You can track content modifications, compare versions to see the difference, revert to the past state, or even recover deleted pages or export a page version, or **MediaWiki**, it is an open and free wiki software that includes wikis like Wikipedia, Wikimedia Commons, Wiktionary etc. It helps collects knowledge, organizes it, and then make it available to all. So this report primarily works towards extending Yioop as a wiki system.

The remaining sections have been organized as follows:
The first section describes how we create a sub-system to handle direct-messaging systems, next section, we create a shortened URL for wiki pages, thirds section we add a pastebin like wiki page, the fourth section introduces view enhancements for different media like audio, video, images etc. for Yioop and finally we conclude with what was achieved and the scope of the future work.

Pre-requisites: **Yioop** (can be found at: https://www.seekquarry.com/p/Downloads), used for all deliverables. This application is developed in **PHP**.

CREATE A PRE-DEFINED GROUP FOR A DIRECT MESSAGING SYSTEM

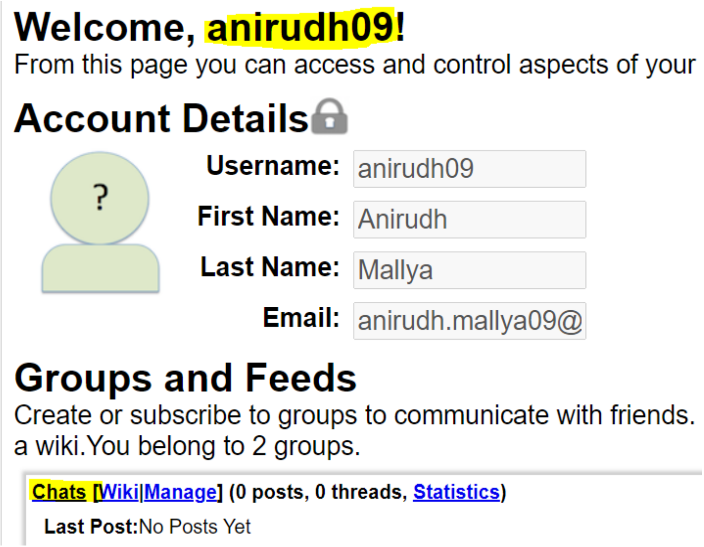
Direct messaging (DM) technology is a type of online chat that offers real-time text transmission over the Internet. Some systems permit messages to be sent to users not then 'logged on' (*offline messages*), thus removing some differences between DM and email[6]. So, the aim for Yioop chat system is similar here to make it possible to save a text conversation for later reference.

The motive behind this deliverable is to setup a pre-defined group that will enable direct messaging between users of Yioop. Yioop follows the model-view-controller (MVC) software design pattern, so the models that create/edit users and groups, were used to add aforementioned pre-defined groups called "Chats" through the "AccountaccessComponent " controller.

The groups table in the SQLite “work_directory” manages the details of all user groups and the different permissions set for their groups. As a single table is used to manage all user group information, the groups’ names are required to be unique. In order to do so a unique name id was created for users with their username and group id in the table for example, “Chats$anirudh09$1”.

Some of the challenges in this deliverable that required multiple iterations was creating a “Chats” group for existing users. We first tried creating the group in one click for all existing users. This quickly became complex to handle as there were many dependencies between the different tables in Yioop’s database. The second and easier method was to create groups when a user logs into Yioop for the first time and to check if the active status of the user is 1, if so all we create the “Chats” group for that user using the same logic when a new user registers in Yioop and change his active status to 5 in the system. This logic is planned for later use to help identify different Yioop versions when new features are to be added. Another challenge was managing the title view of the "Chats" group as it displayed the full username and id of a user on different webpages that use a user's group details for e.g. all the groups a user is part of or the menu bar of a group. To tackle this first we determined that all of the display functionality is handled by "elements" in the views section and created a method used across different elements to handle this issue.

Sample Output:



Welcome, anirudh09!

From this page you can access and control aspects of your

Account Details

Username: anirudh09

First Name: Anirudh

Last Name: Mallya

Email: anirudh.mallya09@

Groups and Feeds

Create or subscribe to groups to communicate with friends.
a wiki.You belong to 2 groups.

