Presents of Mind

By Alex DeFiore, Alexis Fatica, and Zach Janice

Table of Contents

1. Abstract

Our project idea is called Presents of Mind, which is a website that will aid a person in finding the perfect gift for any occasion (birthday, holiday, anniversary, etc). The user will input some information about the occasion as well as the target person's Facebook profile (and/or manually inputted information). A machine learning algorithm will be used to develop possible gift ideas using a training set of Facebook profiles matched to enjoyed gifts. The output of the machine learning algorithm will be run through Amazon's API to determine other similar gifts that the target may enjoy, which will be returned to the user.

2. Introduction

2.1 Relevant background

Finding the perfect gift for a loved one can be difficult. The pressure to find a gift that is personal and thoughtful can be overwhelming. Using the information posted on the person's Facebook profile could make this process much easier as the amount of personal information people put on Facebook continues to increase. This information combined with Amazon's wide range of products could be a powerful tool in making gift giving a much more enjoyable and less stressful task.

2.2 Project goals

The primary goals for the project are to write a script to scrape the target's information from their Facebook profile, and to develop a connection to Amazon's API to pull gift ideas according to the interests of the target. Another goal is to develop a machine learning algorithm to associate a target's profile with training data in order to find potential gifts tailored to the individual. The output of the machine learning feed directly into the Amazon search. Other goals for the project include creating a user interface to allow users to enter target information manually, develop a database to house training data as well as users' data, and develop a feedback system to determine which, if any, gifts were chosen by the user for purchase.

2.3 Related and/or similar systems

The Gift-O-Matic is a website which aims to aid people in finding a gift by using a target's Twitter account. However, it uses gifts on Ebay which is more limited and less popular than Amazon. Additionally, a person's Twitter provides different information from that of their Facebook profile as Facebook allows for more personal information to be shared with friends.

2.4 Advantages

1) Provides gift ideas for special occasions
2) Can spark other similar gift ideas
3) The results will be unbiased in that no specific line of products will be promoted

2.5 Disadvantages

1) Requires a training set of data
2) Only provides gifts that are available on Amazon

3) Only uses information posted on the target's Facebook (and manually entered information)
4) Due to Facebook restrictions, the website can only visit the Facebook profiles of people who have visited the site
    a) This means that a user will have to manually enter the information of the target unless the target has previously visited our site
    b) Additionally, this means that information can't be pulled from the Facebook profiles of the training set data individuals - in place of this each individual's profile was manually read to extract data. However this is obviously not ideal

3. Application Description

3.1 Constraints

1) Difficult to collect enough relevant training data
2) Data present on Facebook is limited and varies person to person

3.2 Challenges

1) Determining which information on Facebook is useful
2) Having enough training data to effectively map target to similar people
3) Sorting through Amazon search results for decent results
4) Keeping the personal information that we gather private

3.3 Solutions

1) Word frequency analysis is used on the target's posts to determine words/phrases that the target focuses on in their posts.
2) A Google form was sent to family and friends asking them to help gather information by providing the link to their Facebook profiles as well as gifts they've received in the last year that they enjoyed.
3) After the information from the target's Facebook profile is retrieved, the top 10 words/phrases used in their posts are stored in the database. All other data is discarded. The target's name is not stored in connection with their Facebook profile information.

3.4 Methodology

Their were no problems encountered with data strucutures, as none more complicated than databases, dictionaries and lists were used. The machine learning algorithm posed some technical challenges due to analyzing strings. The algorithm chosen determines the distance between strings using Levenshtein distance, which equates distance with the number of times

you would have to change one string to transform it into the other. This is a comparison of the characters themselves and not the meaning of the words, which is not ideal. However, with adequate training data this problem would be solved as the post data contains more and more words. Considering the small amount of training data collected, this is the biggest source of error for choosing gifts to show the user.

4. Software Requirement Specifications

4.1 Functional Requirements

4.1.1 F1 Facebook Integration

1) F1.1 The user logins to Facebook through the website.
2) F1.2 As a backup, on each login, the user's name, Facebook ID, and Facebook access token are updated in the database in case the information is not otherwise available.
3) F1.3 The user may begin a Facebook search by entering the name of the friend they'd like to search for, permitting that friend has used this application and is subsequently available for search.
4) F1.4 The website, using the user's access token granted by Facebook, pulls relevant information from the identified friend's Facebook profile using Facebook's provided API.
5) F1.5 In receiving this response from Facebook, the information is parsed through for information relevant to potential interests, and pruned for relevancy (see 4.1.2 F2 "Facebook Data Analysis"). The resulting information is then used in other procedures of this web application.
6) F1.6 As an alternative to searching via Facebook - in the event a desired target does not have a Facebook profile, the desired target has not used the web application, or the information received from Facebook about the target and/or the results are insufficient - a manual form exists that mimics the intended functionality of the Facebook integration. This form may be filled out by the user as to manually provide information regarding the user's target.

4.1.2 F2 Facebook Data Analysis

1) F2.1 Information is received from Facebook in the form of an object containing (unless otherwise unavailable or not provided) the target's name, age, gender, hometown, and various information regarding potential interests (likes, posts, etc.)).
2) F2.2 Information regarding the target's posts on Facebook are divided into two collections: Stories (overviews of what the post is) and Messages (the posts themselves).
3) F2.3 Story information is pruned and parsed for stories regarding the sharing of some page. As this page may represent an interest, page titles of this nature are returned.

