มหาวิทยาลัยอัสสัมชัญ
ASSUMPTION UNIVERSITY
of THAILAND

VINCENT MARY
SCHOOL of SCIENCE and
TECHNOLOGY

Senior Project Report

iLearn

Pinky Shahi,
Suchawan Chaiworn,
Lipika Bania

A. Kiratijuta Bhumichitr (Advisor)

CS 4200 Senior Project 2 (2/2021)

Senior Project Approval

Project title:	iLearn	
Academic Year:	2/2021	
Authors:	Suchawan Chaiworn	6135102
	Pinky Shahi	6135205
	Lipika Baniya	6138108
Project Advisor:	A. Kiratijuta Bhumichitr	

The Senior Project committee from the Department of Computer Science, Vincent Mary School of Science and Technology, Assumption University had approved this Senior Project. The Senior Project in partial fulfilment of the requirement for the degree of Bachelor of Science in Computer Science and Information technology.

Approval Committee:

……..
(A. Kiratijuta Bhumichitr)
Project Advisor

……..
(A. Chayapol Moemeng)
Committee Member

……..
(A. Tapanan Yeophantong)
Committee Member

Acknowledgement

We would like to give a special thanks to those who contributed to our senior project. This project is extremely important for us not only to apply Computer Science knowledge in the real world but also learn and utilize the technologies.

We would like to express our deepest appreciation to our project advisor A. Kiratijuta Bhumichitr for giving us practical suggestions during working progress. Much useful advice was given.

Moreover, we thank the committee members A. Chayapol Moemeng and A. Tapanan Yeophantong for constructive comments on iLearn so we can utilize it for improving our application. Furthermore, we would like to thank our Assumption University lecturers who provided us with knowledge and helped us develop our skill, which helped us out to complete the project.

Lastly, we would like to thank our families and friends who inspired, encouraged, and supported us along the way. They gave us hope that the project would be possible, and we could make it more interesting.

Abstract

Many companies want to gather data from their customers or users but are unable to do so, because they might be a small company. Big companies like Facebook and Google have enormous amounts of data about all their users but face many privacy issues and concerns. Small companies don’t want enormous amounts of data but just enough to help with their research. That is the problem we’re going to solve. As a solution, we will build a platform for companies to host surveys on a website and gather user’s opinions and data. A web scraper will help gather images to display to the user while a search expansion will help analyze the user’s responses and scrape better images to display while also increasing the accuracy. For our senior project, our system will scrape images from the internet, take survey answers from users as well as perform search expansion for higher accuracy and performance.

Table Of Contents

Table Of Figures

Chapter 1: Introduction

This chapter explains an overview of the online survey platforms and its related resources. It also includes the purpose of this project as well as goals and limitations.

1.1 Rational and motivation

Online survey platforms are very commonly used to gather user's data. Many companies or people wanting to do some market research opt for online surveys. These increase productivity by saving time as the data is instantly available and can easily be transferred into specialized statistical software or spreadsheets when more detailed analysis is needed.

iLearn is a web application that is available on all platforms. Our rationale behind iLearn was to create a 'human learning' system where the image recognition is completely human cognition based. For that purpose, online surveys will be hosted on the website that has over thousands of images and various categories gathered from the image scraper. To make the project more useful, trend analysis and search expansion were also included. These two features will help analyze user's responses and scrape better images for the survey questions.

1.2 Overview of iLearn

iLearn is a simple application where various surveys can be hosted and enormous amounts of data can be collected for research purpose. It is an image survey platform designed to get data from humans rather than machines.

1.3 Goals and Objectives

The goal of this project is to develop an application with an image scraper and trend analysis. Our goal is further divided into the following objectives:

(1) Host surveys with various categories.

(2) Gather user responses and gain useful insights.

(3) Keep perfecting the system with the huge amount of data gathered.

1.4 Scope and Limitations

Having to collect the related data by ourselves and guaranteeing the privacy of users, are scope and limitations as shown below:

(1) Make an online image scraper using different platforms that have over thousands of images.

(2) Create a database to store the survey responses.

(3) Create web APIs to let the application interact with the database.

(4) Create a web application to host the surveys.

Chapter 2: Literature Review

This chapter explains an overview of image scraping. It also includes a review of an academic research related to this project.