Chats [Wiki|Manage] (0 posts, 0 threads, Statistics)

Last Post:No Posts Yet

Here, "Chats" group for all users of Yioop is created to allow users to make threads with other users and act as form direct messaging. This system would also allow sending offline messages mentioned earlier to mimic an emailing system. The big picture here is to enable a clever direct messaging system.

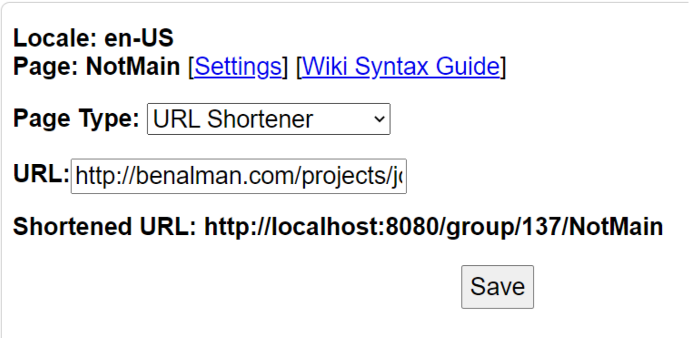
CREATE A URL SHORTENER

Many web developers pass descriptive attributes in a URL to represent data hierarchies, command structures, transaction paths or session information. This can result in URLs that are hundreds of characters long and that contain complex character patterns. Such URLs are difficult to memorize, type out or distribute. As a result, long URLs must be copied and pasted for reliability. Thus, short URLs may be more convenient for websites or hard copy publications (e.g. a printed magazine or a book), the latter often requiring that very long strings be broken into multiple lines (as is the case with some e-mail software or internet forums) or truncated. A permanent URL is not necessarily a good thing. Some services that provide this are bit.ly, goo.gl, TinyURL etc.

So if we have a long URL like “https://en.wikipedia.org/wiki/URL_shortening” TinyURL shortens the link to look like “tinyurl.com/xtnftxyx”. Similarly, we aim to create a URL shortener for Yioop. Yioop already has a functionality that allows users to be able to redirect to pages within its ecosystem This can be used to handle things like different names for the same topic or to do localization of pages. For example, if you switch the locale from English to Spanish.

The motive behind this deliverable is to give a user the ability to redirect from Yioop to webpages outside of Yioop and also be able to perform some of the functionality mentioned earlier. When users go to create a wiki page the “URL Shortener” is presented as an option in edit settings. In the URL text area, the user places the desired redirection webpage for e.g. yahoo.com, on clicking the "Save" button a shortened version of the link is created using the group id and the wiki pages title name example shown below.

Sample Input & Output:



Locale: en-US
Page: NotMain [Settings] [Wiki Syntax Guide]

