DESIGN AND PROGRAM A HEALTH NEWS RSS AGGREGATOR

Advisor: Asst. Prof. Salih Bayar

Atilla Gündoğan

MARMARA UNIVERSITY

MARMARA ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

1. Introduction

Today, readers find themself in a huge pile of information. There is just simply too much publishers, news and researches to keep up with. RSS feeds helped those readers until the 90s but then the technology started to lose its place and now the average person doesn't even know what an RSS feed is. Even though many websites, publishers, blogs and even forum sites offer this service there is not an intuitive way to use this information. In this project we gathered data from RSS XML files from publishers and inserted into a SQL Database so we can use it in other applications and projects. After inserting the data, we can pull them directly or query them to select one or more specific data with requested features.

2. Research Objective

Although having too much information doesn't sound bad at all it can have a negative impact. Having too much unorganized information can make the desired info buried under unrelated topics and researches. But even too much organized information can cause problems. One of those problems is information overload. Information overload is probably something you have felt before after hours of researching the web and you realize that there is too much data floating around and you can no longer make decisions about the topic. To prevent this is in this project we only gathered data from Turkish Heath News Publishers. We defined 6 categories including Title, Link, Media, Meta Description, Content and Publication Date. After inserting the data into a database, we can search and filter the data based on containing keywords in the content, name of the publisher, publication date, title, and content itself. Also, only the data in the form of RSS is not helpful. We need an RSS Reader or an RSS Aggregator. The RSS data is in XML Form, and it is hard to read and not intuitive enough for the public. With the help of PHP, we will also display the database content in a web page, that will make it easier to read the data without full permission access to the database itself.

3. Related Works

There is a few closed-source RSS Aggregators which you can't run it self-hosted. And then there is NewsBlur, the biggest open-source RSS Reader with a plenty of features. But you must run it a dedicated server alongside Docker. Because you will need a dedicated server it may cost a lot. Also, the code base is very large and thus it is hard to understand without a long read into the documentation.

4. Design

4.1 Realistic constraints and conditions

Since the RSS data contains copyrighted work we can't edit its content. We can only use it for ourselves or share it while giving credits to the authors and the publisher.

4.2 Cost of the design

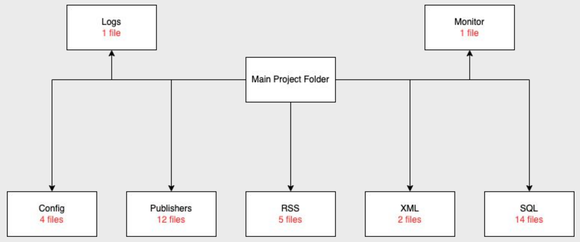
Cost of implementing the codes alongside a website installation is free with a cPanel installation.

4.3 Development Standards

While developing an app or a program we must follow the Programming Language's Syntax, requirements, and standards. Since we only used PHP in this project, we must use it accordingly in an intended way and syntax.

4.4 Details of the design

The project scripts work under 7 main folders. They communicate with require tags. There are 36 PHP files in total and they all help the project work as intended.



5. Methods

I used GitHub to save and track my progress while I developed new features.

For development purposes I used XAMPP to run Apache, MySQL and phpMyAdmin locally.

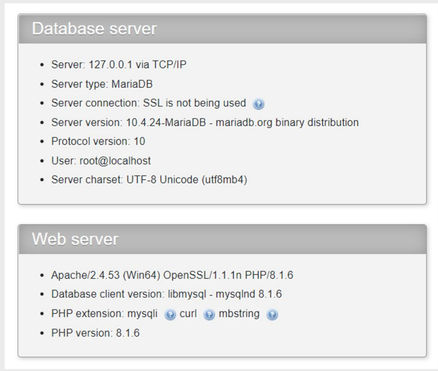
Apache is the most used free and open source web server software that allows users to deploy their websites on the internet.

I used Visual Studio Code through the entire development process. It is a free and open source powerful IDE from Microsoft.

PHP stands for Hypertext Preprocessor. PHP can generate dynamic page content. It can create, open, read, write, and delete files on the server.

MySQL is a relational database which we used to push or insert our data into. We can create databases, create tables into those databases and execute read write operations.

We can display our database with the help of phpMyAdmin. We can also edit and interact with our data and tables in the admin panel.



6. Results and Discussions

We have reached the desired result but we can make it work a lot better if we use a NOSQL database like MongoDB or Firebase since it will get rid of some limitations and give us some new features to look deeper into our data. We can also make use of Google's Machine Learning platform and do more calculations to increase our productivity.

We also did not implement an SSL Certificate and we may need to add it so when we expose our server to the public we can protect our services.

7. Conclusion

Utilizing RSS sources and XML files we have successfully created a Heath News RSS Aggregator. With RSS we can choose to view the news and articles we want, the publications that interest us and are relevant to our work. With RSS we remove unwanted information. RSS services uses XML as a language and XML doesn't have a standard. RSS Readers doesn't read and display all the data. Using PHP we can get access to every data and use it across our projects. At the end we made a lightweight program that integrated RSS data to SQL Database. In the later projects we can use this data extensively and create a Machine Learning Application. This project will also serve as a Search Engine as much as it is now serving as an RSS Aggregator. By using user data end queries we can create a simple Google alternative.

Acknowledgements

I would like to thank our teacher Asst. Prof. Salih Bayar for his valuable support and advice on the project, both for guidance and moral support.