**UNIVERSITY OF THE PHILIPPINES
OPEN UNIVERSITY**

Master of Information Systems

WINELFRED G. PASAMBA

ELLEN G. WHITE WRITINGS SEARCH ENGINE

Project Adviser:
Conception L. Khan, PhD
Faculty of Information and Communication Studies

U.P. OPEN UNIVERSITY
May 05, 2020

Disclaimer

      This project is submitted to the Faculty of Information and Communication Studies in partial fulfillment of the requirements for the degree Master of Information Systems at the University of the Philippines – Open University, Los Baños, Laguna. It is the product of my own work except where indicated in the text. The project report or any portion thereof including the source code or any section may be freely copied and distributed provided that the source is acknowledge.

Acceptance Sheet

This project proposal entitled "Ellen G. White Writings Search Engine" submitted to the Faculty of Information and Communication Studies, University of the Philippines – Open University, Los Baños, Laguna in partial fulfillment of the requirements of the degree Master of Information Systems is hereby accepted.

_____

Signature of the Adviser

_____

Date

_____

Signature of the Dean

_____

Date

Acknowledgements

I would like to thank God for using Ellen G. White to bless his children with so much truth for the last days. And for leading and guiding me in this project. This is a miracle to me. I have been wanting to make this project for a long time but many things had to come to place.

I also thank Pastor Edgar Bryan Tolentino, associate director of the EGW Estate at Adventist International Institute of Advanced Studies, for the support in this project.

Thank you for UPOU faculty and staff for allowing me to do something I'm very interested in.

Abstract

Ellen G. White writings, which are about 5,000 periodical articles and 40 books in her lifetime, some of which are translated to 140 languages, are one of the most translated literature and are vital importance to Seventh-day Adventists. There was currently no system to take questions and present quotes, articles, or chapters/sections as answers. A system was be built to cater to this opportunity.

Table of Contents

I. THE PROBLEM DOMAIN

A. Statement of the Problem

The project took the opportunity to make an interface to Ellen G. White writings that attempts to answer questions with the most relevant quotes, articles, and chapters.

B. Background and Objectives of the Project

Ellen G. White writings, which are about 5,000 periodical articles and 40 books in her lifetime, some of which are translated to 140 languages, are one of the most translated literature and are vital importance to Seventh-day Adventists. There was currently no known system to take questions and present quotes, articles, or chapters/sections as answers. A system was built to cater to this opportunity.

The previously existing interfaces or apps for accessing EGW Writings were:

1. http://egwwritings.org/
2. Android EGW Writings App (1st version)
3. Android EGW Writings App (2nd version)
4. iOS EGW Writings App
5. EGW Family Edition / Research Edition CDROM/DVD
6. Unofficial EGW Writings app for Symbian phones

All of the above had a "keyword" search functionality. Some of them had the EGW Topical Index which is very helpful in presenting an outline to any topic.

There was a need to develop an interface to answer questions. Typing a question into the search function of the above systems didn't yield the best answers.

The writer is an avid reader of EGW Writings and The EGW Estate has expressed support for this attempt.

C. Significance and Scope of the Project

Users had difficulty recalling the exact words to find in EGW writings. So they were not very productive with the keyword search. More so, many Adventists who had real questions had them because they were not familiar with EGW Writings.

The project took the opportunity to fill in the gap for a system that tries to match queries in question format and give relevant EGW Writings.

D. Documentation of Existence and Seriousness of the Problem

1. http://egwwritings.org gave an incorrect answer.

The answer to "is sabbath cooking prohibited?" went to Isaiah. Probably because the first word, "is" is a shortcut for "Isaiah".
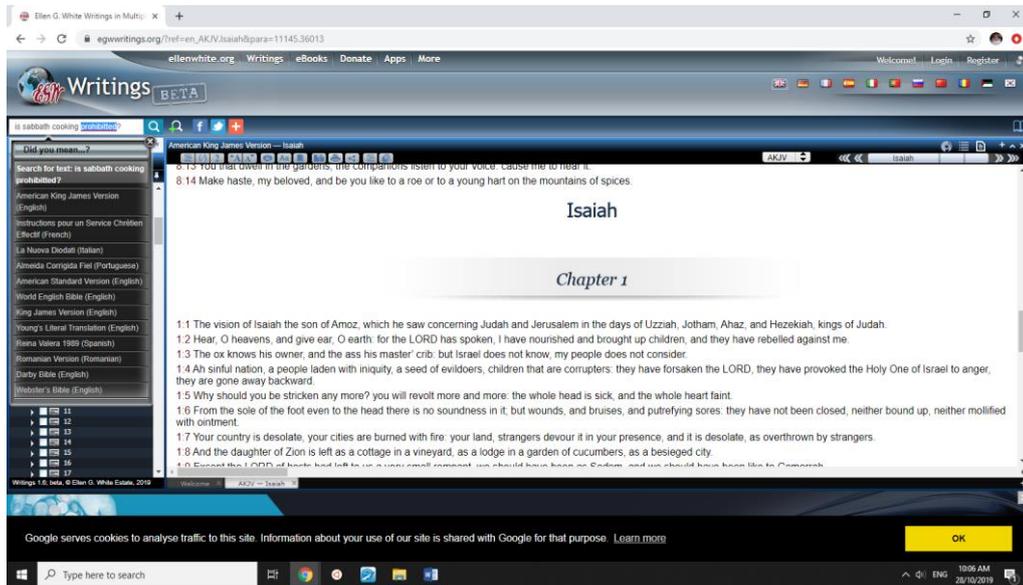
While this was ok for "keyword" search, a Q&A was more intuitive, although programmatically more challenging.

2. EGW Writings Android app had keyword search but could not answer simple question.
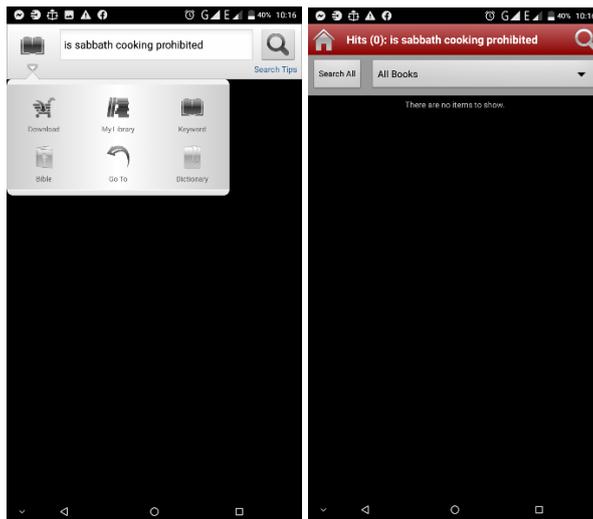
The keyword search feature was good but was not meant to answer questions.

3. Process models of existing systems.

While source code was not public for the systems available, it was apparent that they had not been designed to deal with question format queries. They had been designed as an index for keyword searches.

4. Data models of the existing systems

Originally EGW Writings were in books, pamphlets, letters, and articles in magazines. They have been digitized over the years and are freely available thru the website, the apps, and in CDROM/DVD format. There was a pattern in the existing digitized format which could be parsed for input to the proposed system.

5. Data that may be relevant to prove existence and seriousness of the identified problems.

An experiment[1] done by the author, showed that more people type in questions rather than keywords in Youtube to search for answers to their inquiries.

---

[1] http://winelfredpasamba.blogspot.com/2014/06/draft-062414-factors-affecting-online.html

II. REVIEW OF EXISTING ALTERNATIVES

The best way to find EGW quotes is thru googling. However non-EGW material may appear if one asks a question.

Another way is the EGW Comprehensive Topical Index. However, after finding the correct word, one had to scroll to the relevant usage or idea, and then click the reference to get to the text. Then scan the text for the quote.

Both the Topical Index and the Fulltext are very important for the proposed system.

The writer proposed to use the Fulltext and the Topical Index to build the system. Parsing the fulltext created an index. Combining it with the Topical Index helped the system have higher quality results. A database of questions was be built and grown as the system was being used, it became more intelligent over time. Answers from the indexes can be "Liked" and have more priority in the next queries. The algorithm of sorting answers was and will still be refined continuously. Initial list of questions was sourced from other religious websites like http://openbible.info/.

III. Methodology

A. Theoretical Framework

The general idea of the project is as follows:

```
                    ┌─────────────────┐   ┌─────────────────┐
                    │  User Questions │   │  User Feedback  │
                    └─────────────────┘   └─────────────────┘
  ┌─────────────────┐        │                    │
  │  EGW Fulltext   │──┐     │                    │
  └─────────────────┘  │     ▼                    ▼
  ┌─────────────────┐  │  ┌─────────────────────────┐
  │ EGW Topical Index│──┼─▶│     EGW Q&A System      │
  └─────────────────┘  │  └─────────────────────────┘
  ┌─────────────────┐  │     ▲
  │  Seed Questions │──┘
  └─────────────────┘
```
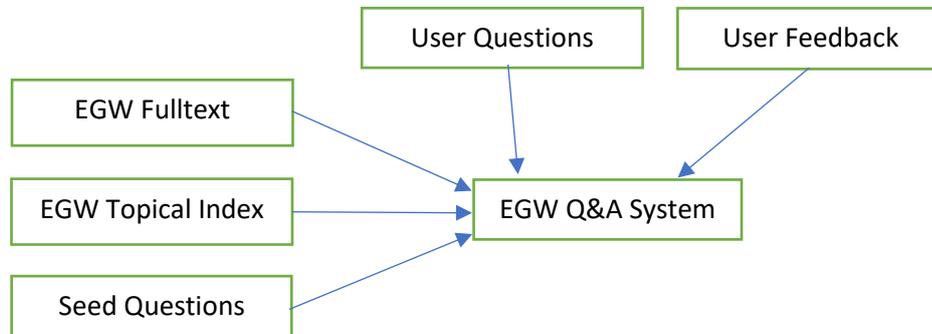
Combining several good databases as the EGW Fulltext, Topical Index, sample questions, and user feedback in the form of suggested answers or votes, was a great beneficial advance over existing EGW Writings search engines.

Furthermore the following ideas were used:

1. Information System Theories
    a. Indexing – is the collecting of all words in a text and lining up all the references like book, page number, paragraph number, line number, etc.. under each word. The benefit is searching of any word will eliminate the need to scan the whole text.
    b. Ranking – in Google times, ranking was the improvement Google made over search engines wherein more relevant results were shown before less relevant results.[2]
    c. Machine Learning – a system which learns from the environment (users, experiences, feedback, aside from static data).

---

[2] http://infolab.stanford.edu/~backrub/google.html

d. Fallback – the idea that in the absence of a higher quality answer, a lower quality answer is given.

2. System Design Principles

Spiral Model[3] is a development cycle where multiple iterations of the following steps are done until a release or deadline has been reached.

1.   Determine objectives

2.   Identify and resolve risks

3.   Development and Test

4.   Plan the next iteration

It prioritizes core functionality and improves the system in each cycle.

B. Rationale for the Framework

The proposed solution aimed to make it easier for users to find answers to their questions in a way that has not been utilized fully by combining existing databases (topical index, fulltext, seed questions from openbible.info), and utilized ranking of answers for specific questions. Ranking of answers is influenced by user feedback and algorithm and data tweaking, to give the best results.

C. Technologies Used

a. PostgreSQL database was used because it is free and mature.

b. PHP scripting language was used because it is free and popular. It is understood by many, and online support thru forums were abundant.

---

[3] https://en.wikipedia.org/wiki/Spiral_model

c. Linux operating system was used because it is free, and it offers the most flexibility for setup.

d. Memcache could be used to speed up popular and recent queries, and it is free. Specifically when website traffic makes the database server too busy.

IV. PROJECT PLAN

A. Concept

The idea was to make a system that tries to answer questions from EGW Writings data. Answers would come from an algorithm that would search from the following databases:

1) EGW Writings Fulltext index (we build our own)

2) EGW Topical Index (this was manually made in 1962, but this project re-indexed the index)

3) Questions from www.openbible.info (to seed)

The algorithm then was iteratively tested and fine-tuned to improve the quality of results.

Process Model
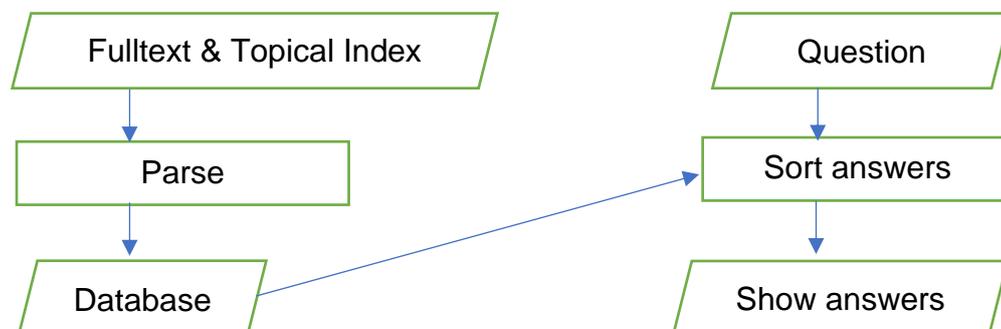
Data Model

egwwritings (referenceid int primary key, reference text, content text)

questions (questioned int primary key, question text)

answers (answerid int primary key, questioned int, relevance int, votes int, referenceid int)

Key Features

- Includes all of EGW Writings Fulltexts

- Includes EGW Comprehensive Topical Index – this is high quality index manually made and released in 1962.

- Growing question, answers, and ranking database as users ask more questions and vote / suggest good answers – copying success of http://openbible.info/

- Shows answers according to relevance – copying success from google pagerank

- Computes relevance with following factors – idea from google pagerank and openbible.info

  - Topical index entry
  - Mentions of keywords text
  - Popularity of phrase
  - User votes

B. Methods

1. Design Studies

People have religious questions[4]. The author has been part of and observed many ministries where listeners and viewers ask questions. In fact being able to ask questions is more interesting that just plain reading or listening or watching.

EGW Writings have answers to most questions people ask[5], but they needed to be matched with the questions. Not everyone knew the keywords to type in the keyword search. And using external service gave material which was less trusted.

The experience of Google Pagerank, http://openbible.info/, contributed good ideas to combine with the great resource of EGW Writings.

2. Review of Existing Systems

- http://egwwritings.org/ - This is the main web front for EGW Writings. It has a keyword search, but queries in question format are not well taken.

- https://whiteestate.org/about/issues/ - has some very good answers to questions but only about Ellen G. White and the issues surrounding interpretation, unusual statements, etc.. It does not include all EGW Writings

- EGW Writings Android/iOS Apps – contain keyword searches but not Q & A interface / database

---

[4] http://winelfredpasamba.blogspot.com/2014/06/draft-062414-factors-affecting-online.html
[5] Comprehensive Index to EGW Writings Volumes 1-4, 1962.

- EGW Writings DVD/CDROM – contains great Topical Index which is very high quality index, but one has to know what to search. It is still keyword based. Typing a question does not even remove stopwords.

- EGW Writings for Symbian OS – independently made by some programmer in Europe, but Symbian OS has lost popularity already.