4) F2.4 Message information is processed by cleaning the messages (removing emojis, other special characters, etc.), separating out hashtags (to be returned as separate keywords), and sent through word analysis to result in a list of keywords that identify potential interests the target posts about with frequency.
5) F2.5 The story, message, and hashtag information - as well as the other information provided by Facebook - are then available for use by other procedures that follow in the search process (see 4.1.3 F3 “Machine Learning”).

4.1.3 F3 Machine Learning

1) F3.1 Training data (friends’ Facebook accounts & gifts they liked) is obtained via a Google form with express permission to use their Facebook profiles, and training data will be added with iterations of the website
2) F3.2 The Facebook profiles of the training set users will be parsed in the same manner as the target’s profiles
3) F3.3 The data obtained through the training set user’s Facebook profile is stored in a database along with the gift ideas
4) F3.4 A k nearest neighbors clustering algorithm is used to match the target’s Facebook data with similar users in the training set and the gift ideas of these similar users will be fed into Amazon

4.1.4 F4 Amazon Integration

1) F4.1 The website receives the suggested gifts matched from the training data, from either analysis on Facebook data, or manually-entered data on the target via the user interface
2) F4.2 The website searches for products on Amazon through the Amazon Product Advertising API, using the received suggestions
3) F4.3 The website receives a list of products, each with name, ASIN, link, price, and image

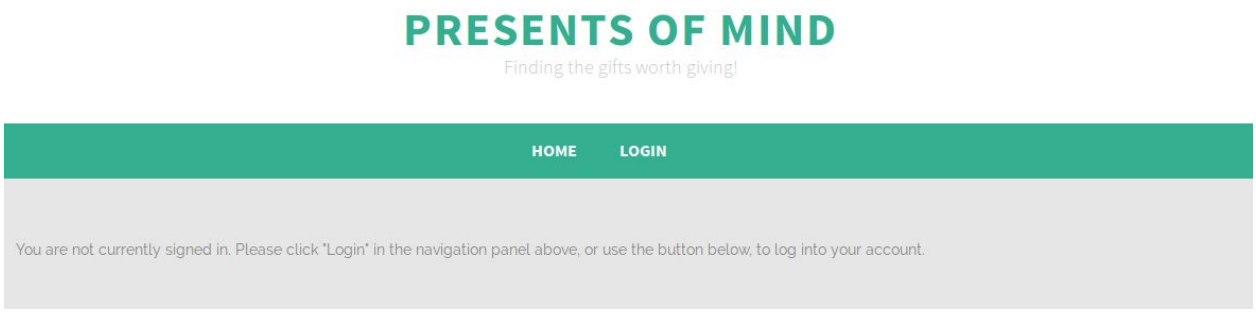
4.1.5 F5 User Accounts

1) F5.1 Users are required to login to their Facebook account in order to use the website.
2) F5.2 The user may visit the “Account” page, where they will see their name (to verify their identity) as well as their past searches performed.
3) F5.3 The user can select a specific search and obtain information regarding the gift ideas that resulted from that search.