Page Type: URL Shortener

URL: http://benalman.com/projects/j

Shortened URL: http://localhost:8080/group/137/NotMain

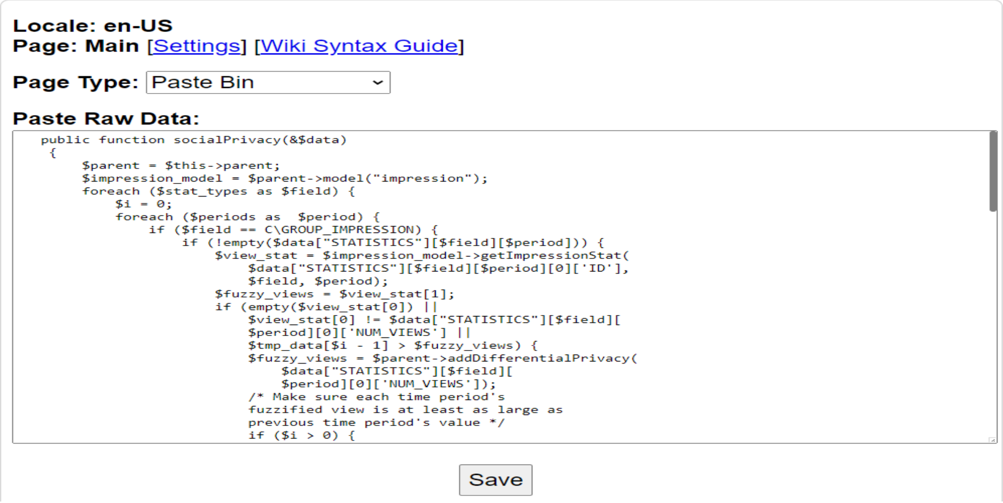
Here, we see "NotMain" is wiki page title, and the group id is 137, so we get the shortened URL as http://localhost:8080/group/137/NotMain which redirects to http://benalman.com/projects/jquery-starwipe-plugin/. There are some security implications where obsolete short URLs may remain in existence and still circulated long after they cease to point to a relevant or even extant destination. So some URL shorteners offer a time-limited service, which will expire after a specified period. Services available include an ordinary, easy-to-say word as the URL with a lifetime from 5 minutes up to 24 hours, creation of a URL which will expire on a specified date or after a specified period and password protection[3]. Yioop has a functionality that allows to implement such a timed URL life wherein it expires after a fixed amount of time if the group created is meant to expire as well. Some possible future work here can be to customize Yioop's timed group life for webpages thus making the timed URL unique to different pages in a group.

CREATE A PASTEBIN LIKE MEDIA PAGE FORMAT

A pastebin or text storage site is a type of online content hosting service where users can store plain text, e.g. to source code snippets for code review via Internet Relay Chat (IRC). The first pastebin was pastebin.com. Pastebins may allow commenting where readers can post feedback directly on the page. They were developed to facilitate IRC chatrooms devoted to computing, where users naturally needed to share large blocks of computer input or output in a line-oriented medium. So the motive behind this deliverable is to give any user i.e. both anonymous and registered the ability to read and edit code posted by users on Yioop and mimic a pastebin like service.

When users go to create a wiki page the Pastebin is presented as an option in settings. In the Paste Raw Data text area, the user places the desired code snippet for review and collaboration with other registered users or anonymous users. The $_REQUEST functionality in php is specially handled to allow anonymous and registered users to edit a posted code snippet, provided the wiki page group has the read permission set for anyone to access it. On clicking "Save", a sample code snippet is shown below.

Sample Input (View for Creating a Wiki page for Pastebin):



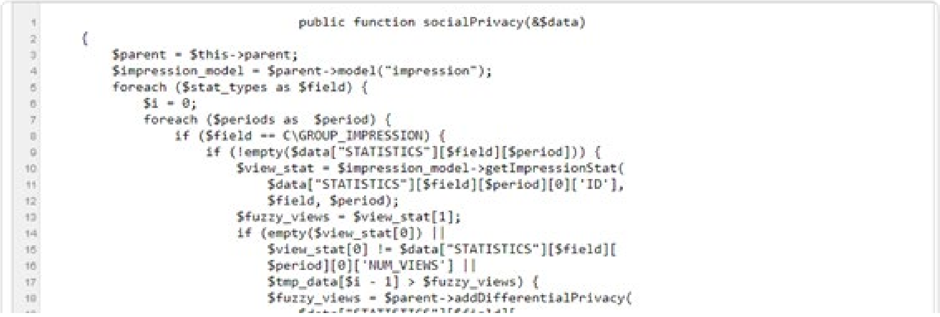
Locale: en-US
Page: Main [Settings] [Wiki Syntax Guide]

Page Type: Paste Bin

Paste Raw Data:

```
  public function socialPrivacy(&$data)
   {
        $parent = $this->parent;
        $impression_model = $parent->model("impression");
        foreach ($stat_types as $field) {
            $i = 0;
            foreach ($periods as  $period) {
                if ($field == C\GROUP_IMPRESSION) {
                    if (!empty($data["STATISTICS"][$field][$period])) {
                        $view_stat = $impression_model->getImpressionStat(
                            $data["STATISTICS"][$field][$period][0]['ID'],
                            $field, $period);
                        $fuzzy_views = $view_stat[1];
                        if (empty($view_stat[0]) ||
                            $view_stat[0] != $data["STATISTICS"][$field][
                            $period][0]['NUM_VIEWS'] ||
                            $tmp_data[$i - 1] > $fuzzy_views) {
                            $fuzzy_views = $parent->addDifferentialPrivacy(
                                $data["STATISTICS"][$field][
                                $period][0]['NUM_VIEWS']);
                            /* Make sure each time period's
                            fuzzified view is at least as large as
                            previous time period's value */
                            if ($i > 0) {
```