2.1 Image Scraping

Image scraping is a subset of web scraping, which is a data scraping technique in which data are extracted from websites through a computer program. As seen in Figure 2.1, a web scraper will navigate through websites and extract data. These data will then be processed and returned as useful information. The type and amount of extracted data depends on the designer of the scraper. In the case of this project, the extracted data will be related to image. The information gained from the scraper can then be used for various purposes such as image recognition, image tagging and product design.

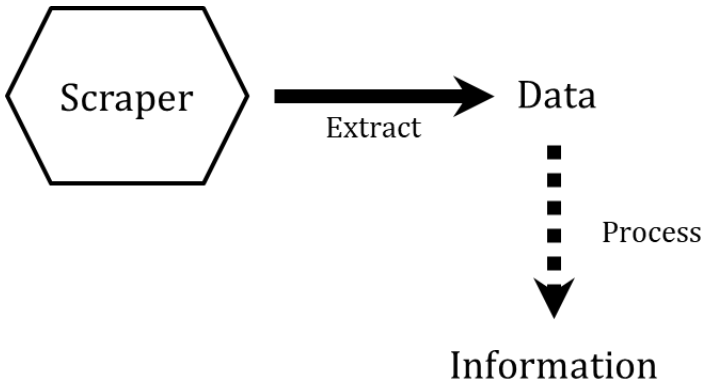

Figure 2.1: Web Scraping

2.2 Related Works

In this part, we will research existing systems and analyze how other people solve this problem or problem of a similar nature.

2.2.1 A web scraping methodology for bypassing twitter API restrictions

This research paper introduces a web scraper solution to extract information from websites. It aims to gather proper information for training and testing data science algorithms to obtain useful results. Fields like Natural Language Processing and Machine Learning use online social media platforms to retrieve user information and transform it into machine-readable inputs, which are used by various algorithms to obtain predictive outputs. For this project, Twitter was used because of its usability and widespread. Its engines may dispatch approximately 1 billion of user-generated content per month. Their approach is implementing a web scraping methodology for crawling and parsing tweets bypassing Twitter API restriction taking advantage of public search endpoints, such that, given a query with optional parameters and set of HTTP headers we can request an advanced search going deeper in collecting data. The first part is the web scraping solution which uses Scrapy, an open source and collaborative framework for extracting data from websites written in Python. This enhances the power of scraping engines to obtain an unlimited volume of tweets bypassing date ranges limitations. The second part is deploying and deaminizing the scraper. This paper made use of Graphical User Interface for testing the scraping approach. Specifically, it developed a web service with Django, a framework for developing web applications using Python as base language, which includes a model-view-controller design for quickly project escalation. Responses from the Scrapy engine are retrieved as instances from a Scrapy class named Items, but they can be transformed into comma-separated values or plain text files. The last part is to evaluate the performance of the scraper. The set of metrics are *total amount of time* for retrieving blocks of tweets, and the *volume of tweets* for a query q and the maximum number of historical tweets given a range of dates. Finally, it was concluded that by using Python technologies such as Scrapy, Django and customized daemons, it is possible to develop and escalate a web interface for launching, controlling, and retrieving information from web crawlers [1].

Chapter 3: Project Framework and Methodology

In this chapter, we will go through the details for the frameworks and algorithms that we used to implement our application and scraping.

3.1 Image Scraper

An image scraper uses keywords to search for images on websites such as Pinterest, Twitter, Reddit and Flickr. As seen in Figure 3.1, the image scraper will take in a keyword as its input. This inputted keyword will then be used as a search term for a website's search engine so that data related to the keyword can be extracted and processed into information.

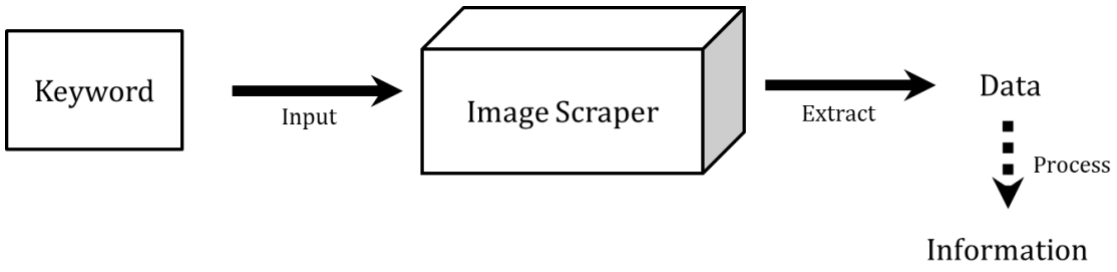

Figure 3.1: Image Scraper

As seen in Figure 3.2, the image scraper is divided into five parts. Input keyword, where we manually input the keyword related to the image we want to scrape. Navigate webpages, where the scraper navigates through the webpages. Extract web data, where the scraper extracts data from each of the webpages. Process web data, where the system goes through the extracted data and process it, keeping only the necessary data. Lastly, image data, where the processed data is inserted into the database. These stored data can then be used for any intended purposes.