3. Literature Review

- Google Pagerank Paper[6] - improved search results by ranking pages. Rank is computed by several factors, notably how many links a url has from other webpages, the url, the title, mentions in the text, meta-data, etc.

- http://openbible.info/ - has a good Question and Answer database which can be freely downloaded.

4. Assessment of Existing Alternatives

Alternatives are grouped into two:

1. EGW Search engines with topical search but no Q & A database
   - Website, android/ios app, cd/dvd
2. Q & A databases without emphasis / focus on EGW Writings
   - Google, openbible.info

Therefore, there was opportunity gap to do this project.

5. Production of Prototypes

- Author's partial success parsing EGW Writings from DVD successfully.

---

[6] http://infolab.stanford.edu/~backrub/google.html

- Author has also previously tried to parse EGW Topical Index successfully and put it into a database. Others have requested access, but it is still under testing and development.

- Author has tried to import questions and answers from openbible.info into own database. This is being used personally only.

- The above have all to be redone (many times) to fit them all together.

6. Tryouts of Prototypes with Users

Aside from personal use, author has tried to use the personal versions of the EGW Topical Index and the openbible.info Q&A data in church and classroom questions and answers where Google was not available. It has been a great help but can be improved a lot.

The partial success of the tryouts was the motivation for this project.

A temporary website was made to let academe from two Adventist institutions test the system. Students of Adventist International Institute of Advance Studies (AIIAS) and Adventist University of the Philippines (AUP).

User Interface Usability Testing Plan

Scope: search engine user interface of https://a.aolis.aup.edu.ph/egwcindex/

Purpose: To identify possible bottlenecks of the UI. To identify possible improvements to the UI. To get feedback from users how to make the UI more intuitive.

Schedule & Location: January-May 2020 at the EGW Estate in Adventist International Institute of Advanced Studies and at Adventist University of the Philippines, and online due to Taal volcano eruption and COVID-19.

Sessions: 30 minutes each participant.

Equipment: Desktop/laptop, and mobile/smartphone.

Participants: 5 users at least (enough for usability test[7])

Scenarios: participants were be asked to search for the following questions:

  a. is sabbath cooking is prohibited or not

  b. is coffee drinking good or not

  c. references that the Holy Spirit is also God

  d. prescribed length of dress for women

  e. advantages of not eating flesh foods

Metrics: Questions asked to participants prior to the sessions:

  a. How often do you search EGW writings? daily, weekly, monthly, yearly or less often

  b. How many minutes do you spend to find something?

Questions after the session:

  a. How many minutes did it take you to find the answer?

---

[7] https://www.usability.gov/how-to-and-tools/methods/recruiting-usability-test-participants.html

b. Are the answers sufficient or lacking?

Quantitative metrics: The following was be measured:

a. time to finish all questions

b. time from first typed letter to clicking search / enter

c. time scrolling thru results

d. number of pages browsed for each search

e. errors, if any

f. likes, dislikes, and recommendations

Roles:

a. developer – observe and record sessions

b. EGW Estate personnel – observe, facilitate, take notes

7. Development of Assessment Instruments

The real assessment was be the number of users who find it useful. Another good indicator is if EGW Estate later imports the idea into their main website. Another indication of success is the number of votes and clicks the system gets. Additionally, a questionnaire can be given to testers with the following questions:

1. Is it easier to get relevant answers to questions compared to using the keyword search in the App / CD/DVD / Website?
2. How would you prefer the answers to be sorted?

8. Analysis of assessment data

The author tried to understand the logs and see what users preferred and tried to enable them to vote on answers so the order of answers would be improved.

A higher percent of people who think the system is better at answering questions meant that the system is successful.

More traffic, more votes, and more reactions also meant success/popularity of the project.

9. Revision of prototypes

The Spiral Model of development was preferred as this project went thru many iterations of algorithm improvements to raise the quality of search results. The algorithm was continually monitored, improved and fine-tuned, and probably even after initial development.

10. Preparation of final project documentation/report

The ingest/import parsers, the ranking algorithm, and the Q&A database was continuously tested and improved for preparation towards this final project report.

Documentation was continuously improved based on suggestions from the software engineering advisory team composed of the adviser, the external evaluator, classmates, EGW Estate personnel, friends, and testers.

C. User testing and project assessment

1. How you determined if the problem identified has been successfully addressed?

If users found answers easier than the plain keyword search.

The system's base performance was the same with a plain text search engine, but after feeding information for a specific topic/query/question, it became highly efficient in giving the best answers to queries. This could be duplicated to all conceivable topics the more users utilize and shape the system.

2. What questions do you plan to focus on answering?

- Did you find the best answers within one click?
  - This was true for questions that were already primed. But questions that were not fell back to the basic keyword search needing several clicks and scrolls.
- Did you find the best answers with minimal scrolls?
  - Yes, for questions that were already primed. But questions that were not yet previously primed also fell back to basic keyword search performance.

3. What kinds of evidence were collected?

- Webserver log files
  - This was available at https://a.aolis.aup.edu.ph/egwcindex/sh.php
- Answer votes
  - These can be seen for each question. Answers that have higher votes were shown first.
- Search, click, hover, scroll logs
  - Click, hover, and scroll logs were planned but not used at this point of the experiment. Maybe should be moved at another phase.

- Survey answers

    o The survey answers showed what users were thinking and what they wished to improve. Very helpful. Though some ideas were better off done on next major overhauls.

4. What methods of inquiry were used to collect the evidences?

- yes/no

    o The questionnaire had to be short and easy because people don't have much time for questionnaires.

- Suggestions

    o People with plenty of ideas can give valuable input on open ended questions.

- Questions

    o See questionnaire and answers in the Survey Results and Discussion section.

4.5 What users, occasions, treatments, and outcomes were studied?

1. Anything that can help people get good answers to their questions.

    o A mix of students, scholars, pastors, friends, theology instructors, laymen were surveyed. Only a handful returned with answers maybe probably because of COVID-19 stress / distractions. This has to be studied more in the future.

5. How did you analyze and interpret your findings?

1. Simple statistics was sufficient to measure the feedback from the users.

2. It is possible that the project lost focus from the users because the results were linked to the main egw website. It is possible that more

users would use the index if they were not directed to read the fulltext on the main website of egwwritings.org.

3. It is possible that at this point of time of COVID-19 pandemic, not many people have time or interest for studies and experiments of this kind. Most people's minds are occupied by stress and how to survive the pandemic.

V. Results and Discussions

Words in the EGW Writings: 25,645,174

Paragraphs in EGW Writings: 250,582

Distinct words in EGW Writings: 37,843

Topics in Topical Index: 13,668

Lines in Topical Index: 284,911

Seed questions from openbible.info: 5,684

Algorithm major revisions: 5

Most queries finish in less than one second, some queries take 8 seconds.

Re-indexing, searching, parsing takes 25mins to 6 hours depending on which iteration of the index. About 12hrs to rebuilt everything from plaintext.

Almost same performance on i7-3635QM CPU @ 2.40GHz 8gb ram versus Xeon 4114 CPU @ 2.20GHz 16gb DDR4 @ 2666 MT/s.

Algorithm

The algorithm was changed about 5 major ways in this study.

Version 1: Index version 1. Parsing of references. Stopwords. Just to make it work. Plain indexing. Table too big.

Version 2: Index version 2. Database separate for index, and separate for writings and other tables. Separate table for each token/word. Great speed improvement.

Version 3. Topical Index included in separate database. Stopped computing for word instances in paragraph – better results.

Version 4. Stored aggregate results instead of computed.

Version 5. Facebook sharing optimization / beautification – customized meta-image and meta-description.

User Interface
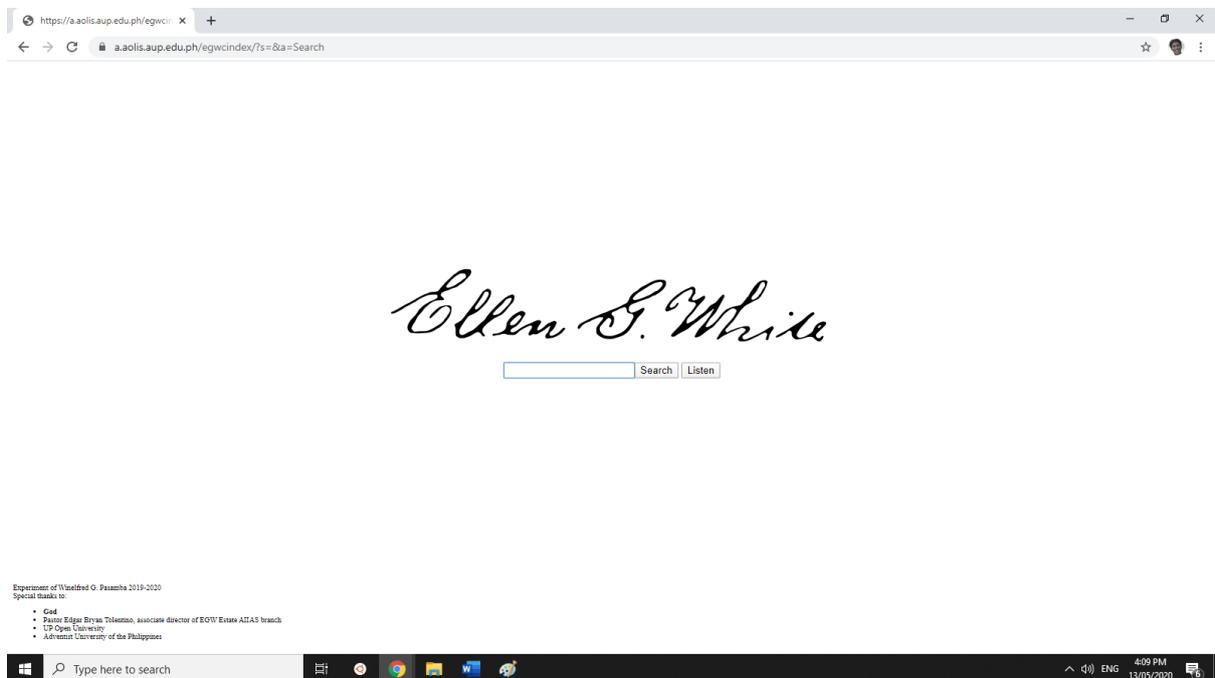
A clean user interface was made.
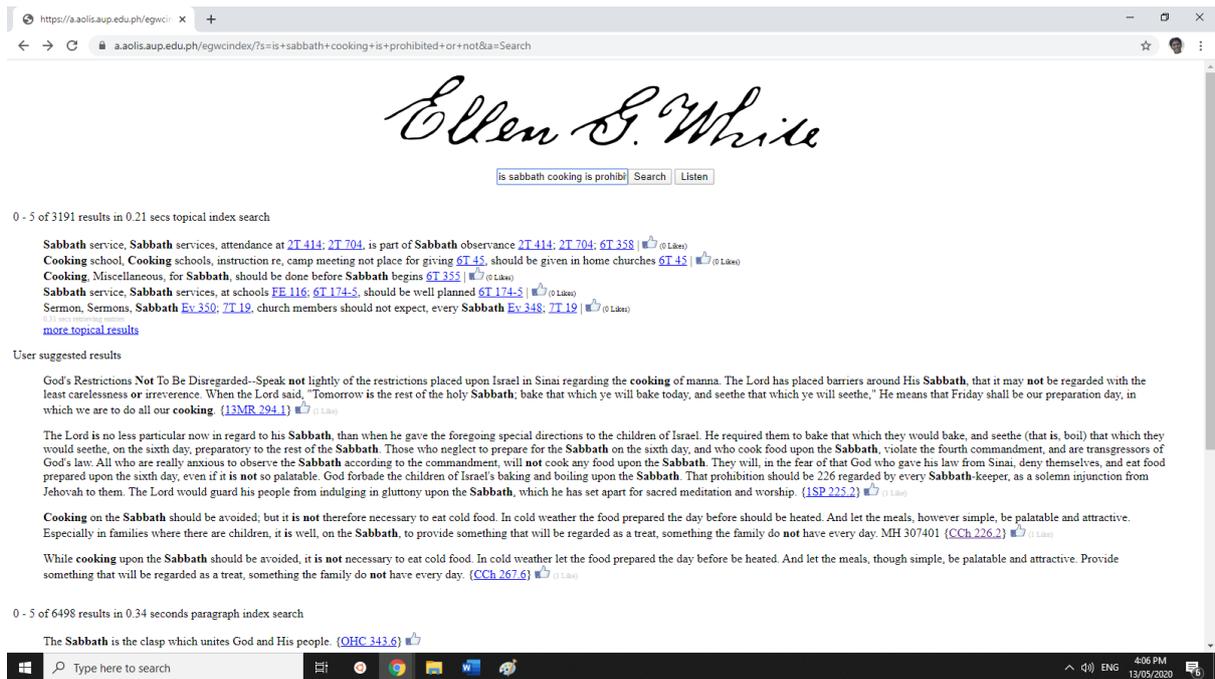
The user interface presents topical results, user suggested paragraphs, and index results.
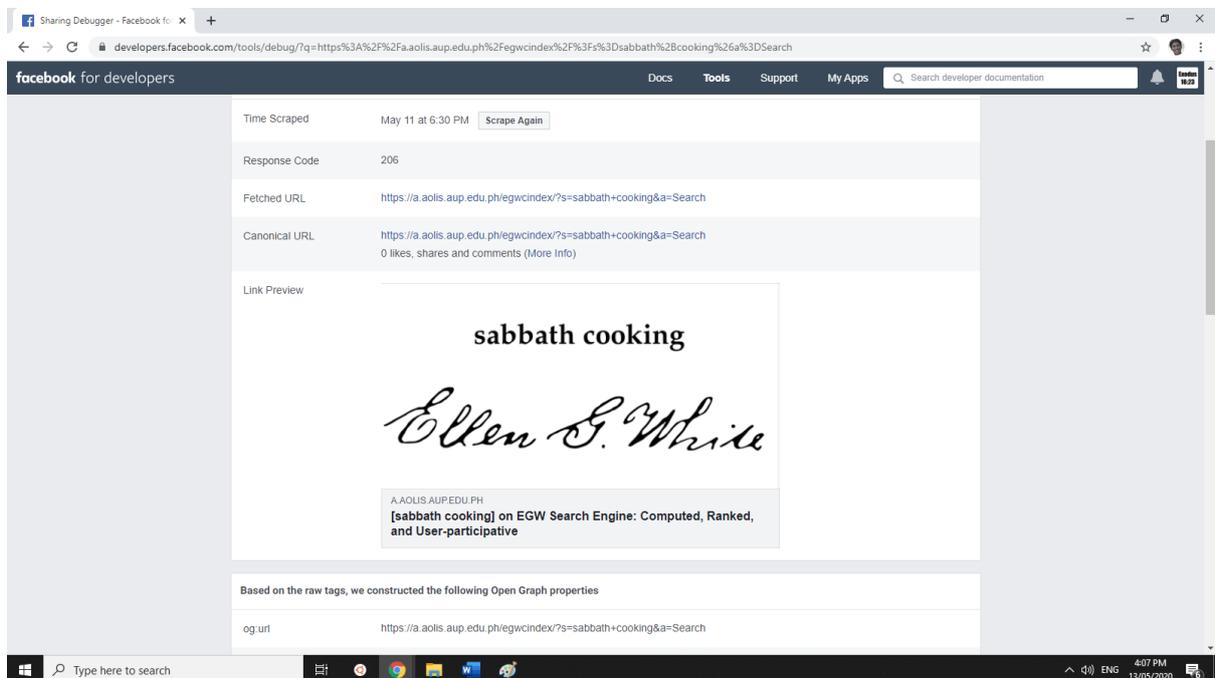

Facebook posts would include the search string in the picture and in the description.