Sample Output:



```
                            public function socialPrivacy(&$data)
    {
        $parent = $this->parent;
        $impression_model = $parent->model("impression");
        foreach ($stat_types as $field) {
            $i = 0;
            foreach ($periods as  $period) {
                if ($field == C\GROUP_IMPRESSION) {
                    if (!empty($data["STATISTICS"][$field][$period])) {
                        $view_stat = $impression_model->getImpressionStat(
                            $data["STATISTICS"][$field][$period][0]['ID'],
                            $field, $period);
                        $fuzzy_views = $view_stat[1];
                        if (empty($view_stat[0]) ||
                            $view_stat[0] != $data["STATISTICS"][$field][
                            $period][0]['NUM_VIEWS'] ||
                            $tmp_data[$i - 1] > $fuzzy_views) {
                            $fuzzy_views = $parent->addDifferentialPrivacy(
                                $data["STATISTICS"][$i-1][
```

The output here is generated using a picture resource for the text area tag in HTML and setting it in “search.css”. While testing this deliverable, one issue that particularly made it a challenge to implement was the browser cache incorrectly saving the resource at times requiring to restart the entire browser. Replacing the resource with pure Document Object Model manipulation is a possible future work for pastebin in Yioop.

ADD VIEW ENHANCEMENTS FOR DIFFERENT MEDIA TYPES

The motive behind this deliverable is to give any user the ability to view media in a format other than list view on Yioop. In this particular case we will focus on a responsive GRID view. Many web pages are based on a grid-view, which means that the page is divided into columns. Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page. It also acts as a potential precursor to add a column like view for ease of drag and drop of different items into the user's folder.

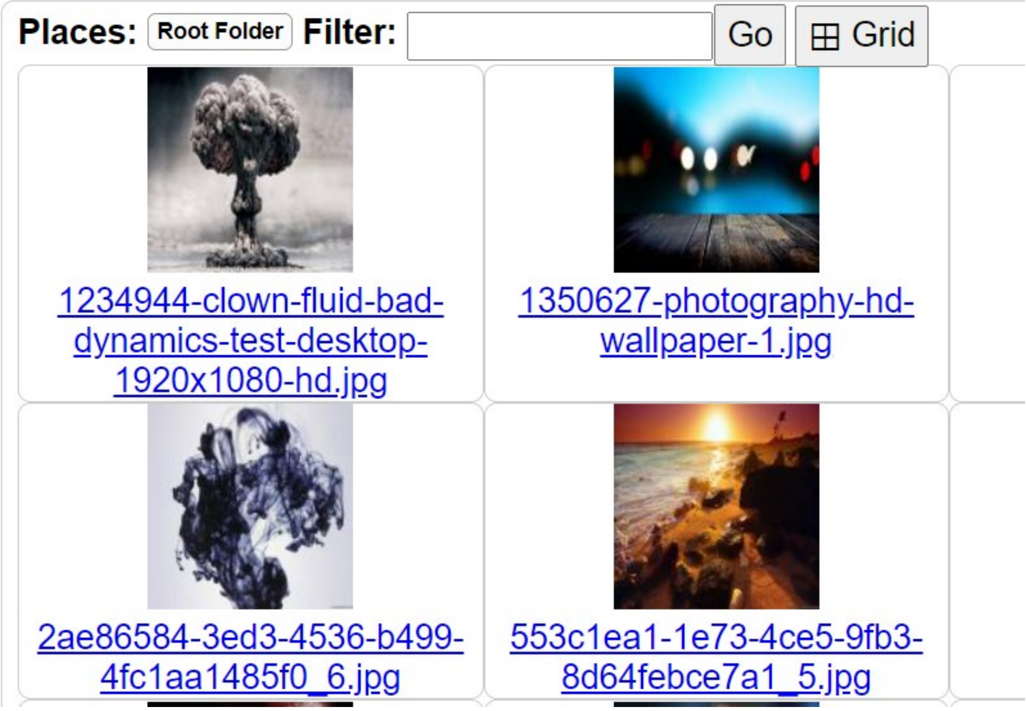
This deliverable was primarily done by manipulating the Document Object Model (DOM). Each of the items in the desired media folder was wrapped in a <div> tag and based on what the user selects the view would dynamically change.