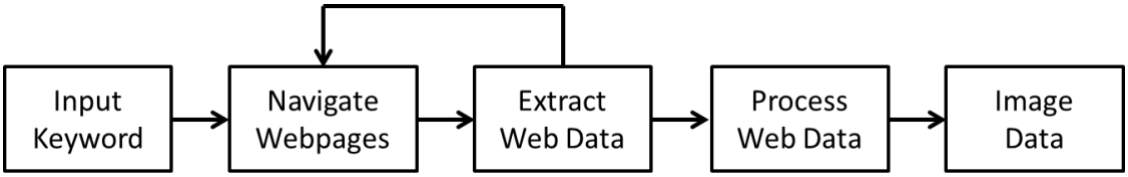

Figure 3.2: Image Scraper Workflow

3.1.1 Input Keyword

The keyword that will be used as the search term is the most vital part of the image scraper. For example, if we were to search for apple images, we should use “apple fruit” rather than just “apple” as our keyword considering how “apple” could refer Apple Inc., the technology company, and its products such as smartphones and tablets. Thus, the keyword we choose should be as connected to what we are searching for as possible.

3.1.2 Navigate Webpages

As seen in Figure 3.3, during this process, the image scraper will navigate through the pages of a website using Selenium, a tool for browser automation, and ChromeDriver, a tool to control Chrome. It will save the subpages where the image is located before going through them one by one during the next step.

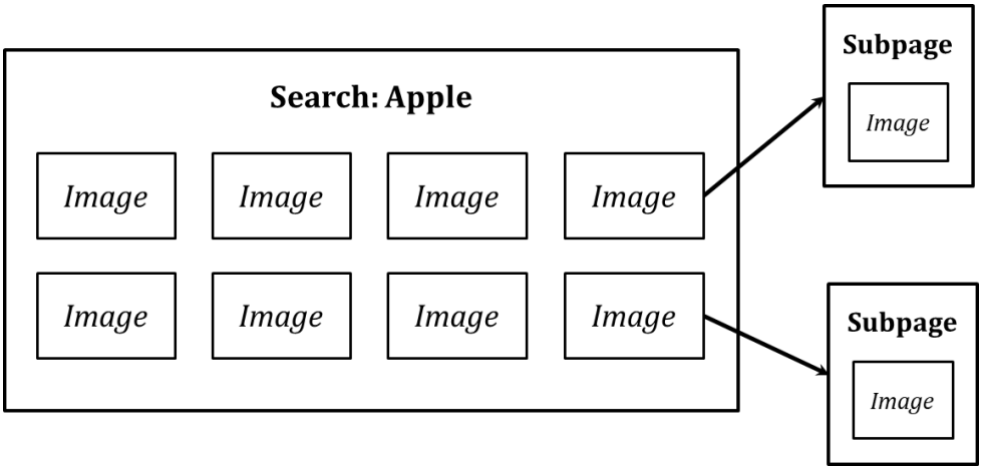

Figure 3.3: Navigating Webpages

3.1.3 Extract and Process Web Data

The data from each subpage will be messy. With the help of BeautifulSoup, a Python library for pulling data out of HTML and XML files, we can look through the messy HTML data and extract the data we need. The data that we will extract will be the link to the image and the keyword we used to search for the image.

3.2 Design Processes and Research

Before deciding the functionalities of our app and the UI, we conducted small research online. Through this research we were able to decide the applications features.

A good online survey provides clear, reliable, actionable insight to inform your decision-making. The most famous online survey platform is SurveyMonkey because of how easy it is to use. You can create a simple survey in minutes, then share the link or embed it directly on your site. Functionality is limited on the free version, though, as it only allows for 10 questions and 100 responses per survey [2]. Though our functionalities are not completely similar to SurveyMonkey, we wanted our user experience to be as easy as theirs. So, we made an image chooser question which is a choice based, easy to understand question type, that prompts respondents to pick the best image(s) that suit their answer. All the features of the application are discussed in the next section.