4.2 User Interface Requirements

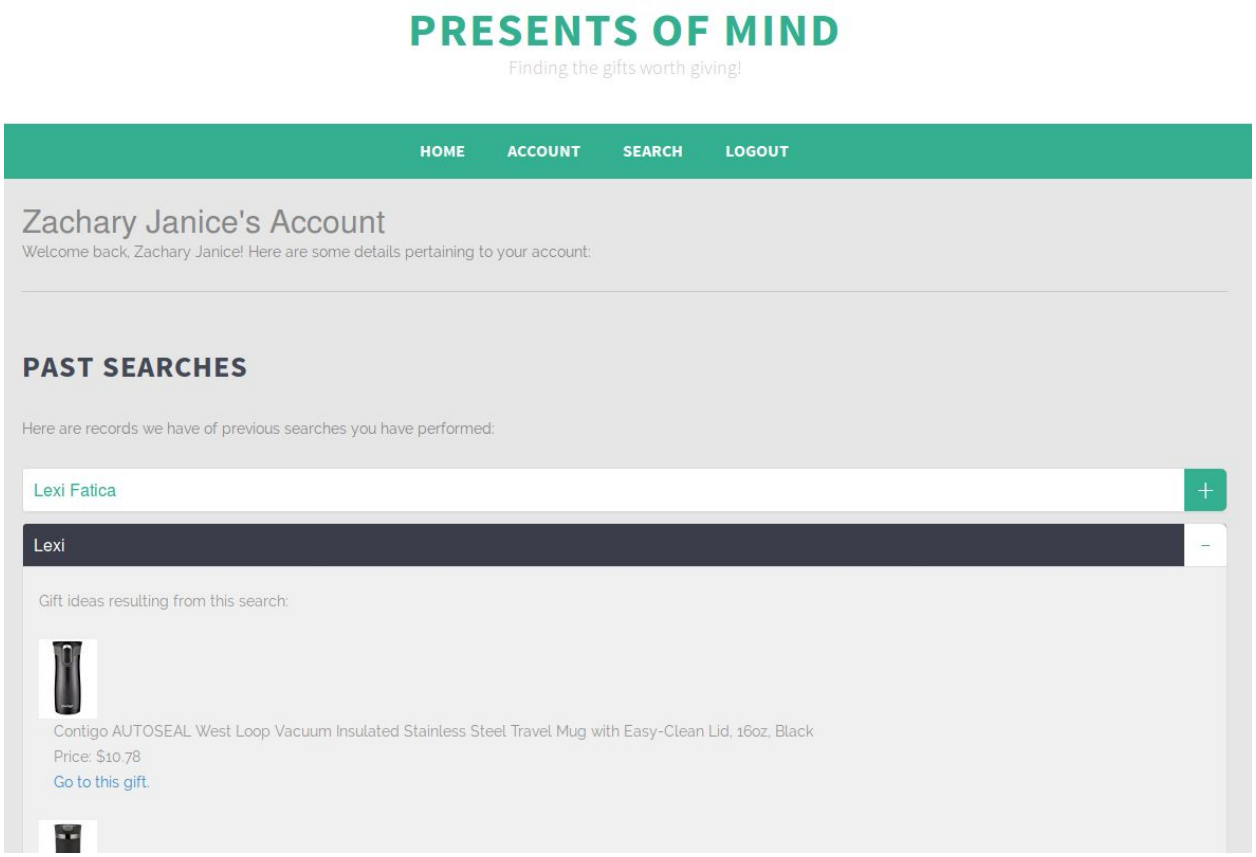
4.2.1 U1 User login

1) U1.1 To enforce the necessity of being logged into Facebook, the user must be logged in to utilize the web application. Necessary links and navigation are provided as to direct the user to log in using Facebook's login API.
2) U1.2 A logged-in user has the ability to log out of the web application. This is done using Facebook's logout API.



4.2.2 U2 Viewing past searches

1) U2.1 A logged in user can view their past searches by navigating to the "Account" page.
2) U2.2 The past searches will be ordered by date searched; the most recent search appears at the beginning of the list, and the list continues by ascending age of the searches.
3) U2.3 The past searches are labeled by the name of the user's target in that search, or by an arbitrary label the user provided in the case of a manual search performed.
4) U2.4 The user may select a search to view the gift ideas that resulted from the search. Information shown for these ideas include the ideas' names, prices, image, and URL's for their pages on Amazon.

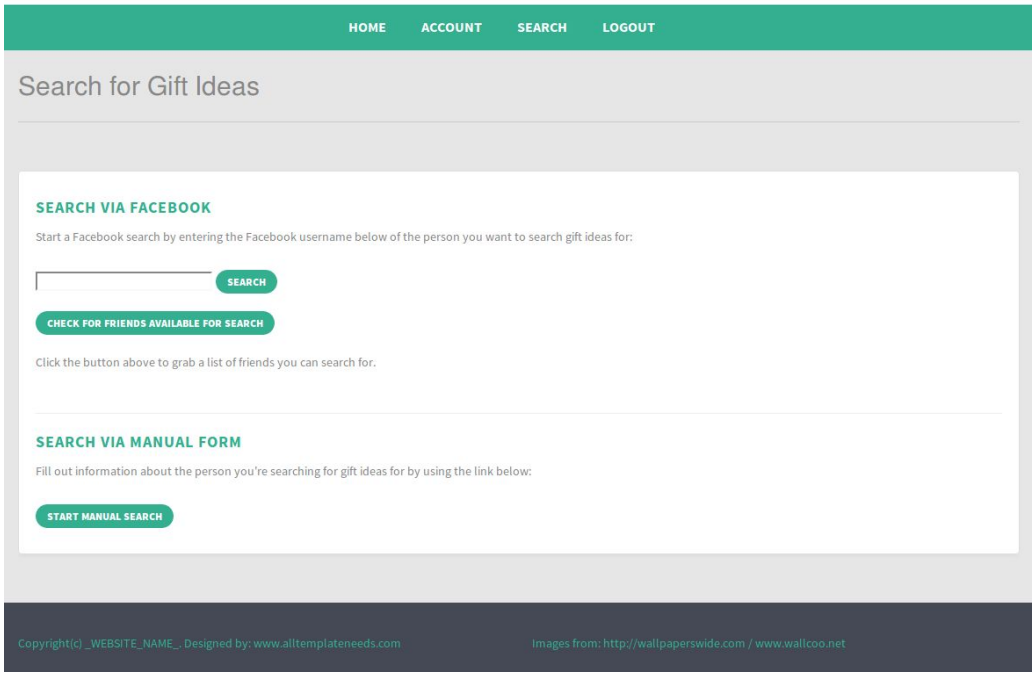


4.2.3 U3 Starting a new search

1) U3.1 The user may begin a search through Facebook by entering the name of a target that has used the web application.
2) U3.2 The user may prompt the web application to display a list of valid targets, or friends of the user that have used the web application.
3) U3.2 The user may start a manual search, where the user completes a form of information including an arbitrary label for the search, the target's age, the target's gender, the target's hometown, and a list of interests for the target.