Survey Results and Discussions

A google forms questionnaire was sent via social media to at least 5 groups and at least 50 individuals. Here are the results of the survey regarding their existing practice / usage:

How often do you search EGW writings? daily, weekly, monthly,yearly or less often

8 responses

- daily
- weekly
- monthly
- yearly or less
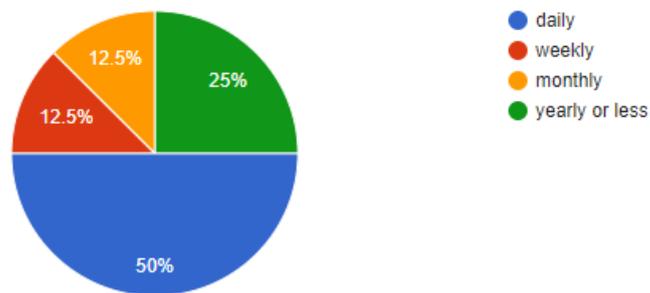
How many minutes do you spend to find something?

8 responses

- less than a minute
- around 5 minutes
- around 10 minutes
- around 30 minutes
- longer than 40 minutes
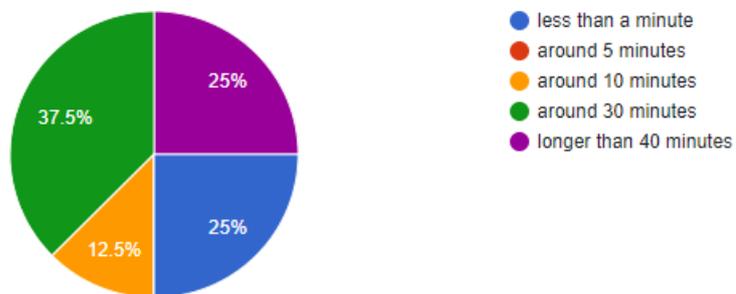
Here are the results after trying to use this project:

How many minutes did it take you to find the answer using the new search engine?

8 responses

● less than a minute
● around 5 minutes
● around 10 minutes
● around 30 minutes
● longer than 40 minutes
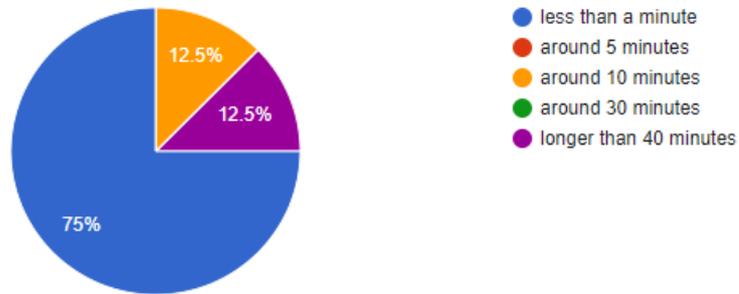
Are the answers sufficient or lacking?

8 responses

● sufficient
● lacking
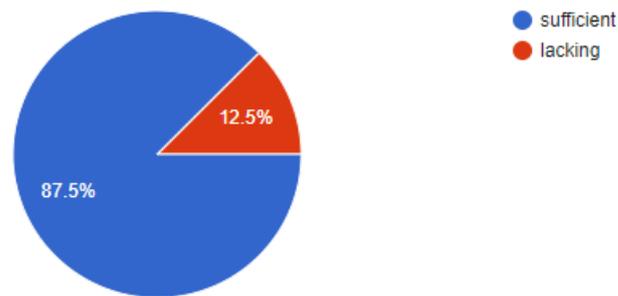
The results are amazing! Search time has decreased.

How many minutes to finish all the questions?

8 responses
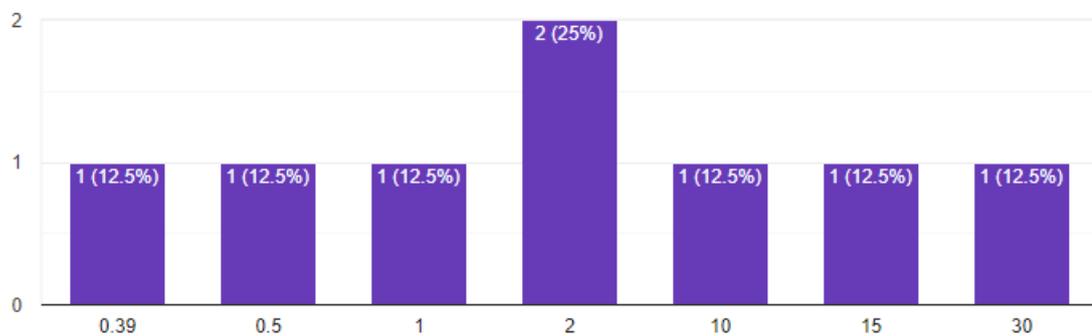
We are not sure if some respondents are very quick users, finishing in 0.39 minutes or if they actually understood the short question. Nevertheless it is interesting to see the distribution.

How many seconds to type a question?

8 responses
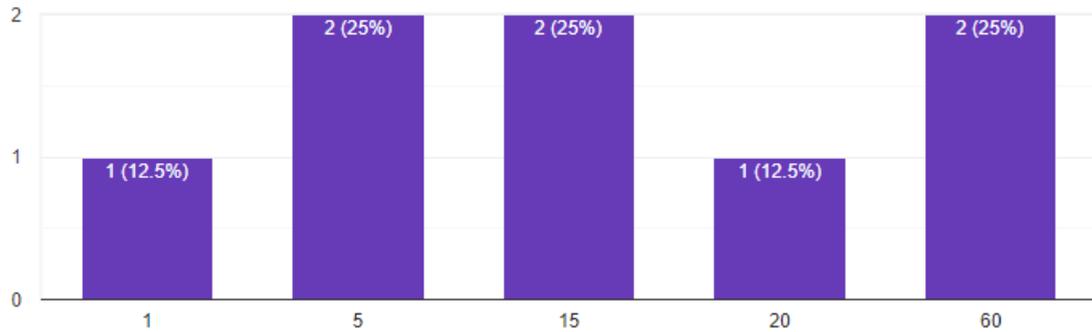
We see here that users are slow typers, or slow copy-and-paste'rs.

How many minutes spent scrolling thru the results?
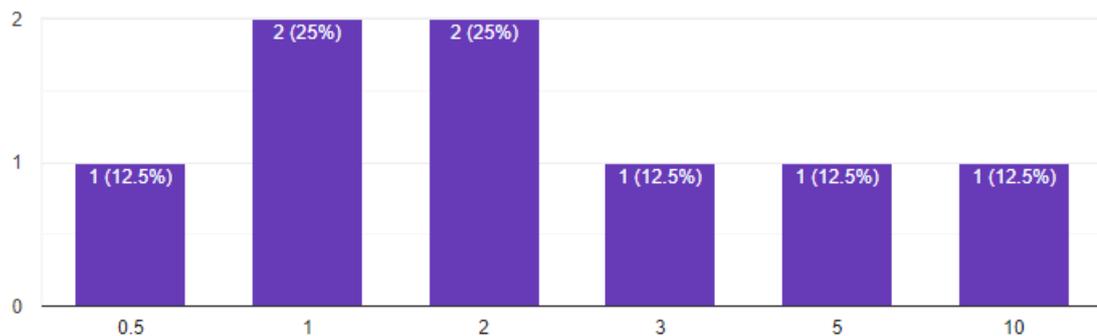
8 responses

Interesting. Those who did not scroll much probably found what they wanted. And those who scrolled a lot probably wanted to read more about the topic.

How many minutes spent scrolling thru the results?

8 responses

Value	Count
0.5	1 (12.5%)
1	2 (25%)
2	2 (25%)
3	1 (12.5%)
5	1 (12.5%)
10	1 (12.5%)
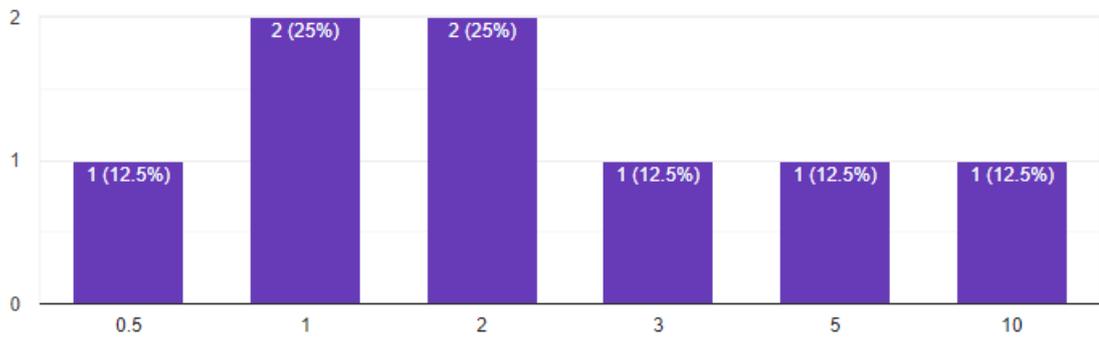
How many pages of results browsed?

8 responses

Value	Count
2	4 (50%)
3	1 (12.5%)
5	1 (12.5%)
10	2 (25%)
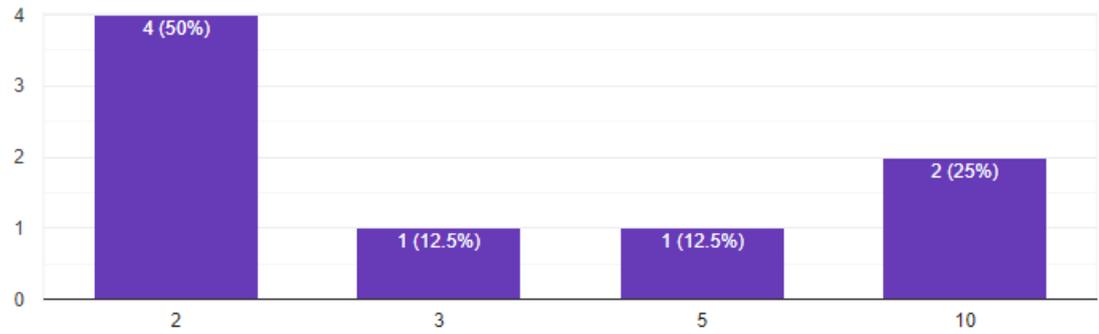
Errors found (if any)?

6 responses

0
No
n
None

Making a complex system look simple to use makes troubleshooting also simple and error less or unfound.

Interesting, but the author has a different solution for including church documents – that is a google customized engine or "cse".

Regarding the design, it can be improved.
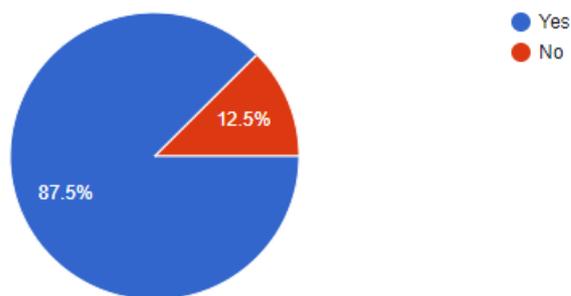
How would you prefer the answers to be sorted?

5 responses

As provided
Chronological
Tohachi Tampo
Capellanus MJ
Lexi

The "chronological" here means that since the EGW Estate has put a year on each paragraph, as the year that paragraph was written, it would be nice to see a chronological sequence of results for a specific query.

This is the best result ever.

Are answers easier to find here compared to existing ones (EGW mobile app, EGW cd/dvd, egwwritings website)?

8 responses

● Yes
● No

12.5%

87.5%
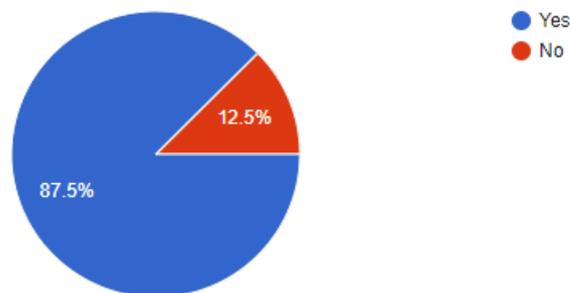
This is the other best result ever. God has given us success. Praise the LORD!

VI. CONCLUSIONS

The objectives of the study have been met and the author is very happy with the results. Indexing the EGW Writings and the Topical Index and combining the seed questions from openbible.info into an algorithm and database architecture has been a great experience to the author.

Users are very happy about the project.

We thank God for the miracles in this project.

VII. RECOMMENDATIONS

For future studies, experiments, and or projects, the following are recommended:

1. That more users use the system and contribute their favorite paragraphs as user-suggested references to queries or questions they are interested in.

2. That EGW Estate study how to use this algorithm (or the ideas herein) in their search engines, desktop apps, and mobile apps.

3. When the website traffic reaches a level where the database becomes the bottleneck, database results should be Memcached to increase server capacity.

The following are user contributed thru the survey and are valuable:

4. Paragraphs can be timestamped, and results can be sorted by date.

5. Better graphical design.

VIII. REFERENCES

Ellen G. White Writings Website
http://egwwritings.org/

OpenBible.info Topical Bible
http://openbible.info/topics/

The Anatomy of a Large-Scale Hypertextual Web Search Engine
Sergey Brin and Lawrence Page
Computer Science Department, Stanford University, Stanford, CA 94305
http://infolab.stanford.edu/~backrub/google.html

PHP Manual
https://www.php.net/manual/en/index.php

PostgreSQL Manual
https://www.postgresql.org/docs/manuals/

IX. APPENDICES

A. Deliverables and Milestones

The project planning was 6 months, while the construction was another 6 months.

Activity \ Month	1	2	3	4	5	6	7	8	9	10	11	12
Proposal	■	■	■	■	■	■						
Prototype							■	■				
User Testing and assessment							■	■	■			
Assessment Report							■	■	■	■		
Suggestions for revisions							■	■	■	■	■	

As soon as there was a workable website, a select group of users were testing, assessing, and giving suggestions for the improvements. This was because the information system is basically easy, but the algorithm fine-tuning and the question database, or the self-learning feature, were the challenging parts which required significant time in improving.