Code Snippet (.css):

```
.grid-container {
    display: grid;
    grid-template-columns: auto auto auto;
 }
 .solid-border .grid-item {
    width:3in;
    text-align: center;
 }
```

Here, each <div> item wrapped around the individual media items had a “grid-item” class assigned to it, while all the items themselves were wrapped around a <div> tag and was assigned the “grid-container” class. This presents the output as shown below.

Sample Output:


Places:
Root Folder
Filter:
Go
⊞ Grid
1234944-clown-fluid-bad-dynamics-test-desktop-1920x1080-hd.jpg
1350627-photography-hd-wallpaper-1.jpg
2ae86584-3ed3-4536-b499-4fc1aa1485f0_6.jpg
553c1ea1-1e73-4ce5-9fb3-8d64febce7a1_5.jpg

When a user clicks on one of these items in list a dark diamond icon appears on the left of the media item to indicate that the item has been visited once. Similarly for grid view the icon is set to appear at the top of the tile. Additionally, the view here will shrink and expand as you resize the browser window.

CONCLUSION

To enable an easy introduction to the world of wikis and create a comfortable working atmosphere, wiki syntax was made as simple as possible on a web portal like Yioop. In this project, we developed new and improved existing wiki pages. We developed a “Chats” group for old and new users on Yioop as a quick communication channel to deliver short messages. Next, we developed a URL shortener to improve user engagement, prevent email with these links from ending up in spam and ultimately to save time. The third deliverable focused on creating a pastebin like service to provide quick and efficient feedback on user content in Yioop. Finally, the last deliverable provides a grid view for different media items to improve responsiveness and to more evenly spread user’s attention on web page.

In the second semester of the project, we will focus on finishing the direct-messaging system using the work done in Deliverable 1. A possible extension to view enhancements in Deliverable 4 is adding a column like view and a drag-drop feature for files when in this view. Further we try to implement a clever recommendation system to show more relevant items on a user’s feed page possibly by looking at users past and current engagements in Yioop.

REFERENCES

[1] M. Ahlholm, S. Grünthal, E. Harjunen, “What Does Wiki Reveal About the Knowledge Processing Strategies of School Pupils?”, Scandinavian Journal of Educational Research, 2017, pp. 448-464, doi: 10.1080/00313831.2016.1172495.

[2] M. Squire, A. Smith, "The Diffusion of Pastebin Tools to Enhance Communication in FLOSS Mailing Lists.", Open Source Systems: Adoption and Impact. Cham: Springer International, 2015, pp. 45-57, IFIP Advances in Information and Communication Technology.

[3] H. Zhang, X. Wu, Y. Zhang, C. Zheng, J. Liu, Q. Lin, "A URL Shortener for Mobile Web Consumption." Proceedings of SPIE 8302.1, 2012, doi: 830208-30204.

[4] S. Mader, “Wikipatterns : A Practical Guide to Improving Productivity and Collaboration in Your Organization”, John Wiley & Sons, Incorporated, 2007. Available: https://ebookcentral.proquest.com/lib/sjsu/detail.action?docID=331532.

[5] G. Dueck, A. Ebersbach, M. Glaser, R. Heigl, "Wiki: Web Collaboration.”, Berlin; New York: Springer, 2006.

[6] R.B. Jennings, E.M. Nahum, D.P. Olshefski, D. Saha, S. Zon-Yin, C. Waters, "A Study of Internet Instant Messaging and Chat Protocols.", IEEE Network 20.4, 2006, pp. 16-21, doi: 10.1109/MNET.2006.1668399.