Chapter 4: Features of the application

4.1 System Architecture

As seen in Figure 4.1, the design of the technology is separated into two major parts: the user side (application) and the system side (service).

At the user side, the user visits the website and their google account is authenticated using firebase authentication. Upon visiting one of the survey categories to answer the survey, the API then retrieves the images stored in the database for that category and displays it to the user. Upon clicking submit after choosing image(s), the answer is sent back to the API and stored in the database to be used in the search expansion.

At the system side, an image scraper is used to collect and process image data through the inputted keyword. These image data are then stored in a table that is a part of the database. The database is MongoDB, a NoSQL and document-oriented database. Finally, the web API allows the web applications to interact and retrieve the stored data.

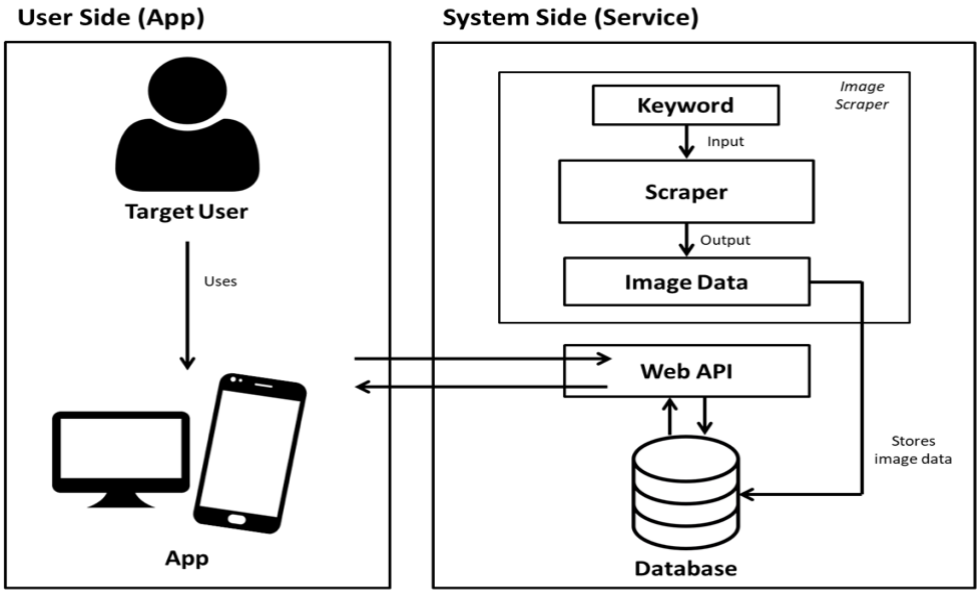

Figure 4.1: Project Workflow

4.2 Functionalities

4.2.1 Home Screen

The first screen user will see is the Home Screen as shown in Figure 4.2. This screen is just to let the user know about the application and what it does.