B. Budget

- $100 for hosting

C. Qualifications of Author

1. ~20 years of database applications development, and maintenance
2. ~19 years teaching computer science major subjects
3. Interest in EGW Writings. Interest in text processing. Interest in Q&As. Interest in solving the gap between Q&As and EGW Writings corpus.
4. BS Computer Science, 2000. Adventist University of the Philippines

D. Contributors

1. Pastor Edgar Bryan Tolentino, EGW Estate Associate Director, supported this project.

E. Resources

1. EGW Writings Research Edition DVD (at hand, complementary gift from EGW Estate)

F. Complete Program Listing

index.php
```
<?php
include "egwmain1.php";
```

egwmain1.php
```
<?php
        $mt = microtime (true);
        include "dbfuncs.php";
        $s = @$_GET['s'];
        $limit = @$_GET['limit'];
        if ($limit < 1) $limit = 20;

if (strlen($s)) {
        echo "<meta property=\"og:title\"            content=\"[$s] on EGW Search
Engine: Computed, Ranked, and User-participative\" />";
        echo "<meta property=\"og:image\"
content=\"https://a.aolis.aup.edu.ph/egwcindex/i.php?s=".urlencode($s)."\" />";
} else {
        echo "<meta property=\"og:title\"            content=\"Experimental EGW
Search Engine: Computed, Ranked, and User-participative\" />";
        echo "<meta property=\"og:image\"
content=\"https://a.aolis.aup.edu.ph/egwcindex/egw.png?2020050802\" />";
}

?>
<meta property="og:description"      content="Experimental Computed Search
Engine to EGW Writings" />
<meta property="og:url"           content="<?php echo
$_SERVER['REQUEST_SCHEME']."://".$_SERVER['HTTP_HOST'].$_SERVER['RE
QUEST_URI']; ?>" />
<meta property="og:type"            content="website" />
<meta property="fb:appid"           content="234457047969920" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<?php
if (!strlen($s))
        echo "<table width=100% height=100% border=0 cellpadding=0
cellspacing=0><tr><td valign=bottom>";
?>
<center><img src=egw0.png><br><form><br>
<input autofocus type=text name=s id=s value="<? echo $s; ?>"><input type=submit
name=a value=Search>
<input type=button value=Listen onclick=recognition.start()>
<style>
img {
        max-width: 100%;
   max-height: 100%;}
</style>
<script>
```

```javascript
    var recognition = new webkitSpeechRecognition();
    recognition.continuous = false;
    recognition.interimResults = true;

    recognition.onresult = function (e) {
       var textarea = document.getElementById('s');
       for (var i = e.resultIndex; i < e.results.length; ++i) {
          if (e.results[i].isFinal) {
             //textarea.value += "..." + e.results[i][0].transcript;
             textarea.value = e.results[i][0].transcript;
                             document.getElementById("form1").submit();
          }
       }
    }

    // start listening
//    recognition.start();

        getText = function (url, callback)
        {
                var request = new XMLHttpRequest();
                request.onreadystatechange = function()
                {
                        if (request.readyState == 4 && request.status == 200)
                        {
                                callback(request.responseText);
                        }
                };
                request.open('GET', url);
                request.send();
        }

        function submitquestionwritinglike (q, ref) {
                function mycallback(data) {
                        alert (data);
                }
                getText ('egwwritingquestionlikelog.php?q='+q+'&ref='+ref,
mycallback);
        }
        function submitquestiontopiclike (q, topic) {
                function mycallback(data) {
                        alert (data);
                }
                getText ('egwtopicquestionlikelog.php?q='+q+'&topic='+topic,
mycallback);
        }
</script>
</center>
</form>
<script>
```

```javascript
getText = function (url, callback)
{
        var request = new XMLHttpRequest();
        request.onreadystatechange = function()
        {
                if (request.readyState == 4 && request.status == 200)
                {
                        callback(request.responseText);
                }
        };
        request.open('GET', url);
        request.send();
}

function loadtopics (s) {
        topical.innerHTML = "loading topics...";
        function mycallback(data) {
                topical.innerHTML = data;
        }
        getText ('egwtopicalsearch5.php?limit=5&ro=1&s='+s, mycallback);
}

function loadparagraphs (s) {
        paragraphs.innerHTML = "loading paragraphs...";
        function mycallback(data) {
                paragraphs.innerHTML = data;
        }
        getText ('egwrankedindex5.php?limit=5&ro=1&s='+s, mycallback);
}
</script>
<?php
if (strlen ($s)) { ?>
        <div id=topical onload==></div>
        <div id=paragraphs onload=loadparagraphs()></div>
        <script>
        loadtopics ("<?php echo $s; ?>");
        loadparagraphs ("<?php echo $s; ?>");
        </script>
        <?php
}
if (!strlen($s))
        echo "<tr><td valign=bottom><font size=1>";
?>
Experiment of Winelfred G. Pasamba 2019-2020<br>
Special thanks to:<ul>
<li><b>God</b>
<li>Pastor Edgar Bryan Tolentino, associate director of EGW Estate AIIAS branch
<li>UP Open University
<li>Adventist University of the Philippines
</ul>
```

```php
<?php
if (!strlen($s))
        echo "</table>";
?>
```
egwtopicalsearch5.php
```php
<?php
        $mt = microtime (true);
        include "dbfuncs.php";
        $s = @$_GET['s'];
        $limit = @$_GET['limit'];
        $debug = @$_GET['debug'];
        $debug = 0;
        if (isset ($_GET['start']))
                $start = $_GET['start'];
        else
                $start = 0;
        if (!$debug) error_reporting (0);
        if ($limit < 1) $limit = 20;

if (!isset($_GET['ro'])) { ?>
        <form method=GET id=form1>
        <center><h2>EGW Ranked Index Search</h2>
        <input autofocus type=text name=s id=s value="<? echo $s; ?>"><input
type=submit name=a value=Search>
        <input type=button value=Listen onclick=recognition.start()>
        <script>
                var recognition = new webkitSpeechRecognition();
                recognition.continuous = false;
                recognition.interimResults = true;

                recognition.onresult = function (e) {
                        var textarea = document.getElementById('s');
                        for (var i = e.resultIndex; i < e.results.length; ++i) {
                                if (e.results[i].isFinal) {
                                        //textarea.value += "..." + e.results[i][0].transcript;
                                        textarea.value = e.results[i][0].transcript;
                                        document.getElementById("form1").submit();
                                }
                        }
                }

                // start listening
        //    recognition.start();

        </script>
        </center>
        </form>
        <?php
}
        ?>
```

```
<script>
getText = function (url, callback)
{
        var request = new XMLHttpRequest();
        request.onreadystatechange = function()
        {
                if (request.readyState == 4 && request.status == 200)
                {
                        callback(request.responseText);
                }
        };
        request.open('GET', url);
        request.send();
}
function submitquestionwritinglike (q, ref) {
        function mycallback(data) {
                alert ("Thank You!");
        }
        getText ('egwwritingquestionlikelog.php?q='+q+'&ref='+ref,
mycallback);
}
function submitquestiontopiclike (q, topic) {
        function mycallback(data) {
                alert ("Thank You!");
        }
        getText ('egwtopicquestionlikelog.php?q='+q+'&topic='+topic,
mycallback);
}
</script>
<?php

if (strlen ($s) == 0) exit(0);
$s = pg_escape_string ($s);
$delims = " \\|\t\n.,<>?/.,\":';}{][~!~!@#$%^&*()_+=-}{][`";
$booksregex =
'(AA|AH|ApM|AY|CME|CSA|CG|CTr|COL|CE|CET|ChL|ChS|CTBH|CEv|CM|CC|Con|
CCh|CD|CH|CSW|CS|CT|CW|CL|DD|DG|DA|EW|Ed|Ev|FW|FLB|FE|AG|GW92|GW|
GC88|GC|Hvn|HFM|HL|HDL|HS|HP|LDE|LYL|LS80|LS88|LS|LHU|MC|Mar|MM|MYP|
1MCP|2MCP|MH|ML|NL|OFC|OHC|PaM|PP|PCP|Pr|PK|PM|RC|RY|SL|1SM|2SM|3S
M|1BC|2BC|3BC|4BC|5BC|6BC|7BC|7ABC|ExV|LP|SA|SD|SW|SpTEd|1SP|2SP|3SP
|4SP|1SG|2SG|3SG|4aSG|4bSG|SC|SJ|SR|ExV54|Te|1T|2T|3T|4T|5T|6T|7T|8T|9T|
TSS|TSB|TM|TSA|TSDF|TMK|TDG|MB|BLJ|TA|UL|VSS|WM|WLF|YRP)';

        $stopwords =
",a,able,about,above,abst,accordance,according,accordingly,across,act,actually,add
ed,adj,affected,affecting,affects,after,afterwards,again,against,ah,all,almost,alone,alo
ng,already,also,although,always,am,among,amongst,an,and,announce,another,any,
anybody,anyhow,anymore,anyone,anything,anyway,anyways,anywhere,apparently,a
pproximately,are,aren,arent,arise,around,as,aside,ask,asking,at,auth,available,away,
awfully,b,back,be,became,because,become,becomes,becoming,been,before,before
```

hand,begin,beginning,beginnings,begins,behind,being,believe,below,beside,besides,between,beyond,biol,both,brief,briefly,but,by,c,ca,came,can,cannot,can't,cause,causes,certain,certainly,co,com,come,comes,contain,containing,contains,could,couldnt,d,date,did,didn't,different,do,does,doesn't,doing,done,don't,down,downwards,due,during,e,each,ed,edu,effect,eg,eight,eighty,either,else,elsewhere,end,ending,enough,especially,et,et-al,etc,even,ever,every,everybody,everyone,everything,everywhere,ex,except,f,far,few,ff,fifth,first,five,fix,followed,following,follows,for,former,formerly,forth,found,four,from,further,furthermore,g,gave,get,gets,getting,give,given,gives,giving,go,goes,gone,got,gotten,h,had,happens,hardly,has,hasn't,have,haven't,having,he,hed,hence,her,here,hereafter,hereby,herein,heres,hereupon,hers,herself,hes,hi,hid,him,himself,his,hither,home,how,howbeit,however,hundred,i,id,ie,if,i'll,im,immediate,immediately,importance,important,in,inc,indeed,index,information,instead,into,invention,inward,is,isn't,it,itd,it'll,its,itself,i've,j,just,k,keep

keeps,kept,kg,km,know,known,knows,l,largely,last,lately,later,latter,latterly,least,less,lest,let,lets,like,liked,likely,line,little,'ll,look,looking,looks,ltd,m,made,mainly,make,makes,many,may,maybe,me,mean,means,meantime,meanwhile,merely,mg,might,million,miss,ml,more,moreover,most,mostly,mr,mrs,much,mug,must,my,myself,n,na,name,namely,nay,nd,near,nearly,necessarily,necessary,need,needs,neither,never,nevertheless,new,next,nine,ninety,no,nobody,non,none,nonetheless,noone,nor,normally,nos,not,noted,nothing,now,nowhere,o,obtain,obtained,obviously,of,off,often,oh,ok,okay,old,omitted,on,once,one,ones,only,onto,or,ord,other,others,otherwise,ought,our,ours,ourselves,out,outside,over,overall,owing,own,p,page,pages,part,particular,particularly,past,per,perhaps,placed,please,plus,poorly,possible,possibly,potentially,pp,predominantly,present,previously,primarily,probably,promptly,proud,provides,put,q,que,quickly,quite,qv,r,ran,rather,rd,re,readily,really,recent,recently,ref,refs,regarding,regardless,regards,related,relatively,research,respectively,resulted,resulting,results,right,run,s,said,same,saw,say,saying,says,sec,section,see,seeing,seem,seemed,seeming,seems,seen,self,selves,sent,seven,several,shall,she,shed,she'll,shes,should,shouldn't,show,showed,shown,showns,shows,significant,significantly,similar,similarly,since,six,slightly,so,some,somebody,somehow,someone,somethan,something,sometime,sometimes,somewhat,somewhere,soon,sorry,specifically,specified,specify,specifying,still,stop,strongly,sub,substantially,successfully,such,sufficiently,suggest,sup,sure