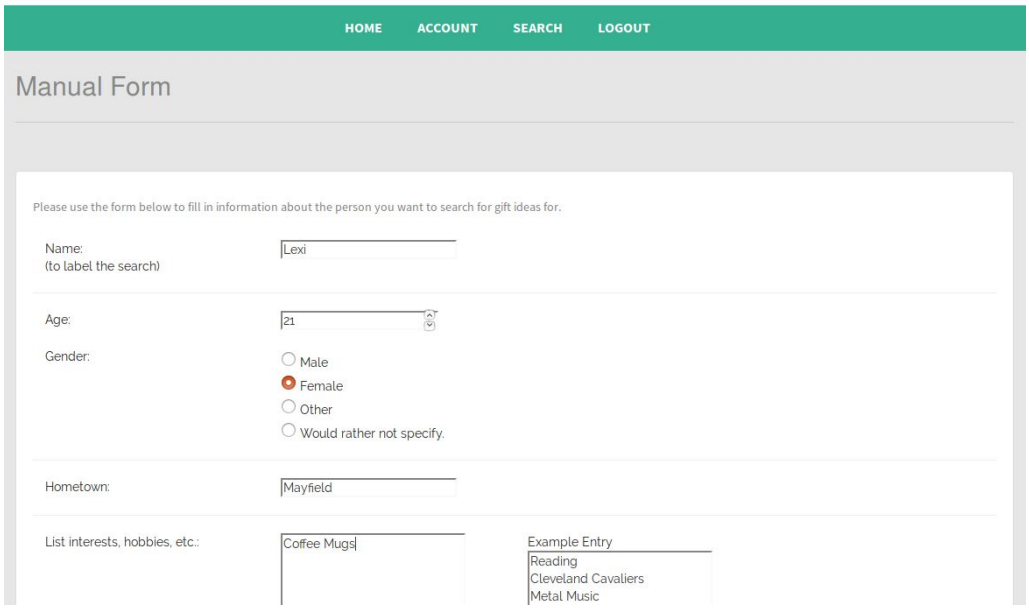
PRESENTS OF MIND

Finding the gifts worth giving!



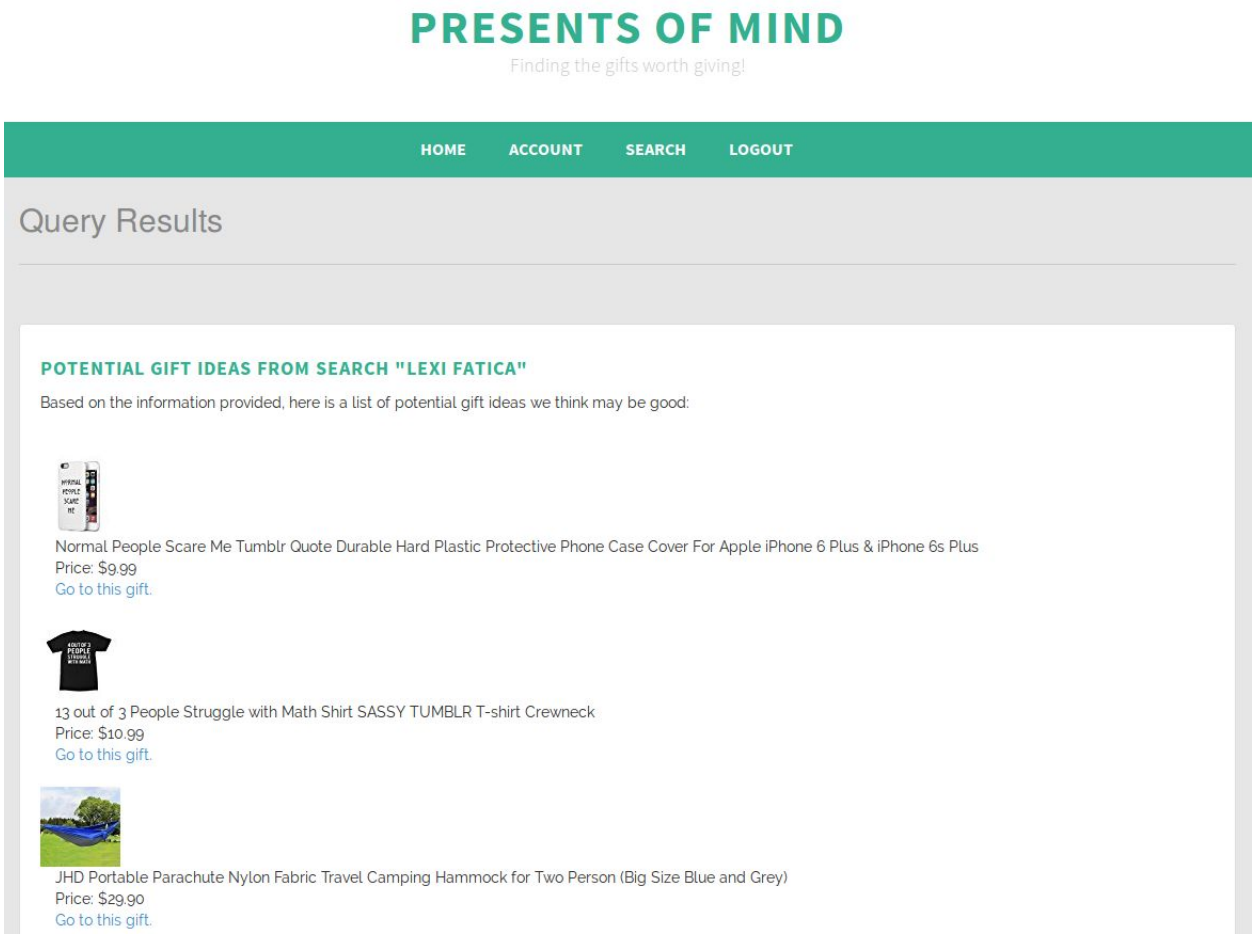
PRESENTS OF MIND

Finding the gifts worth giving!