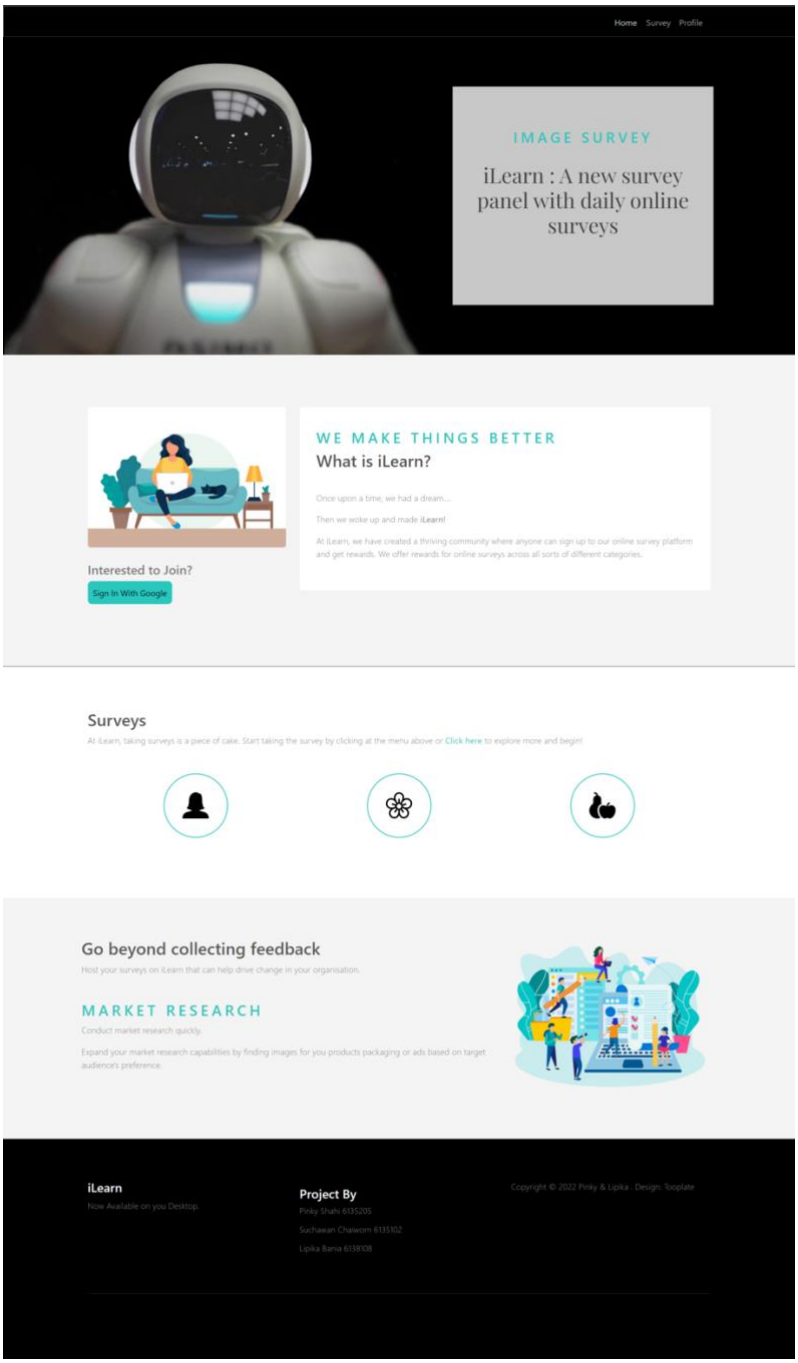

Figure 4.2: Home Screen

4.2.2 Survey Categories Screen

This is the 2nd screen of our application Where many different categories of surveys will be displayed for users to choose from.