t,take,taken,taking,tell,tends,th,than,thank,thanks,thanx,that,that'll,thats,that've,the,their,theirs,them,themselves,then,thence,there,thereafter,thereby,thered,therefore,therein,there'll,thereof,therere,theres,thereto,thereupon,there've,these,they,theyd,they'll,theyre,they've,think,this,those,thou,though,thoughh,thousand,throug,through,throughout,thru,thus,til,tip,to,together,too,took,toward,towards,tried,tries,truly,try,trying,ts,twice,two,u,un,under,unfortunately,unless,unlike,unlikely,until,unto,up,upon,ups,us,use,used,useful,usefully,usefulness,uses,using,usually,v,value,various,'ve,very,via,viz,vol,vols,vs,w,want,wants,was,wasnt,way,we,wed,welcome,we'll,went,were,werent,we've,what,whatever,what'll,whats,when,whence,whenever,where,whereafter,whereas,whereby,wherein,wheres,whereupon,wherever,whether,which,while,whim,whither,who,whod,whoever,whole,who'll,whom,whomever,whos,whose,why,widely,willing,wish,with,within,without,wont,words,world,would,wouldnt,www,x,y,yes,yet,you,youd,you'll,your,youre,yours,yourself,yourselves,you've,z,zero,";

```
        $foundtopic = 0;
        $sql = "select * from egwtopicalindexflat where lower(entry) like lower('$s%')
order by entry;";
```

```
//echo $sql;
$res = dbexec ($sql);
$nres = pg_numrows ($res);
if ($res && $nres) {
        $foundtopic = 1;
        echo "<br>".min($start,$nres)." - ".min($start+$limit,$nres)." of $nres
results in ";
        echo sprintf("%.2f",(microtime(true) - $mt))." secs topical search";
        echo "<blockquote>";
        for ($i = max(0, $start); $i < min($start+$limit,$nres); $i++) {
                $text = pg_result ($res, $i, "entry");

                $body = preg_replace("/\b($s)\b/i", "<b>$1</b>", $text);

                // replaces the commas with books - winwin :D God helped me
to do this April 27, 2020
                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);

                // replace references with links - winwin :D God helped me to do
this April 27, 2020
                //$body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])*/',
'<a href=>${1}${2}${3}${4}</a>', $body);
                $body = preg_replace ('/'.$booksregex.' ([0-9]+)(:|-)?([0-9]+)*/',
'<a target=_egw href=https://egwwritings.org/?ref=en_${1}.${2}.${4}>${1}
${2}${3}${4}</a>', $body);
                $text = $body;

                echo $text;
                echo " | <Font size=1><a onclick=\"submitquestiontopiclike('$s',
'$entry')\"><img src=like.png></a>";
                $cc = dbexec ("select count(*) from egwquestiontopicliklog
where question = '$s' and topic = '".pg_escape_string($entry)."';");
                echo " (".pg_result ($cc, 0, 0)." Likes)</font>";
                echo " <font color=white>".pg_result ($res, $i, "sum")."pts</font>
";
```

```php
                    echo "<br>";
                    flush (); ob_flush();

            }
            echo "</blockquote>";
    }

    if (!$foundtopic) {
            $sqlb = $sql = "";

            //$sql .= "select topicid, /*10 as prority,*/ topic, body from
egwtopicalindex where topic = '$s' ";
            //$sql .= "select reference, sum(count) from (select word, reference,
count(reference) from egwindex where FALSE group by word, reference ";

            $tok = strtok ($s, $delims);
            $sql = "select topicid, parlen, max(sum) as sum from (select topicid,
ref, parlen, sum(count) as sum from (";
            $sqlb = $sql;
            $sqlx = "(";
            $sqlu = "(";
            $words = 0;
            $vtoks = 0;
            $valids = "";
            $sqlbx = "(";
            $sqlbu = "(";
            $wordsb = 0;
            $vtoksb = 0;
            $validsb = "";
            while ($tok !== false) {
                    $words++;
                    $wordsb++;
                    $stok = pg_escape_string (strtolower($tok));
                    if (!stristr($stopwords, ",$tok,")) {
                            $valids .= " $tok ";
                            $vtoks++;
                            if ($vtoks > 1) {
                                    $sqlu .= " UNION ALL ";
                                    $sqlx .= " INTERSECT ALL ";
                            }
                            $sqlu .= " select topicid, ref, parlen, count(ref)*1 as count
from raw_$stok group by topicid, ref, parlen ";
                            $sqlx .= " select topicid, ref, parlen, count(ref)*20.01 as
count from raw_$stok group by topicid, ref, parlen ";
                    }
                    $validsb .= " $tok ";
                    $vtoksb++;
                    if ($vtoksb > 1) {
                            $sqlbu .= " UNION ALL ";
                            $sqlbx .= " INTERSECT ALL ";
```

```php
				}
				$sqlbu .= " select topicid, ref, parlen, count(ref)*1 as count from raw_$stok group by topicid, ref, parlen ";
				$sqlbx .= " select topicid, ref, parlen, count(ref)*20.01 as count from raw_$stok group by topicid, ref, parlen ";

				$tok = strtok ($delims);
			}
			$sqlu .= ")";
			$sqlx .= ")";
			$sql .= " ($sqlx) UNION ($sqlu) ) as foo group by topicid, ref, parlen order by sum desc, parlen asc, topicid) as bar group by topicid, parlen order by sum desc, parlen asc";
			$sqlbu .= ")";
			$sqlbx .= ")";
			$sqlb .= " ($sqlbx) UNION ($sqlbu) ) as foo group by topicid, ref, parlen order by sum desc, parlen asc, topicid) as bar group by topicid, parlen order by sum desc, parlen asc";
			if ($debug) echo $sql;
			$includedstopwords = 0;
			$sratio = (strlen ($valids)+1)/(1+strlen($validsb));
			if ($debug) echo "SRATIO:$sratio ";
			if ($sratio > 0.1) {
				$res = dbexec ($sql, "egwtopicalindexindex");
				$includedstopwords = 0;
			} else {
				$valids = $validsb;
				$words = $wordsb;
				$sql = $sqlb;
				$vtoksb = $vtoks;
				if ($debug) echo $sql;
				$res = dbexec ($sql, "egwtopicalindexindex");
				$includedstopwords = 1;
			}
			if (!$includedstopwords && pg_numrows ($res) < 1) {
				if ($debug) echo "0 results without stopwords. trying with stopwords.<br>";
				$valids = $validsb;
				$words = $wordsb;
				$sql = $sqlb;
				$vtoksb = $vtoks;
				if ($debug) echo $sql;
				$res = dbexec ($sql, "egwtopicalindexindex");
			}
			if ($res)
				$nresults = pg_numrows ($res);
			else
				$nresults = 0;
```

```php
                dbexec ("insert into egwsearchlog (url, nresults, datetime) values
('".pg_escape_string($_SERVER['REQUEST_URI'])."','$nresults','now'::timestamp);")
;
                echo "<br>".min($start,$nresults)." - ".min($start+$limit,$nresults)." of
$nresults results in ";
                echo sprintf("%.2f",(microtime(true) - $mt))." secs topical index search";
                //echo "$s -> $valids<br>";
                echo "<blockquote>";
                for ($i = max(0, $start); $i < min($start+$limit,pg_numrows ($res));
$i++) {
                        $topicid = pg_result ($res, $i, "topicid");
                        //echo $topicid;
                        $rsql = "select * from egwtopicalindexflat where topicid =
'$topicid'";

                        //echo $rsql;
                        $resq = dbexec($rsql);
                        for ($qi = 0; $qi < pg_numrows ($resq); $qi++) {
                                $entry = $text = pg_result ($resq, $qi, "entry");
                                //echo $s;

                                $tok = strtok ($valids, $delims);
                                while ($tok !== false) {
                                        //if (!stristr($stopwords, ",$tok,")) {
                                                $text = preg_replace("/\b($tok)\b/i",
"<b>$1</b>", $text);

                                                $tok = strtok ($delims);
                                        //}
                                }

                                $body = $text;
                                // replaces the commas with books - winwin :D God
helped me to do this April 27, 2020
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
                                $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-
9])+,( [0-9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
```

```php
                              // replace references with links - winwin :D God helped me to do this April 27, 2020
                              //$body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])*/', '<a href=>${1}${2}${3}${4}</a>', $body);
                              $body = preg_replace ('/'.$booksregex.' ([0-9]+)(:|-)?([0-9]+)*/', '<a target=_egw href=https://egwwritings.org/?ref=en_${1}.${2}.${4}>${1} ${2}${3}${4}</a>', $body);
                              $text = $body;

                              echo $text;
                              echo " | <Font size=1><a onclick=\"submitquestiontopiclike('$s', '$entry')\"><img src=like.png></a>";
                              $cc = dbexec ("select count(*) from egwquestiontopiclikelog where question = '$s' and topic = '".pg_escape_string($entry)."';");
                              echo " (".pg_result ($cc, 0, 0)." Likes)</font>";
                              echo " <font color=white>".pg_result ($res, $i, "sum")."pts</font> ";
                      }
                      echo "<br>";
                      flush (); ob_flush();
              }
      }

      echo "<font color=lightgray size=1>".sprintf("%.2f",(microtime(true) - $mt))." secs retrieving entries</font>";
      echo "<br><a href=".$_SERVER["PHP_SELF"]."?start=".($start+$limit)."&s=".urlencode($s).">more topical results</a>";
      echo "</blockquote>";
?>
```

egwrankedindex5.php

```php
<?php
      $mt = microtime (true);
      include "dbfuncs.php";
      $s = @$_GET['s'];
      $limit = @$_GET['limit'] * 1;
      $debug = @$_GET['debug'] * 1;
      if (isset ($_GET['start']))
              $start = $_GET['start'];
      else
              $start = 0;

      $debug = 0;
      if (!$debug) error_reporting (0);
      if ($limit < 1) $limit = 20;

      if (!isset($_GET['ro'])) { ?>
```

```php
		<form method=GET id=form1>
		<center><h2>EGW Ranked Index Search</h2>
		<input autofocus type=text name=s id=s value="<? echo $s; ?>"><input
type=submit name=a value=Search>
		<input type=button value=Listen onclick=recognition.start()>
		<script>
			var recognition = new webkitSpeechRecognition();
			recognition.continuous = false;
			recognition.interimResults = true;

			recognition.onresult = function (e) {
				var textarea = document.getElementById('s');
				for (var i = e.resultIndex; i < e.results.length; ++i) {
					if (e.results[i].isFinal) {
						//textarea.value += "..." + e.results[i][0].transcript;
						textarea.value = e.results[i][0].transcript;
						document.getElementById("form1").submit();
					}
				}
			}

			// start listening
	//	recognition.start();


		</script>
		</center>
		</form>
		<?php
}

		if (strlen ($s) == 0) exit(0);

		$s = pg_escape_string ($s);
		$delims = " \\|\t\n.,<>?/.,\":';}{][~!~!@#$%^&*()_+=-}{[`";
		$stopwords =
",a,able,about,above,abst,accordance,according,accordingly,across,act,actually,add
ed,adj,affected,affecting,affects,after,afterwards,again,against,ah,all,almost,alone,alo
ng,already,also,although,always,am,among,amongst,an,and,announce,another,any,
anybody,anyhow,anymore,anyone,anything,anyway,anyways,anywhere,apparently,a
pproximately,are,aren,arent,arise,around,as,aside,ask,asking,at,auth,available,away,
awfully,b,back,be,became,because,become,becomes,becoming,been,before,before
hand,begin,beginning,beginnings,begins,behind,being,believe,below,beside,besides,
between,beyond,biol,both,brief,briefly,but,by,c,ca,came,can,cannot,can't,cause,caus
es,certain,certainly,co,com,come,comes,contain,containing,contains,could,couldnt,d,
date,did,didn't,different,do,does,doesn't,doing,done,don't,down,downwards,due,durin
g,e,each,ed,edu,effect,eg,eight,eighty,either,else,elsewhere,end,ending,enough,esp
ecially,et,et-
al,etc,even,ever,every,everybody,everyone,everything,everywhere,ex,except,f,far,fe
w,ff,fifth,first,five,fix,followed,following,follows,for,former,formerly,forth,found,four,fro
```

m,further,furthermore,g,gave,get,gets,getting,give,given,gives,giving,go,goes,gone,got,gotten,h,had,happens,hardly,has,hasn't,have,haven't,having,he,hed,hence,her,here,hereafter,hereby,herein,heres,hereupon,hers,herself,hes,hi,hid,him,himself,his,hither,home,how,howbeit,however,hundred,i,id,ie,if,i'll,im,immediate,immediately,importance,important,in,inc,indeed,index,information,instead,into,invention,inward,is,isn't,it,itd,it'll,its,itself,i've,j,just,k,keep

keeps,kept,kg,km,know,known,knows,l,largely,last,lately,later,latter,latterly,least,less,lest,let,lets,like,liked,likely,line,little,'ll,look,looking,looks,ltd,m,made,mainly,make,makes,many,may,maybe,me,mean,means,meantime,meanwhile,merely,mg,might,million,miss,ml,more,moreover,most,mostly,mr,mrs,much,mug,must,my,myself,n,na,name,namely,nay,nd,near,nearly,necessarily,necessary,need,needs,neither,never,nevertheless,new,next,nine,ninety,no,nobody,non,none,nonetheless,noone,nor,normally,nos,not,noted,nothing,now,nowhere,o,obtain,obtained,obviously,of,off,often,oh,ok,okay,old,omitted,on,once,one,ones,only,onto,or,ord,other,others,otherwise,ought,our,ours,ourselves,out,outside,over,overall,owing,own,p,page,pages,part,particular,particularly,past,per,perhaps,placed,please,plus,poorly,possible,possibly,potentially,pp,predominantly,present,previously,primarily,probably,promptly,proud,provides,put,q,que,quickly,quite,qv,r,ran,rather,rd,re,readily,really,recent,recently,ref,refs,regarding,regardless,regards,related,relatively,research,respectively,resulted,resulting,results,right,run,s,said,same,saw,say,saying,says,sec,section,see,seeing,seem,seemed,seeming,seems,seen,self,selves,sent,seven,several,shall,she,shed,she'll,shes,should,shouldn't,show,showed,shown,showns,shows,significant,significantly,similar,similarly,since,six,slightly,so,some,somebody,somehow,someone,somethan,something,sometime,sometimes,somewhat,somewhere,soon,sorry,specifically,specified,specify,specifying,still,stop,strongly,sub,substantially,successfully,such,sufficiently,suggest,sup,sure