4.2.4 U4 Viewing search results

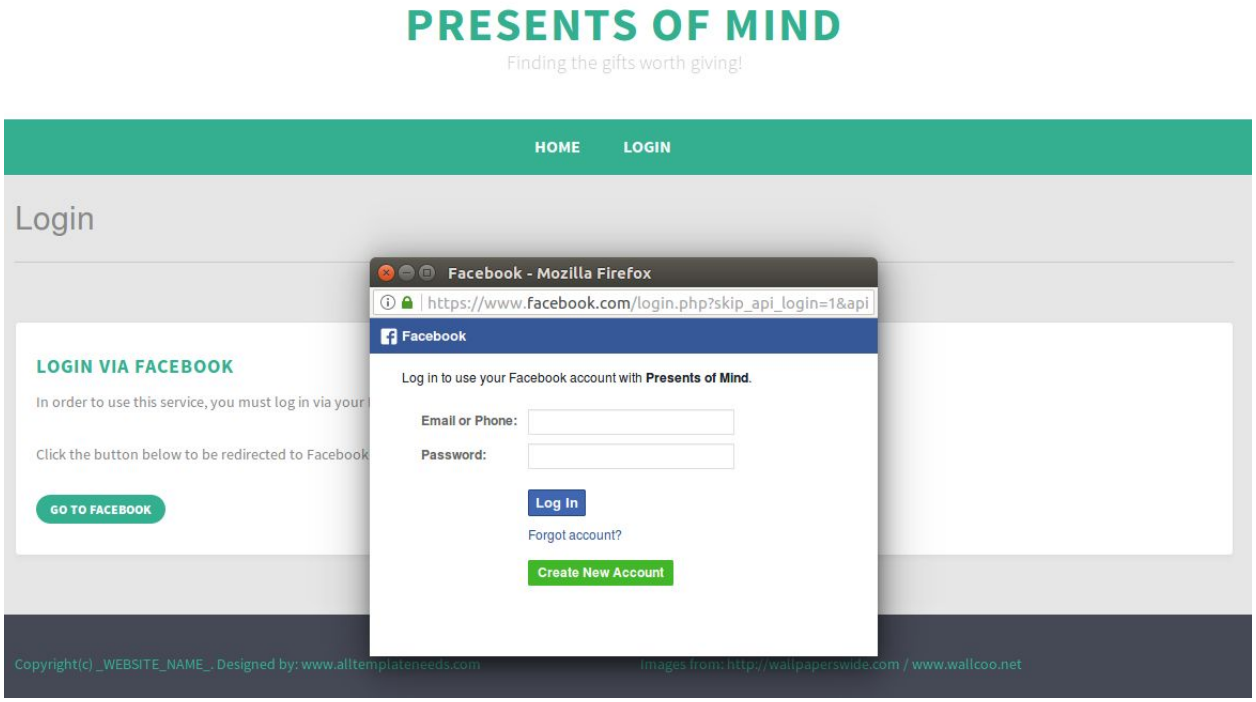
1) U4.1 The user is shown the list of gift ideas resulting from the search just recently performed.
2) U4.2 The associated label of the search is displayed along with the results.
3) U4.3 Each gift idea entry displays information including the name, price, and Amazon URL of the gift idea.

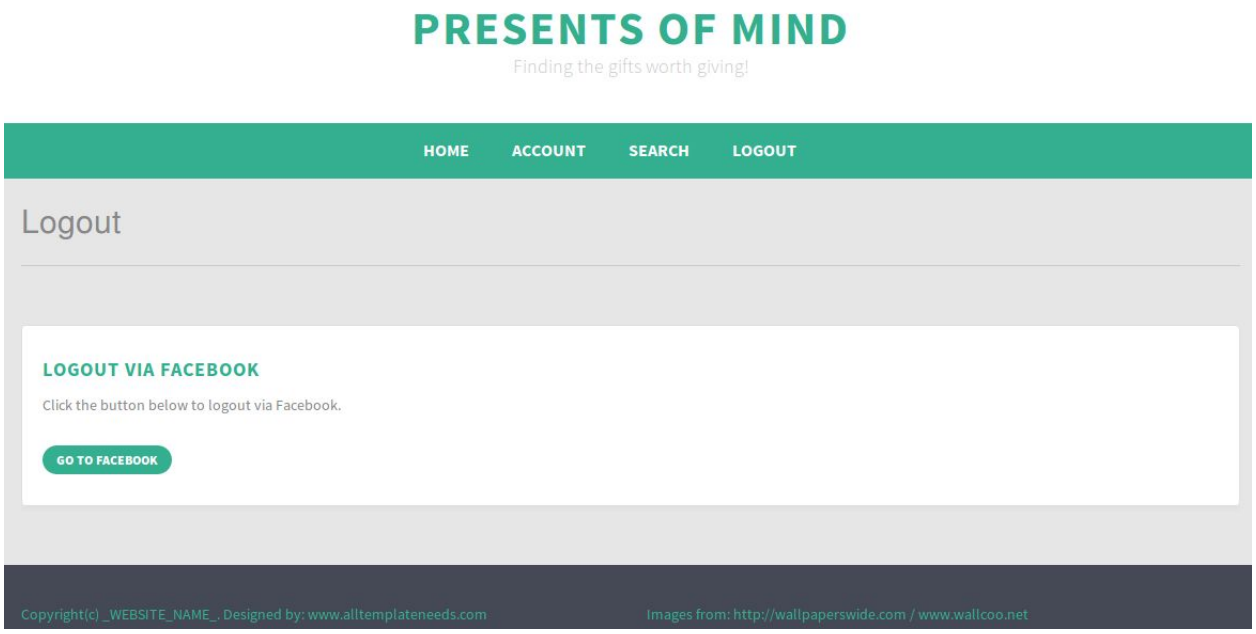


4.2.5 U5 General UI

1) U5.1 There is a navigation bar near the top of each page of the website, allowing for access to different pages of the website.
2) U5.2 The contents of the navigation bar change depending on the login status of the user, and direct functionality for the user as a result.
3) U5.3 The pages of the website maintain a consistent style, using one shared source for CSS resources.