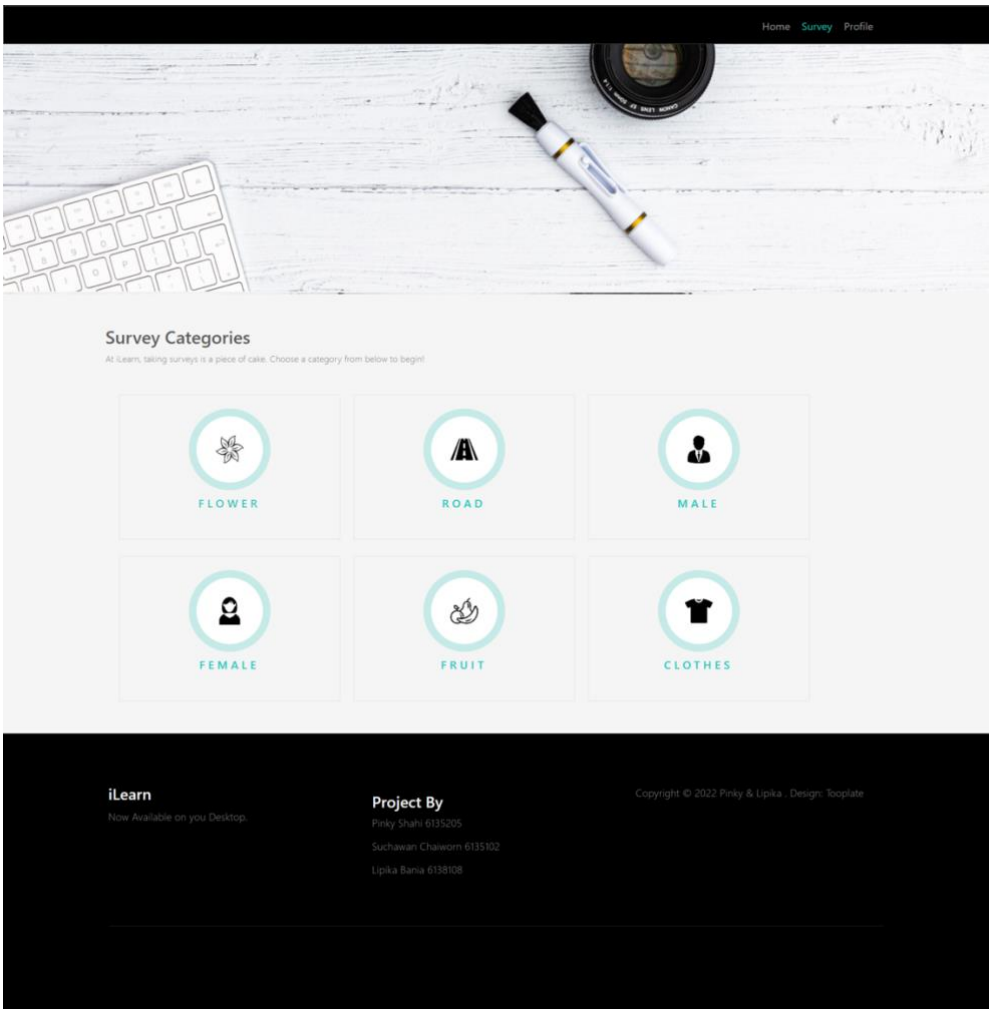

Figure 4.3: Survey Category Screen

4.2.3 Survey Question Screen

Once the user chooses a category from the previous screen, they can finally answer some surveys. If the user has signed up then they will be able to see the survey question and the images as shown in Figure 4.4. 6 images will be displayed at random from the database and user can select 0-6 images as they wish. Once they click the *Submit* button, their response will

be sent to the API and another 6 images will be displayed for that category. This same process will keep going on until the user clicks *Exit Survey* button to end their survey. However, if the user did not sign up for the application, an alert message will be prompted asking the user to sign up as shown in Figure 4.5.