t,take,taken,taking,tell,tends,th,than,thank,thanks,thanx,that,that'll,thats,that've,the,their,theirs,them,themselves,then,thence,there,thereafter,thereby,thered,therefore,therein,there'll,thereof,therere,theres,thereto,thereupon,there've,these,they,theyd,they'll,theyre,they've,think,this,those,thou,though,thoughh,thousand,throug,through,throughout,thru,thus,til,tip,to,together,too,took,toward,towards,tried,tries,truly,try,trying,ts,twice,two,u,un,under,unfortunately,unless,unlike,unlikely,until,unto,up,upon,ups,us,use,used,useful,usefully,usefulness,uses,using,usually,v,value,various,'ve,very,via,viz,vol,vols,vs,w,want,wants,was,wasnt,way,we,wed,welcome,we'll,went,were,werent,we've,what,whatever,what'll,whats,when,whence,whenever,where,whereafter,whereas,whereby,wherein,wheres,whereupon,wherever,whether,which,while,whim,whither,who,whod,whoever,whole,who'll,whom,whomever,whos,whose,why,widely,willing,wish,with,within,without,wont,words,world,would,wouldnt,www,x,y,yes,yet,you,youd,you'll,your,youre,yours,yourself,yourselves,you've,z,zero,";

```php
if ($start == 0) {
    // liked paragraphs for this question
    $sql0 = "select ref, count(ref) from egwquestionwritinglikelog where
question = '$s' group by ref order by count desc;";
    if ($debug) echo "<Font color=white>$sql0</font><br>";
    $res0 = dbexec ($sql0);
    $valids = "$s";
    $nrows = pg_numrows ($res0);
    if ($nrows > 1) {
        echo "User suggested results";
    }
```

```php
                for ($ii = 0; $ii < min(5, pg_numrows ($res0)); $ii++) {
                        $ref = pg_result ($res0, $ii, "ref");
                        $count = pg_result ($res0, $ii, "count");
                        $sql01 = "select * from egwwritings where reference = '$ref';";
                        $res01 = dbexec ($sql01);
                        echo "<blockquote>";
                        $text = pg_result ($res01, 0, "contents");
                        $tok = strtok ($valids, $delims);
                        while ($tok !== false) {
                                //if (!stristr($stopwords, ",$tok,")) {
                                        $text = preg_replace("/\b($tok)\b/i", "<b>$1</b>",
$text);
                                        $tok = strtok ($delims);
                                //}
                        }
                        $ref2 = str_replace (" ", ".", $ref);
                        $text = str_replace ("$ref", "<a target=_egw
href=https://egwwritings.org/?ref=en_".urlencode($ref2).">$ref</a>", $text);
                        echo $text;
                        echo " <a onclick=\"submitquestionwritinglike('$s', '$ref')\"><img
src=like.png></a>";
                        if ($count == 1)
                                echo " <Font color=lightgray size=1>($count
Like)</font>";
                        else
                                echo " <Font color=lightgray size=1>($count
Likes)</font>";
                        echo "</blockquote>";

                }
        }

        $sqlb = $sqlub = $sqlxb = $sql = "";


        //$sql .= "select topicid, /*10 as prority,*/ topic, body from egwtopicalindex
where topic = '$s' ";
    //$sql .= "select reference, sum(count) from (select word, reference,
count(reference) from egwindex where FALSE group by word, reference ";

    $tok = strtok ($s, $delims);
        $sql = "";
        //$sql .= "select ref, parlen, sum, sum * 1/parlen as score from (";
        $sql .= "select ref, parlen, sum(count) as sum from (";
        $sqlb .= $sql;
        $sqlx = "(";
        $sqlu = "(";
        $sqlbx = "(";
        $sqlbu = "(";
    $wordsb = 0;
```

```php
        $vtoksb = 0;
        $validsb = "";
    $words = 0;
        $vtoks = 0;
        $valids = "";
    while ($tok !== false) {
                $words++;
                $wordsb++;
                $stok = pg_escape_string (strtolower($tok));
                $tres = dbexec ("select 1 from craw_$stok;", "egwindex3");
                if ($tres) {
                        if (!stristr($stopwords, ",$tok,")) {
                                $valids .= " $tok ";
                                $vtoks++;
                                if ($vtoks > 1) {
                                        $sqlu .= " UNION ALL ";
                                        $sqlx .= " INTERSECT ALL ";
                                }
                                $sqlu .= " select ref, parlen, 1 /*count*/ as count from
craw_$stok /*group by ref, parlen*/ ";
                                $sqlx .= " select ref, parlen, 1 /*count*/ * 20.01 as count
from craw_$stok /*group by ref, parlen*/ ";
                        }
                        $validsb .= " $tok ";
                        $vtoksb++;
                        if ($vtoksb > 1) {
                                $sqlbu .= " UNION ALL ";
                                $sqlbx .= " INTERSECT ALL ";
                        }
                        $sqlbu .= " select ref, parlen, 1 /*count*/ *1 as count from
craw_$stok /*group by ref, parlen*/ ";
                        $sqlbx .= " select ref, parlen, 1 /*count*/ *20.01 as count from
craw_$stok /*group by ref, parlen*/ ";
                }
        $tok = strtok ($delims);
    }
        $sqlu .= ")";
        $sqlx .= ")";
        $sql .= " ($sqlx) UNION ($sqlu) ) as foo group by ref, parlen order by sum
desc, parlen asc ";
        $sqlbu .= ")";
        $sqlbx .= ")";
        $sqlb .= " ($sqlbx) UNION ($sqlbu) ) as foo group by /*ref, parlen*/ order by
sum desc, parlen asc ";
        //$sql .= ") as bar order by score desc ";
        if ($debug) echo "<font color=white>$sql</font>";
        $includedstopwords = 0;
        $sratio = (strlen ($valids)+1)/(1+strlen($validsb));
        if ($debug) echo "SRATIO:$sratio ";
        if ($sratio > 0.1) {
```

```php
                $res = dbexec ($sql, "egwindex3");
                $includedstopwords = 0;
        } else {
                $valids = $validsb;
                $words = $wordsb;
                $sql = $sqlb;
                $vtoksb = $vtoks;
                if ($debug) echo $sql;
                $res = dbexec ($sql, "egwindex3");
        }
        if (!$includedstopwords && pg_numrows ($res) < 1) {
                if ($debug) echo "<font color=white>0 results without stopwords. trying
with stopwords.</font>";
                $valids = $validsb;
                $words = $wordsb;
                $sql = $sqlb;
                $vtoksb = $vtoks;
                if ($debug) echo $sql;
                $res = dbexec ($sql, "egwindex3");
        }
        if ($res)
                $nresults = pg_numrows ($res);
        else
                $nresults = 0;
        dbexec ("insert into egwsearchlog (url, nresults, datetime) values
('".pg_escape_string($_SERVER['REQUEST_URI'])."','$nresults','now'::timestamp);")
;
        echo "<br>".min($start,$nresults)." - ".min($start+$limit,$nresults)." of
$nresults results in ";
        echo sprintf ("%.2f", (microtime(true) - $mt))." seconds paragraph index
search<br>";
        if ($debug) echo "$s -> $valids<br>";

        ?>
<script>
        getText = function (url, callback)
        {
                var request = new XMLHttpRequest();
                request.onreadystatechange = function()
                {
                        if (request.readyState == 4 && request.status == 200)
                        {
                                callback(request.responseText);
                        }
                };
                request.open('GET', url);
                request.send();
        }
        function submitquestionwritinglike (q, ref) {
                function mycallback(data) {
```

```php
                            alert (data);
                    }
                    getText ('egwwritingquestionlikelog.php?q='+q+'&ref='+ref,
mycallback);
            }
        function submitquestiontopiclike (q, topic) {
                    function mycallback(data) {
                            alert (data);
                    }
                    getText ('egwtopicquestionlikelog.php?q='+q+'&topic='+topic,
mycallback);
            }
</script>
<?php
        echo "<blockquote>";
        for ($i = max(0, $start); $i < min($start+$limit,pg_numrows ($res)); $i++) {
                $ref = pg_result ($res, $i, "ref");

                $resq = dbexec("select * from egwwritings where reference = '$ref'");
                for ($qi = 0; $qi < pg_numrows ($resq); $qi++) {
                        $text = pg_result ($resq, $qi, "contents");
                        //echo $s;
                        $tok = strtok ($valids, $delims);
                        while ($tok !== false) {
                                //if (!stristr($stopwords, ",$tok,")) {
                                        $text = preg_replace("/\b($tok)\b/i", "<b>$1</b>",
$text);

                                        $tok = strtok ($delims);
                                //}
                        }
                        $ref2 = str_replace (" ", ".", $ref);
                        $text = str_replace ("$ref", "<a target=_egw
href=https://egwwritings.org/?ref=en_".urlencode($ref2).">$ref</a>", $text);
                        echo $text;
                        echo " <a onclick=\"submitquestionwritinglike('$s', '$ref')\"><img
src=like.png></a>";
                }
                echo "<font color=white>$ref";
                echo " ".pg_result ($res, $i, "sum")."pts</font><br>";
                flush (); ob_flush();
                echo "<br>";
        }
        echo "".sprintf ("%.2f", (microtime(true) - $mt))."secs retrieving writings.";
        echo "<br><a
href=".$_SERVER["PHP_SELF"]."?start=".($start+$limit)."&s=".urlencode($s).">more
results from writings</a>";
        echo "</blockquote>";
?>
<form>
Submit a reference for this question / search <input type=text id=rref>
```

```php
<input type=button value=Submit onclick="submitquestionwritinglike('<?php echo $s;
?>', rref.value)"> | <a href=<?php echo $_SERVER["PHP_SELF"]; ?>
target=_egwhelp>Open another search tab</a>
```

egwtopicquestionlikelog.php
```php
<?php
        include "dbfuncs.php";
        $q = @$_GET['q'];
        $topic = pg_escape_string(@$_GET['topic']);
        $res = dbexec ("select entry from egwtopicalindexflat where entry = '$topic'");
        if (pg_numrows ($res) > 0) {
                $sql = "insert into egwquestiontopiclikelog (question, topic, datetime)
values ('".pg_escape_string ($q)."', '".pg_escape_string ($topic)."',
'now'::timestamp);";
                dbexec ($sql);
                echo "Thank you!";
        } else {
                echo "Could not find this topic...";
        }
```

egwwritingquestionlikelog.php
```php
<?php
        include "dbfuncs.php";
        $q = @$_GET['q'];
        $ref = @$_GET['ref'];
        $ref = str_replace ("{", "", $ref);
        $ref = str_replace ("}", "", $ref);
        $res = dbexec ("select reference from egwwritings where reference = '$ref'");
        if (pg_numrows ($res) > 0) {
                $sql = "insert into egwquestionwritinglikelog (question, ref, datetime)
values ('".pg_escape_string ($q)."', '".pg_escape_string ($ref)."', 'now'::timestamp);";
                //echo $sql;
                dbexec ($sql);
                echo "Thank you!";
        } else {
                echo "could not find reference in egwwritings...";
}
```

seedquestions.php
```php
<?php
include 'dbfuncs.php';

$byes = 0;
$bno = 0;
$yes = $no = 0;

$sql = "select topic, count(votes) from bibletopicsvotes group by topic order by count
desc;";
$res = dbexec ($sql);
```

```php
$nrows = pg_numrows ($res);
for ($i = 0; $i < min(9999,$nrows); $i++) {
        $topic = pg_result ($res, $i, "topic");
        $votes = pg_result ($res, $i, "count");

        $totalvotes = 0;
        $sqlt = "select * from egwquestiontopiclikelog where question = '$topic';";
        $rest = dbexec ($sqlt);
        if ($rest) {
                $tvotes = pg_numrows ($rest);
                echo "$tvotes topic likes. ";
                $totalvotes += $tvotes;
        }

        $sqlt = "select * from egwquestionwritinglikelog where question = '$topic';";
        $rest = dbexec ($sqlt);
        if ($rest) {
                $wvotes = pg_numrows ($rest);
                echo "$wvotes writing likes. ";
                $totalvotes += $wvotes;
        }

        $sqlt = "select * from egwtopicalindexflat where lower(entry) like
lower('$topic%');";
        $rest = dbexec ($sqlt);
        if ($rest) {
                $wvotes = pg_numrows ($rest);
                echo "$wvotes topical hits. ";
                $totalvotes += $wvotes;
                if ($wvotes)
                        $byes++;
                else
                        $bno++;
        }

        if ($totalvotes == 0) {
                $no++;
                echo "<font color=red><b>";
        } else {
                $yes++;
        }
        echo "$i. yes:$yes no:$no byes:$byes bno:$bno <a target=_egw
href=egwmain1.php?s=".urlencode($topic).">$topic</a> $votes ";
        if ($totalvotes == 0) echo "</font></b>";

        echo "<br>\n";
//      flush (); ob_flush();

}
```

parseegwtopics-rtf-to-txt1.php

```php
<?php
$file = file_get_contents ("egw-index-all.rtf");
$file = str_replace ("\\pard", "\\par", $file);
$lines = explode ("\\par",$file);
$topicdelim = "\\s2 {\\v\\f0\\fs28\\cf2";
$topicdelimlen = strlen ($topicdelim);
foreach ($lines as $line) {
        $i = strstr ($line, $topicdelim);
        $topic = substr ($i, $topicdelimlen + 1);
        if (strlen($topic) > 0) {
                echo "\nTOPIC: $topic";
        } else {
                $style = 0;
                $line = str_replace ("\\*\\bkmkstart ", "", $line);
                $line = str_replace ("\\*\\bkmkend ", "", $line);
                //$line = str_replace ("\n", "", $line);
                if (strstr ($line, "\\s5 ")) $style = 5;
                if (strstr ($line, "\\s6 ")) $style = 6;

                $len = strlen ($line);
                for ($i = 0; $i < $len; $i++) {
                        switch ($line[$i]) {
                                case "\\":
                                case "{":
                                case "}":
                                        $skip = 1;
                                        break;
                                case " ":
                                        $skip = 0;
                                        if (@$line[$i+1]==' ') $i++;
                                        break;
                        }
                        if (!$skip) {
                                if ($style == 6)
                                        echo " ";
                                echo $line[$i];
                                $style = 0;
                        }
                }
        }
}
?>
```


parseegwtopics-txt1-to-db.php

```php
<?php
include "dbfuncs.php";

dbexec ("delete from egwtopicalindex;");
```

```php
$file = file_get_contents ("egw-index-all-1.txt");
$topics = explode ("TOPIC: ",$file);
$n = 0;
foreach ($topics as $topic) {
        $n++;
        $lines = explode ("\n", $topic);
        $topictitle = str_replace ("\n", "", $lines[0]);
        $topictitle = str_replace ("\r", "", $topictitle);
        //echo "TOPIC:$topictitle\n";
        $topicbody = "";
        for ($i = 1; $i < sizeof($lines); $i++) {
                $line = str_replace ("\n", "", $lines[$i]);
                $line = str_replace ("\r", "", $line);
                //echo "++++line:".$line."***\n";
                //echo "===".$lines[$i+1][0].$lines[$i+1][1]."----\n";

                if (@$line[0] == ' ' && @$line[1] == ' ')
                        $topicbody .= "\n".$line;
                else
                        $topicbody .= $line;
        }
        $topicbody = trim ($topicbody);
        //echo "BODY: $topicbody\n";
        $sql = "insert into egwtopicalindex (topic, body) values
('".pg_escape_string($topictitle)."', '".pg_escape_string($topicbody)."');";
        echo "$n\n";
        //echo $sql."\n";
        dbexec ($sql);
}
?>
```

parseegwtopics-db-to-dbindex.php
```php
<?php
include "dbfuncs.php";

$debug = 0;

$booksregex =
'(AA|AH|ApM|AY|CME|CSA|CG|CTr|COL|CE|CET|ChL|ChS|CTBH|CEv|CM|CC|Con|
CCh|CD|CH|CSW|CS|CT|CW|CL|DD|DG|DA|EW|Ed|Ev|FW|FLB|FE|AG|GW92|GW|
GC88|GC|Hvn|HFM|HL|HDL|HS|HP|LDE|LYL|LS80|LS88|LS|LHU|MC|Mar|MM|MYP|
1MCP|2MCP|MH|ML|NL|OFC|OHC|PaM|PP|PCP|Pr|PK|PM|RC|RY|SL|1SM|2SM|3S
M|1BC|2BC|3BC|4BC|5BC|6BC|7BC|7ABC|ExV|LP|SA|SD|SW|SpTEd|1SP|2SP|3SP
|4SP|1SG|2SG|3SG|4aSG|4bSG|SC|SJ|SR|ExV54|Te|1T|2T|3T|4T|5T|6T|7T|8T|9T|
TSS|TSB|TM|TSA|TSDF|TMK|TDG|MB|BLJ|TA|UL|VSS|WM|WLF|YRP)';

$stopwords =
pg_escape_string(",a,able,about,above,abst,accordance,according,accordingly,acro
```

ss,act,actually,added,adj,affected,affecting,affects,after,afterwards,again,against,ah,all,almost,alone,along,already,also,although,always,am,among,amongst,an,and,announce,another,any,anybody,anyhow,anymore,anyone,anything,anyway,anyways,anywhere,apparently,approximately,are,aren,arent,arise,around,as,aside,ask,asking,at,auth,available,away,awfully,b,back,be,became,because,become,becomes,becoming,been,before,beforehand,begin,beginning,beginnings,begins,behind,being,believe,below,beside,besides,between,beyond,biol,both,brief,briefly,but,by,c,ca,came,can,cannot,can't,cause,causes,certain,certainly,co,com,come,comes,contain,containing,contains,could,couldnt,d,date,did,didn't,different,do,does,doesn't,doing,done,don't,down,downwards,due,during,e,each,ed,edu,effect,eg,eight,eighty,either,else,elsewhere,end,ending,enough,especially,et,et-al,etc,even,ever,every,everybody,everyone,everything,everywhere,ex,except,f,far,few,ff,fifth,first,five,fix,followed,following,follows,for,former,formerly,forth,found,four,from,further,furthermore,g,gave,get,gets,getting,give,given,gives,giving,go,goes,gone,got,gotten,h,had,happens,hardly,has,hasn't,have,haven't,having,he,hed,hence,her,here,hereafter,hereby,herein,heres,hereupon,hers,herself,hes,hi,hid,him,himself,his,hither,home,how,howbeit,however,hundred,i,id,ie,if,i'll,im,immediate,immediately,importance,important,in,inc,indeed,index,information,instead,into,invention,inward,is,isn't,it,itd,it'll,its,itself,i've,j,just,k,keep