4) U5.4 Each page extend from a common “wrapper” HTML file, which provides a consistent outlining page setup.





6. Testing and Evaluation

Overall, there are no comprehensive tests for the project code. Obviously this a large oversight, which is not preferable. However this is not to say no testing was performed whatsoever. As code was written, small test scripts were used to make sure it was working in a reasonable fashion. These files are included with the code, local to the files they test. Also, the main goal in terms of the tests performed were the integration tests to have the entire web application working. Due to someone always having to be logged in for integration testing, it was not reasonable to automate it, and it is not as easily replicable. Lastly, evaluating overall gift idea results for a specific person is very subjective. Therefore, in testing the overall results, there is no feasible way to automate it, as each result must be subjectively evaluated.

7. Lessons Learned

1) Connecting the website to Facebook and mounting the website were much harder tasks than anticipated.

   There was trouble with getting Facebook working. The original approach was to use a Facebook API for Python, but there wasn't much support in terms of achieving the API calls that were intended. While the Python API would likely have worked, not enough aid in documentation / example code existed to make it worth pursuing as the form of implementation.

   Facebook's API was then settled on for a web application, as the project was changed from a standalone application to a web application in order to accommodate the Amazon API's needs. However, the main hurdle with this approach was the need for a domain for the web application, as Facebook would have that domain authorized for redirect calls after logging in. Without this domain, Facebook would not allow the application to log in. 'localhost' was attempted to be used as a domain, but at the time it was unclear how to have Facebook allow 'localhost' as a valid domain.

   Part of the secondary goal in solving the domain issue was to make the web application publically available for the sake of testing. Approaches to this included a shared server on GoDaddy, though would not provide the permissions and resources needed to install Python with Flask and SQLAlchemy on the server and thus the server was not used. After more research into the matter, another approachtried was Heroku, a similar service that specifically allows for hosting a Python server with Flask.

   In the midst of investigating this further and trying to make it work, a lucky find on YouTube showed the exact setup needed for 'localhost' to be validated with Facebook,

allowing Facebook to be working and testable on local machines. The problem was this was only achieved after much effort and, as a result, with about a week left to work on the project.

In short, whether attributed to lack of in-depth-enough investigation ahead of development or to unforeseen conflicts between the tools chosen, integrating these tools and attempting to publish the application created many more complications than anticipated, which cost a lot of time in the development of this project. While these complications are (more or less) resolved now, time to develop the project to the extent we would like regrettably ran out.

2) Collecting training data is a major obstacle that is hard to overcome as undergraduate students without the resources to connect with a vast number of people. If this was to be operationalized, it would be well worth taking time (and money) to recruit people into providing gift ideas and logging into the website via their Facebook in order to pull their profile information. This would not be too difficult if an incentive like cookies or a gift certificate could be provided, however resources for such incentives were not available. Additionally, it was believed that, as a senior capstone project, showing the potential in this idea was more important than building a training set to increase the accuracy of the recommendations. On the other hand, using the post information alone yielded more impressive results than expected and could reasonably be used on its own to determine gift ideas in the absence of adequate training data.

3) The idea behind the project remains something that could be a real, working application. Seeing the project come along and speculating on the potential of the application was exciting. Most of the results that were received are fairly applicable and good sources of inspiration in terms of gift ideas. Some of the gift ideas are even worthy of being selected as an immediate idea for someone, should we need to order them a gift that instant.

   That being said, it is understood that this project can be improved. More expansive application of this project (interest gathering, analysis, training data, etc.) could lead to an incredible product, but reaching such a point is admittedly beyond the scope of a semester's worth of work as there would be much more extra work to go into the project to meet that goal. In that sense, the goals set at the beginning of the semester may have been overly ambitious, but the final product is a resemblance of the vision and direction intended.

8. Conclusions, Final Status, and Future Work

The majority of the intended work was completed, but not as much as would be ideal. A major obstacle was encountered in the web hosting portion of the project, and restrictions

imposed by the Facebook API were discovered. As a result, a working prototype is available that can be locally hosted, but a website for anyone to access does not yet exist. Another difficulty that was encountered was gathering a large enough set of training data for effective machine learning. However, the product created is still quite effective. It gathers a large set of accurate interests from the Facebook profile, and the resulting results effectively show the relevant products on Amazon, along with a link to each item's product page.

1) Were the application specifications achieved? If not, why not?

   Most, but not all, specifications were achieved. The ones not achieved were mostly due to unfamiliarity with a few of the tools used, as well as unexpected issues with web hosting. The most unfortunate problem encountered was Facebook not allowing the access of a person's page unless the individual themself had at some point given the application permission.

2) What testing/evaluation tools frameworks have been used?

   Testing and evaluation frameworks were not developed as in depth as would be ideal. Code was tested locally on team members' machines as progress was made, however a comprehensive testing framework was not implemented. These test scripts are included in the code provided. If given more time, a more structured framework around testing would be developed thoroughly. Integration tests were performed ensure the application was run cohesively. These tests could not be automated due to the requirement that someone be logged into Facebook on the application.

3) Which frameworks/tools/libraries were used, dropped, added, and why?

   The frameworks used were Flask, Jinja2, SQLAlchemy, PosgreSQL, Amazon Product Advertising API, and the Facebook API. Changes to these frameworks were not made throughout the project as it was not necessary.