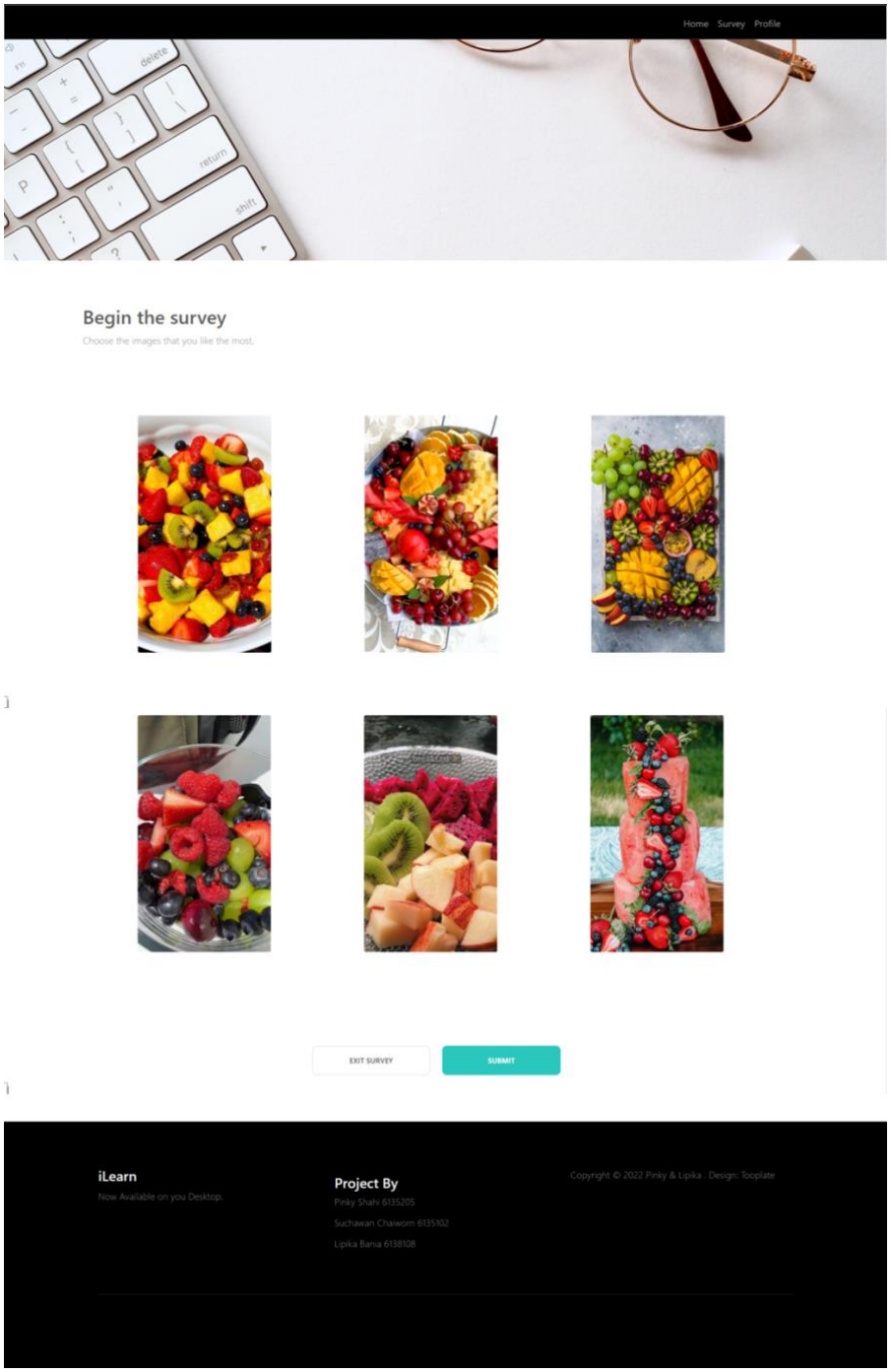

Figure 4.4: Survey Question Screen

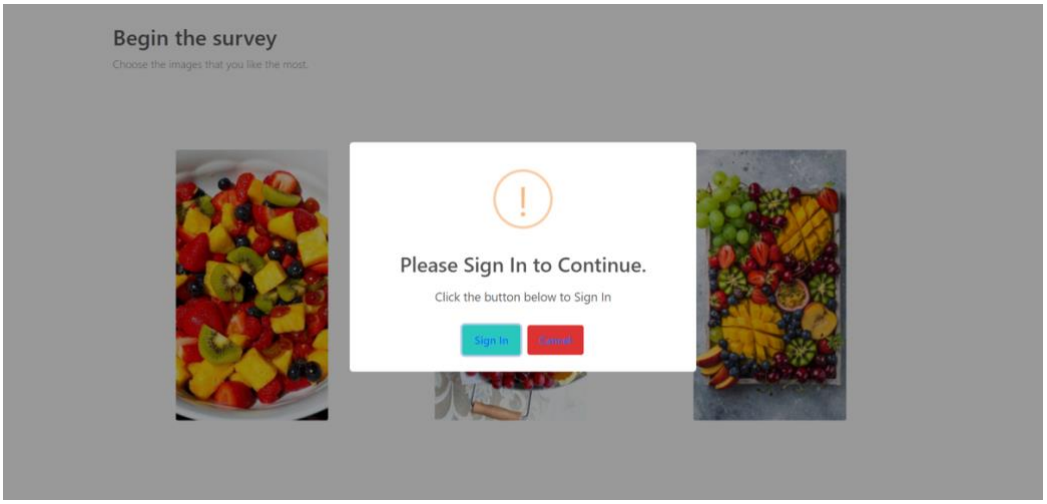

Figure 4.5: Alert Message

4.2.4 Profile Screen

The last screen of the application is the user profile screen which is for the user to sign out from the application.