keeps,kept,kg,km,know,known,knows,l,largely,last,lately,later,latter,latterly,least,less,lest,let,lets,like,liked,likely,line,little,'ll,look,looking,looks,ltd,m,made,mainly,make,makes,many,may,maybe,me,mean,means,meantime,meanwhile,merely,mg,might,million,miss,ml,more,moreover,most,mostly,mr,mrs,much,mug,must,my,myself,n,na,name,namely,nay,nd,near,nearly,necessarily,necessary,need,needs,neither,never,nevertheless,new,next,nine,ninety,no,nobody,non,none,nonetheless,noone,nor,normally,nos,not,noted,nothing,now,nowhere,o,obtain,obtained,obviously,of,off,often,oh,ok,okay,old,omitted,on,once,one,ones,only,onto,or,ord,other,others,otherwise,ought,our,ours,ourselves,out,outside,over,overall,owing,own,p,page,pages,part,particular,particularly,past,per,perhaps,placed,please,plus,poorly,possible,possibly,potentially,pp,predominantly,present,previously,primarily,probably,promptly,proud,provides,put,q,que,quickly,quite,qv,r,ran,rather,rd,re,readily,really,recent,recently,ref,refs,regarding,regardless,regards,related,relatively,research,respectively,resulted,resulting,results,right,run,s,said,same,saw,say,saying,says,sec,section,see,seeing,seem,seemed,seeming,seems,seen,self,selves,sent,seven,several,shall,she,shed,she'll,shes,should,shouldn't,show,showed,shown,showns,shows,significant,significantly,similar,similarly,since,six,slightly,so,some,somebody,somehow,someone,somethan,something,sometime,sometimes,somewhat,somewhere,soon,sorry,specifically,specified,specify,specifying,still,stop,strongly,sub,substantially,successfully,such,sufficiently,suggest,sup,sure

t,take,taken,taking,tell,tends,th,than,thank,thanks,thanx,that,that'll,thats,that've,the,their,theirs,them,themselves,then,thence,there,thereafter,thereby,thered,therefore,therein,there'll,thereof,therere,theres,thereto,thereupon,there've,these,they,theyd,they'll,theyre,they've,think,this,those,thou,though,thoughh,thousand,throug,through,throughout,thru,thus,til,tip,to,together,too,took,toward,towards,tried,tries,truly,try,trying,ts,twice,two,u,un,under,unfortunately,unless,unlike,unlikely,until,unto,up,upon,ups,us,use,used,useful,usefully,usefulness,uses,using,usually,v,value,various,'ve,very,via,viz,vol,vols,vs,w,want,wants,was,wasnt,way,we,wed,welcome,we'll,went,were,werent,we've,what,whatever,what'll,whats,when,whence,whenever,where,whereafter,whereas,whereby,wherein,wheres,whereupon,wherever,whether,which,while,whim,whither,who,whod,whoever,whole,who'll,whom,whomever,whos,whose,why,widely,willing,wi

sh,with,within,without,wont,words,world,would,wouldnt,www,x,y,yes,yet,you,youd,you'll,your,youre,yours,yourself,yourselves,you've,z,zero,");

```php
dbexec ("drop table egwtopics;");
dbexec ("create table egwtopics (topicid serial, topic text);");

dbexec ("drop table egwtopicalindexflat;");
dbexec ("create table egwtopicalindexflat (topicid serial, entry text);");

dbexec ("drop database egwtopicalindexindex;");
dbexec ("create database egwtopicalindexindex;");

$sql = "select * from egwtopicalindex where  topic = 'Sabbath'  order by topicid;";
echo $sql;
$res = dbexec ($sql);
$max = pg_numrows ($res);
for ($i = 0; $i < $max; $i++) {
        $topic1 = pg_result ($res, $i, "topic");
        echo "\ntopic: \"$topic1\"";
        $sqlt = "insert into egwtopics (topic) values ('".pg_escape_string($topic1)."');";
        if ($debug) echo "\n$sqlt";
        dbexec ($sqlt);
        $topicid = pg_result ($tres, 0, 0);
        $body = pg_result ($res, $i, "body");
        $abody = explode ("\n", $body);
        $topic2 = $topic3 = $topic4 = $topic5 = "";
        foreach ($abody as $line) {
                echo "\n-";
                $line = rtrim ($line);
                $line = preg_replace ('/([0-9]+\. )/', '', $line);
                if (!strncmp ($line, "    ", 4) && $line[4] != " ") {
                        $topic2 = trim ($line);
                        $topic3 = $topic4 = $topic5 = "";
                }
                if (!strncmp ($line, "     ", 5) && $line[5] != " ") {
                        $topic3 = trim ($line);
                        $topic4 = $topic5 = "";
                }
                if (!strncmp ($line, "      ", 6) && $line[6] != " ") {
                        $topic4 = trim ($line);
                        $topic5 = "";
                }
                if (!strncmp ($line, "       ", 7) && $line[7] != " ") {
                        $topic5 = trim ($line);
                }
                if (/*$topic2 != trim($line) && */ctype_digit(substr ($line, -1))) {
                        //if ($debug) echo
"\nT1:$topic1|T2:$topic2|T3:$topic3|T4:$topic4|T5:$topic5|L:$line|";
                        $entry = "$topic1";
                        if ($topic1 != $topic2 && strlen ($topic2)) {
```

```php
                $entry .= ", $topic2";
        }
        if ($topic2 != $topic3 && strlen ($topic3)) {
                $entry .= ", $topic3";
        }
        if ($topic3 != $topic2 && strlen ($topic4)) {
                $entry .= ", $topic4";
        }
        if ($topic4 != $topic2 && strlen ($topic5)) {
                $entry .= ", $topic5";
        }
        $entry = preg_replace ('/(,)+/',',', $entry);
        if ($debug) echo "\nENTRY: '$entry'\n";
        $tres = dbexec ("insert into egwtopicalindexflat (entry) values
('".pg_escape_string ($entry)."') returning topicid;");
        $topicid = pg_result ($tres, 0, 0);
        $body = $entry;

        // extract references
        // replaces the commas with books - winwin :D God helped me
to do this April 27, 2020
        $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
        $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
        $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
        $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
        $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
        $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
        $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);
        $body = preg_replace ('/'.$booksregex.'( [0-9]+)(:|-)?([0-9])+,( [0-
9]+)(:|-)?([0-9]+)+/', '${1}${2}${3}${4}; ${1}${5}${6}${7}', $body);

        // replace references with links - winwin :D God helped me to do
this April 27, 2020 - orig regex
        // extract references actually
        preg_match_all ('/'.$booksregex.' ([0-9]+[:-]?[0-9]+)+/', $body,
$refs);

        if ($debug) print_r ($refs[0]);

        $string = str_replace ("\r\n", " ", $body);
        $string = trim($string); // trim the string
        $string = preg_replace('/[^a-zA-Z0-9\' -]/', '', $string); // only take
alphanumerical characters, but keep the spaces and dashes too…
```

```php
                              $string = preg_replace ('/'.$booksregex.' ([0-9]+)(:|-)?([0-9]+)+/',
'', $string);
                              if ($debug) echo "\n$string";
                              $parlen = strlen ($string);

                              preg_match_all('/\b.*?\b/i', $string, $matchWords);
//                            print_r ($matchWords[0]);

                              $matchWords = $matchWords[0];
                              foreach ($matchWords as $key=>$item) {
                                      echo ",";
                                      $itemoc = $item = pg_escape_string(trim($item));
                                      $item = strtolower($item);
                                      if (strlen ($item) > 0              // skip blanks -- not
anymore - and single letters
                                              && ctype_alnum ($item)           // skip numbers
                                              && !is_numeric ($item)           // skip numbers
                                              && !ctype_upper($itemoc)         // skip EGW
estate comments
                                              //&& !strstr($stopwords, ",$item,") // skip stopwords
                                      ) {
                                              if ($debug) echo "\n==$item";
                                              foreach ($refs[0] as $ref) {
                                                      echo ".";
                                                      if ($debug) echo "\n$ref; ";
                                                      $sql1 = "create table \"raw_$item\" (ref text,
topicid int, parlen int);";
                                                      if ($debug) echo "\n$sql1";
                                                      $tr1 = @dbexecmany ($sql1,
"egwtopicalindexindex");
                                                      @dbexec ("drop table \"raw_$item\";");
                                                      //pg_free_result ($tr1);
                                                      $sql2 = "insert into \"raw_$item\" (ref,
topicid, parlen) values ('$ref', '$topicid', '$parlen');";
                                                      if ($debug) echo "\n$sql2";
                                                      $tr2 = @dbexecmany($sql2,
"egwtopicalindexindex");
                                                      //pg_free_result ($tr2);
                                              }
                                      }
                              }
                      }
              }
}


parseegwcomptxt-to-db.php
<?php
ini_set('memory_limit','999M');
include "dbfuncs.php";
```

```php
exit (0); // this is the old index

$redowritings = 0;
$redoindex = 0;
if ($redowritings) {
        echo "deleteing from egwwritings;..\n";
        dbexec ("delete from egwwritings;");
}

if ($redoindex) {
        echo "deleteing from egwindex;..\n";
        dbexec ("delete from egwindex;");
}

echo "reading file...\n";
$file = file_get_contents ("egw-comp-exported-genetic-txt-uniq-book-references-
utf8.txt"); // whole file
//$file = file_get_contents ("egw-comp-exported-genetic-txt-uniq-book-references-
utf8.txt", 2000); // limited - for quick testing
$pars = explode ("}",$file); // separate paragraphs
$n = 0;
foreach ($pars as $par) {
        $n++;
        //if ($n > 10) break; // for speed testing

        echo "paragraph $n\n";
        // extract paragraph and reference
        $lastopen = strrpos ($par, "{") + 1;
        $ref = pg_escape_string(substr($par, $lastopen));
        $par .= "}";
        if ($redowritings) {
                $sql = "insert into egwwritings (reference, contents) values ('".$ref."',
'".pg_escape_string($par)."');";
        //      echo $sql."\n";
                dbexec ($sql);
        }

        if ($redoindex
                //&& !stristr($par, "table of contents")
        ) {
                //$string = preg_replace('/ss+/i', '', $par); // ewan. this replaces all ss+
with nothing
                $string = str_replace ("\r\n", " ", $string);
        //      echo "----------------------\n$string\n-----------------------\n";
                $string = trim($string); // trim the string
                $string = preg_replace('/[^a-zA-Z0-9 -]/', '', $string); // only take
alphanumerical characters, but keep the spaces and dashes too…
                $string = strtolower($string); // make it lowercase

                preg_match_all('/\b.*?\b/i', $string, $matchWords);
```

```php
            $matchWords = $matchWords[0];
            foreach ($matchWords as $key=>$item) {
                $item = trim($item);
                if (strlen ($item) > 1                    // skip blanks and single
letters
                    && !is_numeric ($item)                // skip numbers
                    //&& !ctype_upper($item)              // skip EGW estate
comments
                ) {
                    //echo "$key => $item\n";
                    $sql = "insert into egwindex (word, reference) values
('$item', '$ref')";
                    //             echo "par:$n\t".$sql."\n";
                    dbexec ($sql);
                }
            }
        }
}

if ($redowritings) {
    echo "creating index reference to egwwritings;..\n";
    dbexec ("create index idx_egwwritings_reference on egwwritings
(reference);");
}

?>
```

parseegwcomptxt-to-db2.php

```php
<?php
ini_set('memory_limit','999M');
include "dbfuncs.php";

$redowritings = 0;
$redoindex = 0;
$redodb2 = 1;
if ($redodb2) {
    echo "recreateding egwindex2 db;..\n";
    $res = dbexec ("drop database egwindex2;");
    if (!$res) die ("cannot drop egwindex2 database;");
    $res = dbexec ("create database egwindex2;");
    if (!$res) die ("cannot create egwindex2 database;");
}


echo "reading file...\n";
$file = file_get_contents ("egw-comp-exported-genetic-txt-uniq-book-references-
utf8.txt"); // whole file
//$file = file_get_contents ("egw-comp-exported-genetic-txt-uniq-book-references-
utf8.txt", 2000); // limited - for quick testing
```

```
$pars = explode ("}",$file); // separate paragraphs
$n = 0;
$stopwords =
pg_escape_string(",a,able,about,above,abst,accordance,according,accordingly,acro
ss,act,actually,added,adj,affected,affecting,affects,after,afterwards,again,against,ah,
all,almost,alone,along,already,also,although,always,am,among,amongst,an,and,ann
ounce,another,any,anybody,anyhow,anymore,anyone,anything,anyway,anyways,any
where,apparently,approximately,are,aren,arent,arise,around,as,aside,ask,asking,at,a
uth,available,away,awfully,b,back,be,became,because,become,becomes,becoming,
been,before,beforehand,begin,beginning,beginnings,begins,behind,being,believe,bel
ow,beside,besides,between,beyond,biol,both,brief,briefly,but,by,c,ca,came,can,cann
ot,can't,cause,causes,certain,certainly,co,com,come,comes,contain,containing,contai
ns,could,couldnt,d,date,did,didn't,different,do,does,doesn't,doing,done,don't,down,do
wnwards,due,during,e,each,ed,edu,effect,eg,eight,eighty,either,else,elsewhere,end,e
nding,enough,especially,et,et-
al,etc,even,ever,every,everybody,everyone,everything,everywhere,ex,except,f,far,fe
w,ff,fifth,first,five,fix,followed,following,follows,for,former,formerly,forth,found,four,fro
m,further,furthermore,g,gave,get,gets,getting,give,given,gives,giving,go,goes,gone,g
ot,gotten,h,had,happens,hardly,has,hasn't,have,haven't,having,he,hed,hence,her,her
e,hereafter,hereby,herein,heres,hereupon,hers,herself,hes,hi,hid,him,himself,his,hith
er,home,how,howbeit,however,hundred,i,id,ie,if,i'll,im,immediate,immediately,importa
nce,important,in,inc,indeed,index,information,instead,into,invention,inward,is,isn't,it,it
d,it'll,its,itself,i've,j,just,k,keep
        keeps,kept,kg,km,know,known,knows,l,largely,last,lately,later,latter,latterly,lea
st,less,lest,let,lets,like,liked,likely,line,little,'ll,look,looking,looks,ltd,m,made,mainly,ma
ke,makes,many,may,maybe,me,mean,means,meantime,meanwhile,merely,mg,might
,million,miss,ml,more,moreover,most,mostly,mr,mrs,much,mug,must,my,myself,n,na,
name,namely,nay,nd,near,nearly,necessarily,necessary,need,needs,neither,never,n
evertheless,new,next,nine,ninety,no,nobody,non,none,nonetheless,noone,nor,norma
lly,nos,not,noted,nothing,now,nowhere,o,obtain,obtained,obviously,of,off,often,oh,ok,
okay,old,omitted,on,once,one,ones,only,onto,or,ord,other,others,otherwise,ought,our
,ours,ourselves,out,outside,over,overall,owing,own,p,page,pages,part,particular,parti
cularly,past,per,perhaps,placed,please,plus,poorly,possible,possibly,potentially,pp,pr
edominantly,present,previously,primarily,probably,promptly,proud,provides,put,q,que
,quickly,quite,qv,r,ran,rather,rd,re,readily,really,recent,recently,ref,refs,regarding,rega
rdless,regards,related,relatively,research,respectively,resulted,resulting,results,right,r
un,s,said,same,saw,say,saying,says,sec,section,see,seeing,seem,seemed,seeming,
seems,seen,self,selves,sent,seven,several,shall,she,shed,she'll,shes,should,shouldn
't,show,showed,shown,showns,shows,significant,significantly,similar,similarly,since,si
x,slightly,so,some,somebody,somehow,someone,somethan,something,sometime,so
metimes,somewhat,somewhere,soon,sorry,specifically,specified,specify,specifying,st
ill,stop,strongly,sub,substantially,successfully,such,sufficiently,suggest,sup,sure
        t,take,taken,taking,tell,tends,th,than,thank,thanks,thanx,that,that'll,thats,that've
,the,their,theirs,them,themselves,then,thence,there,thereafter,thereby,thered,therefor
e,therein,there'll,thereof,therere,theres,thereto,thereupon,there've,these,they,theyd,t
hey'll,theyre,they've,think,this,those,thou,though,thoughh,thousand,throug,through,th
roughout,thru,thus,til,tip,to,together,too,took,toward,towards,tried,tries,truly,try,trying,
ts,twice,two,u,un,under,unfortunately,unless,unlike,unlikely,until,unto,up,upon,ups,us
,use,used,useful,usefully,usefulness,uses,using,usually,v,value,various,'ve,very,via,v
iz,vol,vols,vs,w,want,wants,was,wasnt,way,we,wed,welcome,we'll,went,were,werent,
```