4) What should be done next?

   Next steps would include, first and foremost, finding a decent way to mount the web application on a server for public use. A service like Heroku would be the best avenue for this, but, given the timeline of obstacles being encountered and being solved, this became a task that fell out of scope for the sake of proving the concept of the project. Security will also need to be addressed; protections against SQL injection and other forms of malicious usage of the application have not been accounted for yet.

   From there, perhaps more work can be done for investigating what information can be returned by Facebook's API such that more interests can be derived. In the same

respect, perhaps more social media sites or more online shopping sites could be consulted to improve the output of this project. Finally, more work may be able to be done on the machine learning side of the project. Whether the algorithm is improved upon or the training set is expanded, more attention to this area can likely yield better results.

5) Is this ready for production, or can improvements be made? If so, what?

   As discussed in the above point, improvements can be made before this application hits a live, public server. The main improvement needed is security, as protections are not in place against forms of attacks such as SQL injection. Stress testing will also need to occur in order to ensure the server can withstand the load of going into public use. Finally, some system would need to be in place as to limit / clean up the searches kept for an individual user, so as to not bog down the database.

   Beyond these main concerns, the actual functionality of the application should be ready to see production.

6) Were there major problems which require a reevaluation of the project results?

   As discussed in the “Lessons Learned” section, heavy resistance was encountered when trying to implement Facebook’s API and get the web application up and running on a public server. Development around these issues was attempted by making assumptions around Facebook’s input and output regarding the API and trying to form the processes around those assumptions. The project was tested locally on individual team members’ machines. Despite efforts to stay on track, these issues cost valuable time testing integration and getting the project developed as one general whole.

   While solving the Facebook API issue was a great success, the resolution of the issue left little time to bring the pieces of the project together. In light of the new challenges faced by the Facebook API when it was finally working (specifically, Facebook’s privacy policy concerning app usage and whose information is available under what conditions, as well as the actual queryable information), changes had to be made that set the project back.

   Therefore, all goals originally theorized were not fulfilled unfortunately. While most of the functionality exists in working form, stipulations from Facebook and a couple final hurdles prevented the completion of all original goals. In part, the scope of the project is to blame, but unforeseen complications are the main culprit for this.

7) Basically, have the project goals been achieved and what are the next steps?

   In summary, yes*. Yes, using someone’s Facebook account, the interests of an

individual can be gauged and used to come up with gift ideas for that individual. Yes, in this absence, a manual form can be used to accomplish the same depending on what the user knows about the target. Yes, the search information is kept by user and is recallable. Yes, the analysis of interests includes use of machine learning and Amazon's API. And yes, the results are decent (at least, frankly, better than expected).

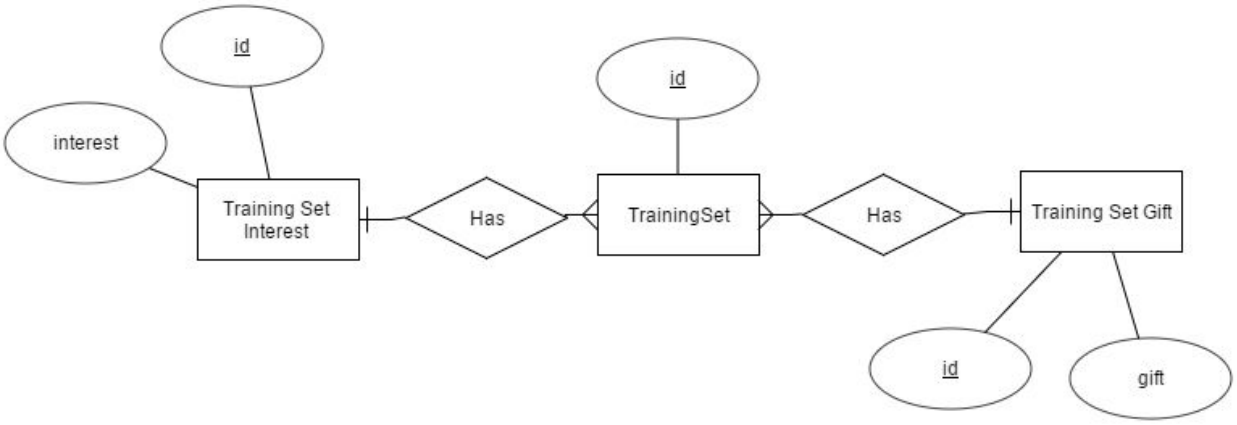
However, where the asterisk in yes* comes into play, Facebook has limited the conditions under which profiles can be pulled. Not only that, but information such as age, gender, and hometown are understandably higher in sensitivity than posts, thus the current failure to utilize those specific pieces of information. And, finally, due to complications that obstructed the process, the website is not publically available.

What remains is still something to be proud of: working proof of concept for a web application which has potential to be something incredibly useful. The immediate next steps, as outlined, are to better ready the application for production, but this application would have popularity if and when that time of production comes. From there, improvements on gathered information and its processing could take this project to that vision we had.