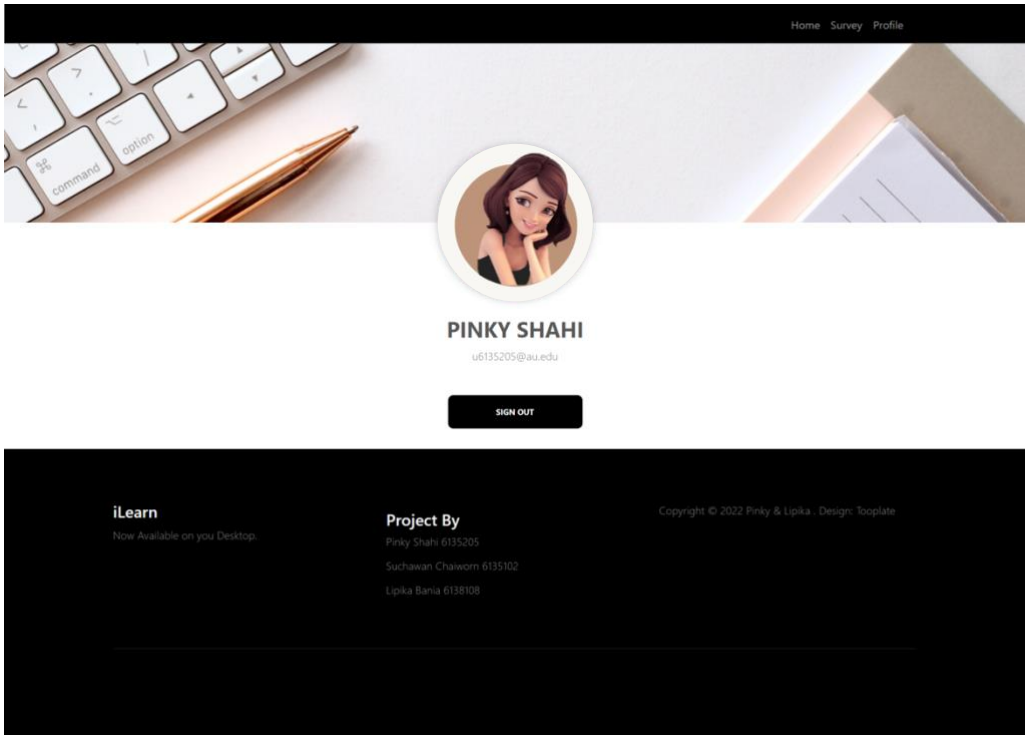

Figure 4.6: Profile Screen

Chapter 5: Preliminary Test

5.1 Survey Evaluation

5.1.1 Trend Analysis

We analyze how well each keyword has performed by comparing the original keyword and the user's answer from the last thirty days. If the accuracy of certain keywords is below a set percentage, then we will have to expand the search on said keywords.

5.1.2 Search Expansion

A number of keywords related to the keyword of an image will be prepared. These keywords will then be used as an associated keyword for the image scraper.

As seen in Figure 5.1, if the accuracy of images with keyword "cat" drops below the set percentage, keywords related to "cat" such as "pet", "cute", "tabby" and "feline" can be used as associated keyword for the image scraping.

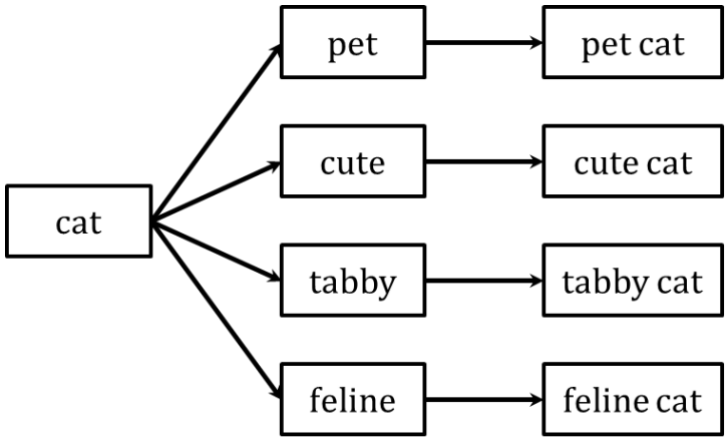

Figure 5.1: Search Expansion

Chapter 6: Conclusion

In conclusion, in this project, we have focused on developing a web application in a form of an image survey in which the images used were gathered by an image scraper.

The image scraper worked as intended. It was able to extract a great number of image data and insert them into the database. We were also able to analyse the trend and discover which image tags are not performing well. The search expansion can be further worked on, for example, we can do image recognition on the existing data using Keras models and the ImageNet dataset to find out more related keywords.

For the Web application, the features we incorporated in our app are good and work well, except for a few aspects that could be improved, rectified, and made more responsive in the future.

References

[1] A. Hernandez-Suarez, G. Sanchez-Perez, K. Toscano-Medina, V. Martinez-Hernandez, V. Sanchez, H. Perez-Meana. 2019. A Web Scraping Methodology for Bypassing Twitter API Restrictions.

[2] Tevin Shirey. 2020. Retrieved March 15, 2022, from https://www.webfx.com/blog/internet/11-free-online-survey-tools-compared/