we've,what,whatever,what'll,whats,when,whence,whenever,where,whereafter,where
as,whereby,wherein,wheres,whereupon,wherever,whether,which,while,whim,whither
,who,whod,whoever,whole,who'll,whom,whomever,whos,whose,why,widely,willing,wi
sh,with,within,without,wont,words,world,would,wouldnt,www,x,y,yes,yet,you,youd,yo
u'll,your,youre,yours,yourself,yourselves,you've,z,zero,");

```php
$ot = $t = time();
$ln = 0;
$speed = 0;
$tpars = 250582;
foreach ($pars as $par) {
        $n++;
        $nt = time();
        if ($nt > $t) {
                $t = $nt;
                $speed = ($n-$ln);
                $ln = $n;
                echo "$speed/sec paragraph $n/$tpars ".($n/$tpars*100)."%
est.".($tpars-$n/($n/($nt-$ot))/60)."mins remaining\n";
        }
        //if ($n > 10) break; // for speed testing

        //echo "$speed/sec paragraph $n/$tpars ".($n/$tpars*100)."%
est.".($tpars/($n/($nt-$ot))/60)."mins remaining\n";
        // extract paragraph and reference
        $lastopen = strrpos ($par, "{") + 1;
        $ref = pg_escape_string(substr($par, $lastopen));
        $par .= "}";
        $parlen = strlen ($par);
        if ($redowritings) {
                $sql = "insert into egwwritings (reference, contents, length) values
('".$ref."', '".pg_escape_string($par)."','$parlen');";
//        echo $sql."\n";
                dbexec ($sql);
        }

        if (stristr("$par","table of contents")) {
                continue;
        }

        if ($redoindex || $redodb2
                //&& !stristr($par, "table of contents")
        ) {
                $string = $par;
                //$string = preg_replace('/ss+/i', '', $par); // ewan. i think this replaces all
ss+ with nothing
                $string = str_replace ("\r\n", " ", $string);
//        echo "---------------------\n$string\n-----------------------\n";
                $string = trim($string); // trim the string
                $string = preg_replace('/[^a-zA-Z0-9\' -]/', '', $string); // only take
alphanumerical characters, but keep the spaces and dashes too…
```

```php
                    preg_match_all('/\b.*?\b/i', $string, $matchWords);
                    $matchWords = $matchWords[0];
                    foreach ($matchWords as $key=>$item) {
                            $itemoc = $item = pg_escape_string(trim($item));
                            $item = strtolower($item);
                            if (strlen ($item) > 0                    // skip blanks -- not anymore
and single letters
                                    && ctype_alnum ($item)           // skip numbers
                                    && !is_numeric ($item)           // skip numbers
                                    && !ctype_upper($itemoc)         // skip EGW estate
comments
                                    //&& !strstr($stopwords, ",$item,") // skip stopwords
                            ) {
                                    //echo "$key => $item\n";
                                    //$sql = "insert into egwindex (word, reference) values
('$item', '$ref')";
                                    //      echo "par:$n\t".$sql."\n";
                                    //$sql1 = "create table if not exists \"raw_$item\" (ref
text);";
                                    $sql1 = "create table \"raw_$item\" (ref text, parlen int);";
                                    //$tr1 = @dbexec ($sql1, "egwindex2");
                                    $tr1 = @dbexecmany ($sql1, "egwindex2");
                                    //pg_free_result ($tr1);
                                    $sql2 = "insert into \"raw_$item\" values ('$ref',
'$parlen');";
                                    //$tr2 = @dbexec($sql2, "egwindex2");
                                    $tr2 = @dbexecmany($sql2, "egwindex2");
                                    //pg_free_result ($tr2);
                            }
                    }
            }
}
?>
```

parseegwcompdb-to-db3.php
```php
<?php
ini_set('memory_limit','999M');
include "dbfuncs.php";

echo "recreateding egwindex3 db;..\n";
$res = dbexec ("drop database egwindex3;");
if (!$res) die ("cannot drop egwindex3 database;");
$res = dbexec ("create database egwindex3;");
if (!$res) die ("cannot create egwindex3 database;");

$sql = "select tablename from pg_tables where tablename like 'raw_%';";
$res = dbexec ($sql, "egwindex2");
if (!$res) die ("no egwindex2 database?");
```

```php
$nmax = pg_numrows ($res);
for ($i = 0; $i < $nmax; $i++) {
        $tablename = pg_result ($res, $i, "tablename");
        echo "$tablename\n";
        $sql2 = "select ref, parlen, count(ref)  from $tablename group by ref, parlen
order by count desc;";
        $res2 = dbexec ($sql2, "egwindex2");
        dbexec ("create table c$tablename (ref text, parlen int, count int);",
"egwindex3");
        if ($res2) {
                $nmax2 = pg_numrows ($res2);
                for ($j = 0; $j < $nmax2; $j++) {
                        $ref = pg_result ($res2, $j, "ref");
                        $parlen = pg_result ($res2, $j, "parlen");
                        $count = pg_result ($res2, $j, "count");
                        $sql3 = "insert into c$tablename values
('$ref','$parlen','$count');";
                        dbexec ($sql3, "egwindex3");
                }
        }
}
```

parseegwcompreferences.sh
```
grep --only-match "{.*}" egw-comp-exported-genetic-txt.txt > egw-comp-
references.txt
```

i.php
```php
<?php
$s = @$_GET['s'];
if (strlen ($s)) {
        $fn = "i/";
        $ss = preg_replace("/[^a-zA-Z0-9 ]+/", "", $s);
        $fn .= $ss;
        $fn .= ".png";
        if (file_exists($fn)) {
                $file = $fn;
        } else {
                if (strlen ($ss) > 24) {
                        $ss = substr ($ss, 0, 9)."...";
                }
                $c = "convert -font Palatino-Bold -pointsize 42 -fill black -draw 'gravity
north text 0,62 \"$ss\" ' egwc.png \"$fn\"";
//              echo $c;
//              exit (0);
                `$c`;
                //copy ("egw.png", $fn);
                $file = $fn;
```

```php
        }
} else {
        $file = 'egw.png';
}

$type = 'image/png';
header('Content-Type:'.$type);
header('Content-Length: ' . filesize($file));
readfile($file);
```

dbfuncs.php
```php
<?php

function dbexec_old ($sql) {
  $con = pg_connect ("dbname=wingp host=localhost password=a user=wingp");
  $res = pg_query ($con, $sql);
  return ($res);
}

function dbexec ($sql, $dbname = 'wingp') {
  static $dbnamex = NULL;
  static $con = NULL;
  if ($dbname != $dbnamex) {
        if ($con) {
                pg_close ($con);
                $con = NULL;
        }
  }
  if ($con == NULL) {
        $con = pg_connect ("dbname=$dbname host=localhost password=a
user=postgres");
        $dbnamex = $dbname;
  }
  $res = pg_query ($con, $sql);
  return ($res);
}

function dbexecmany ($sql, $dbname = 'egwindex2') {
  $constr = "dbname=$dbname host=localhost password=a user=postgres";
  static $con1 = NULL;
  static $con2 = NULL;
  static $con3 = NULL;
  static $con4 = NULL;
  static $con5 = NULL;
  static $con6 = NULL;
  static $con7 = NULL;
  static $con8 = NULL;
  if ($con1 == NULL) { $con1 = pg_connect ($constr); }
```

```php
  if ($con2 == NULL) { $con2 = pg_connect ($constr); }
  if ($con3 == NULL) { $con3 = pg_connect ($constr); }
  if ($con4 == NULL) { $con4 = pg_connect ($constr); }
  if ($con5 == NULL) { $con5 = pg_connect ($constr); }
  if ($con6 == NULL) { $con6 = pg_connect ($constr); }
  if ($con7 == NULL) { $con7 = pg_connect ($constr); }
  if ($con8 == NULL) { $con8 = pg_connect ($constr); }
  $sent = 0;
  $con = NULL;
  while (!$sent) {
        if (!$sent && !pg_connection_busy($con1)) { $sent = pg_send_query ($con1,
$sql); $con = $con1; }
        if (!$sent && !pg_connection_busy($con2)) { $sent = pg_send_query ($con2,
$sql); $con = $con2; }
        if (!$sent && !pg_connection_busy($con3)) { $sent = pg_send_query ($con3,
$sql); $con = $con3; }
        if (!$sent && !pg_connection_busy($con4)) { $sent = pg_send_query ($con4,
$sql); $con = $con4; }
        if (!$sent && !pg_connection_busy($con5)) { $sent = pg_send_query ($con5,
$sql); $con = $con5; }
        if (!$sent && !pg_connection_busy($con6)) { $sent = pg_send_query ($con6,
$sql); $con = $con6; }
        if (!$sent && !pg_connection_busy($con7)) { $sent = pg_send_query ($con7,
$sql); $con = $con7; }
        if (!$sent && !pg_connection_busy($con8)) { $sent = pg_send_query ($con8,
$sql); $con = $con8; }
        //usleep (5);
  }
  return ($con);
}

function dbcount_old ($sql) {
  return pg_numrows (dbexec ($sql));
}

function dbcount ($sql, $dbname) {
  return pg_numrows (dbexec ($sql, $dbname));
}

?>
```

G. Technical Reference

Final system specifications

The system has been developed and tested on the following server hardware and this is the minimum recommended:

Hardware: i7 3rd gen or better with at least 8gb ram.

Operating Systems: Ubuntu 18.04 or Windows 10

Programming Language: PHP, SQL

Server applications used: Apache Webserver, PostgreSQL, bash, ImageMagick (for Facebook customized photos)

Maintenance plan for the software system

The system maintenance should be done by people who love EGW writings. They have to go on discovering new questions people ask and encode answers to those question from EGW Writings. This will continuously make the system up to date with trends. It will also make it more relevant and useful for most people and other search engines.

H. User Manual

Question: **How to search?**

Answer: visit this URL https://a.aolis.aup.edu.ph/egwcindex/ and type in your question /

topic / interest.
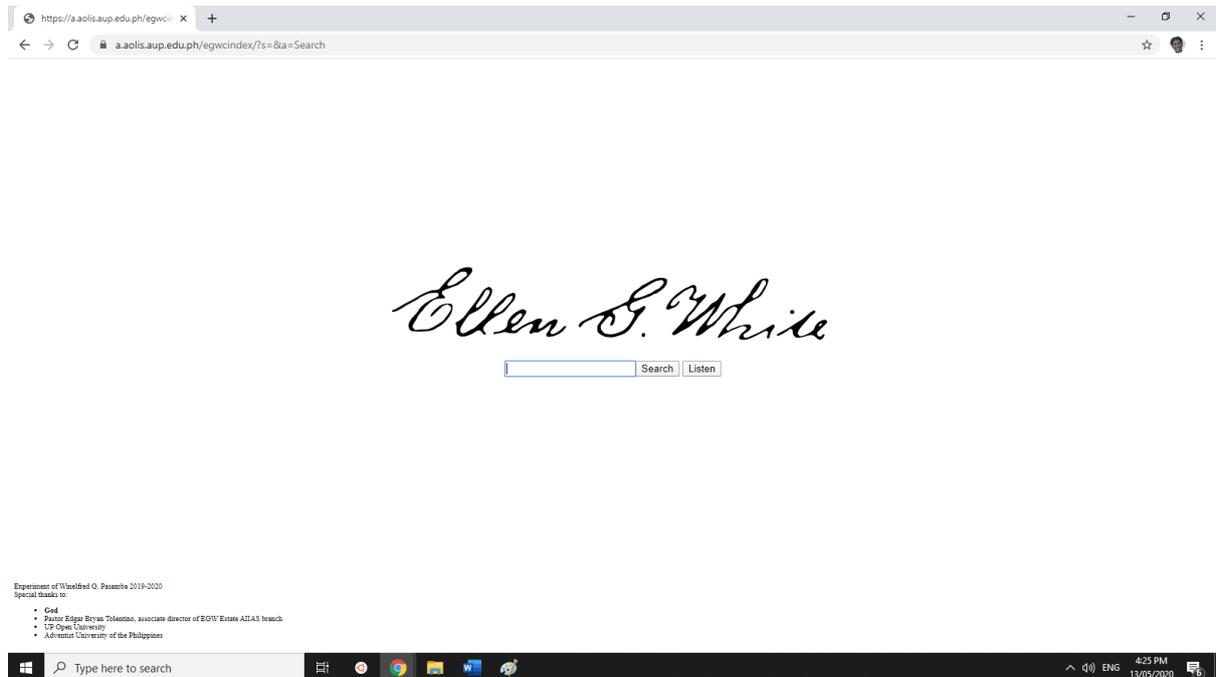
Question: How to contribute user suggested results?

Answer: paste the reference of your suggested paragraph in the textbox below and

click Submit. A "Thank You!" message will show if the paragraph exists in the

database. Then the next time the query is searched, your entry will show.

Above there are not good results. Clicking "Open another search tab" and searching

something easier for the index to find, we find a good answer. Just copy the

reference shown.

Then paste into the textbox below and click "Submit".

The next time the query is searched it will show the paragraph we suggested.

Pressing Like on a paragraph will raise the vote by one. Results that have more votes, come up first.

Question: How to install from sources?

Answer: This is a website and it uses free but complicated software. So minimally the person who wants to run this server has to be comfortable with Linux servers in using the terminal, PostgreSQL server including tuning the database for speed, configuring Apache webserver, and other general website tasks. Linux system administration common sense is required, but here are the steps in general:

- Create default database (wingp/egw1)

- Create database egwindex2, egwindex3, egwtopicalindexindex

- Run parsing scripts: parse*.php in shell; troubleshot errors;

- Update dbfuncs.php as needed

- Test the service online

- Test Facebook customized sharing