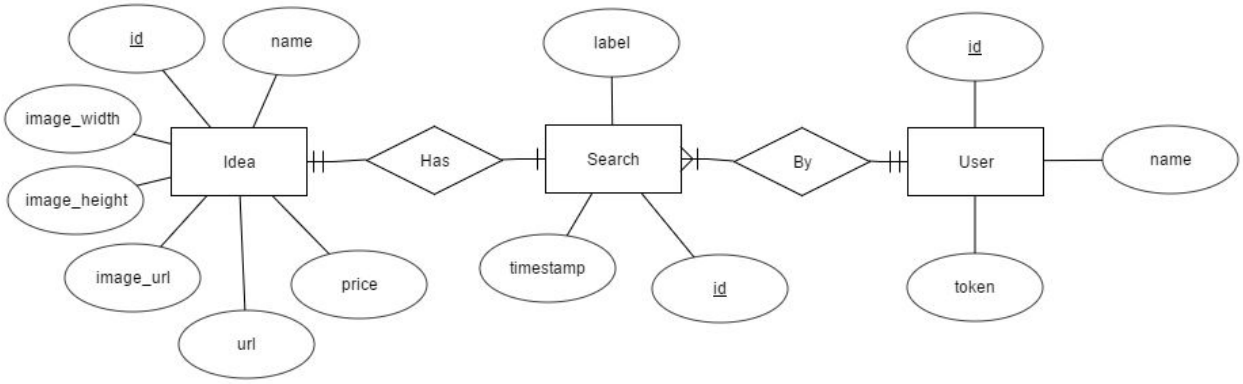
9. Appendices

9.1 Database Design

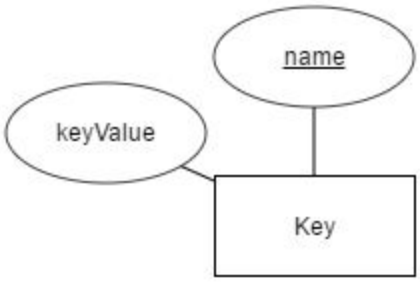
9.1.1 Training Set Data



9.1.2 Users and Searches



9.1.3 Secret Keys (for Facebook)



9.2 User Manual

1) Enter login information
2) Enter target Facebook information
3) Enter price range (optional)
4) Receive results

9.3 Programmer's Manual

The following section describes the process by which Presents of Mind can be set up to run as a local server for operation on that machine.

1. Tool Installation

The web application was constructed and tested in an Ubuntu environment (16.04) using the tools Flask, Jinja2, PostgreSQL, SQLAlchemy, Amazon's Python API for product searching,

LXML, Stemming, NumPy, and SKLearn. Respectively, the following commands can be run to install these dependencies in the Ubuntu environment such that the server and its tools are ready to run:

```
sudo apt-get install python-pip
sudo pip install Flask
sudo pip install Jinja2
sudo apt-get install postgresql
sudo pip install SQLAlchemy
sudo pip install python-amazon-product-api
sudo pip install lxml
sudo pip install stemming
sudo pip install numpy
sudo pip install sklearn
Sudo pip install psycopg2
Sudo pip install scipy
```

As a result of running these commands in terminal, the dependencies needed for the server's operation are downloaded and ready for use, with the exception of Amazon's API.

2. Add Credential File for Amazon API

The next step is to include a configuration file for the Amazon API, to be created in the root directory. Navigate to '~' in a terminal window and, in this directory, create a file titled `.amazon-product-api` and place the following text inside the file:

```
[Credentials]
access_key = AKIAJYNKB62KVJSSFCXQ
secret_key = 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
associate_tag = chewie027-20
```

These credentials will be retrieved by the Amazon API whenever it is called, and as such the API will use these credentials to authenticate the searches.

3. Set Up Database

With PostgreSQL installed, a new database needs to be created within PostgreSQL. Open a session with PostgreSQL by using the default user:

```
sudo -u postgres psql
```

From here a new database can be created with the equivalent query:

```
create database pom;
\q
```

The credentials within `[project]/UI/site/db_creds.txt` need to have working credentials in order to create a session with the database when the application runs. The file provided with the project uses a username ‘test’ and password ‘test’, which corresponded to a test user we created within the database for this project. Update these two fields to match the username and password of the user to be used to connect to the database (again, the default user is ‘postgres’). To, for example, create a user with name <<name>> and password <<password>>, use the following commands:

```
sudo -u postgres psql pom;
create user <<name>>;
alter user <<name>> password ‘<<password>>’;
alter user <<name>> SUPERUSER CREATEDB BYPASSRLS;
\q
```

This username/password combination in the credentials file should be used. **Make sure to change the credentials in db_creds.txt as applicable.**

Next, the training data must be imported into the PoM database. Run the following script within `[project]/UI/site/`:

```
python ImportTraining.py
```

This script creates all the necessary tables in the database, as well as populates the training set tables so that training data is available for the machine learning. The data imported can be found in the directory `[project]/UI/site/data/`.

Finally, necessary keys must be included in the database as well. Both a unique key for the application and Facebook’s application ID will need to be added to the `keys` table of the database. To accomplish this, use the following command to access the PoM database:

```
sudo -u postgres psql pom
```

And then use the following queries to insert the app key and Facebook ID, respectively:

```
insert into keys values (‘app_key’, ‘AAX1$*.d/21532&HSD*[]ASD’);
insert into keys values (‘fb_app_key’, ‘1483387461695099’);
\q
```

These keys will allow the application to keep encrypted cookies and run Facebook’s API calls on behalf of the application.

4. Run the Server

The server is ready to be run. In `[project]/UI/site/`, run:

```
python Run.py
```

In a browser, navigate to `localhost:5000/` and the web application should be running.

9.4 Incorporated Technologies

1. Python 3
2. Flask / Jinja2
3. SQLAlchemy
4. PostgreSQL
5. Facebook API
6. Amazon Product Advertising API