Title

Survey Engine

Course

CSC-17B

Section

40502

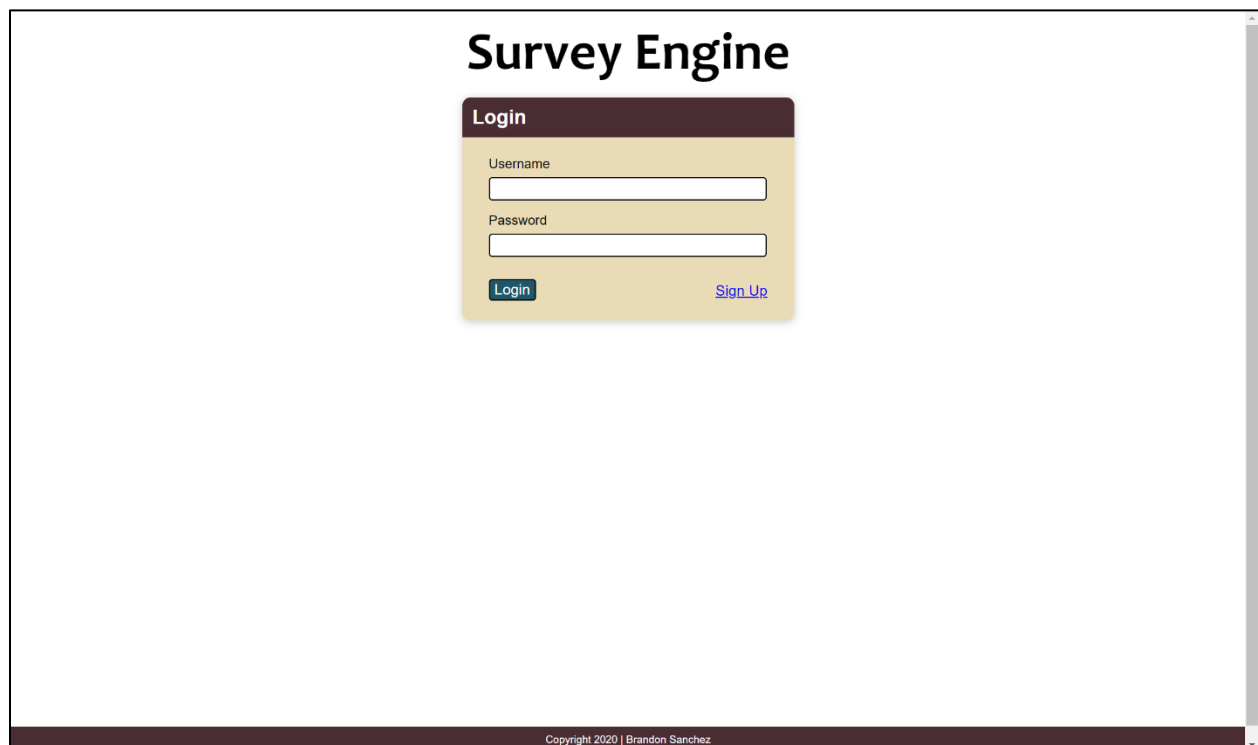Date

June 19th, 2020

Author

Brandon Sanchez

Introduction

This survey engine web program allows users to upload their surveys for others to take. Whoever uploaded their survey can see the number of responses and the percentage of total responses. Javascript is used to allow users to change static text to input forms so owners can edit their inventory. PHP/MySQL is utilized to store login information, surveys, and survey statistics.

How to Run

If utilizing the serverside database, everything should run fine with the MySQL and Apache servers turned on in Xampp. If utilizing a local database, run the surveyengine.sql statement in your GUI of choice and change the connection information in dbh_inc.php to the correct information.

Sample IO

Upon entry to the website, the user is greeted with a login screen. If they have not made an account prior, they can click the sign-up button the create their account.
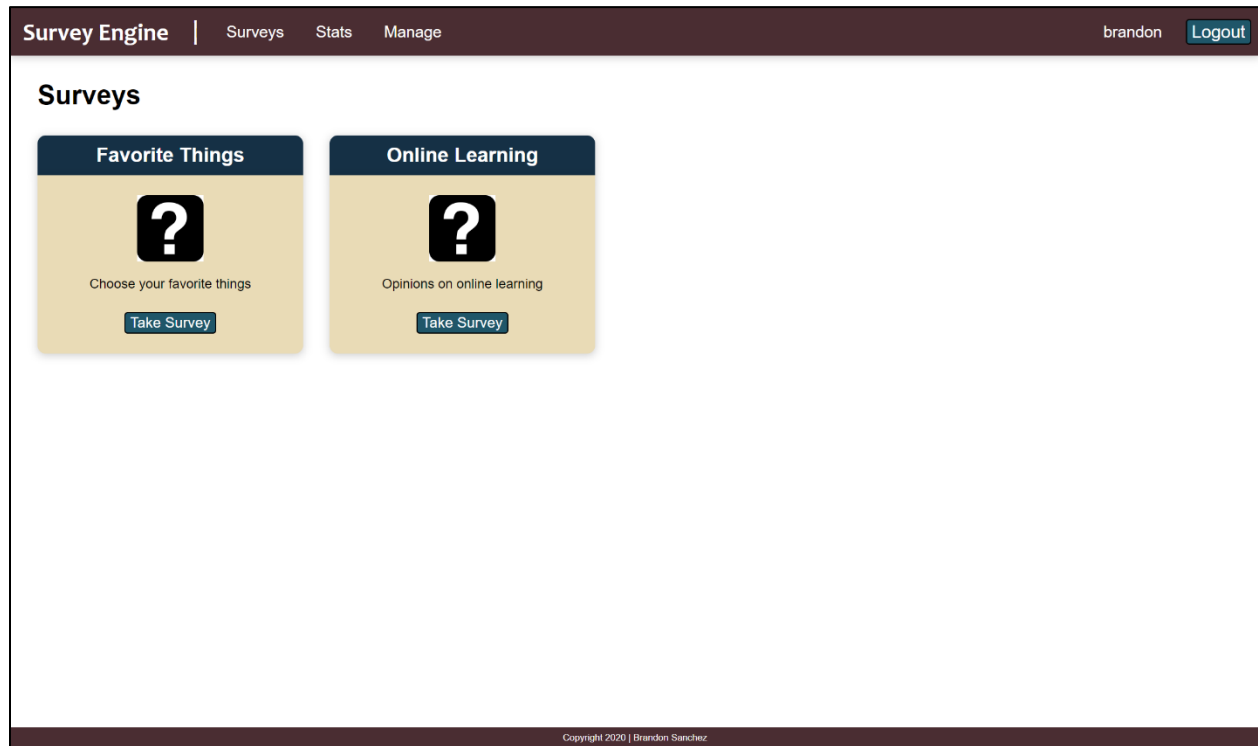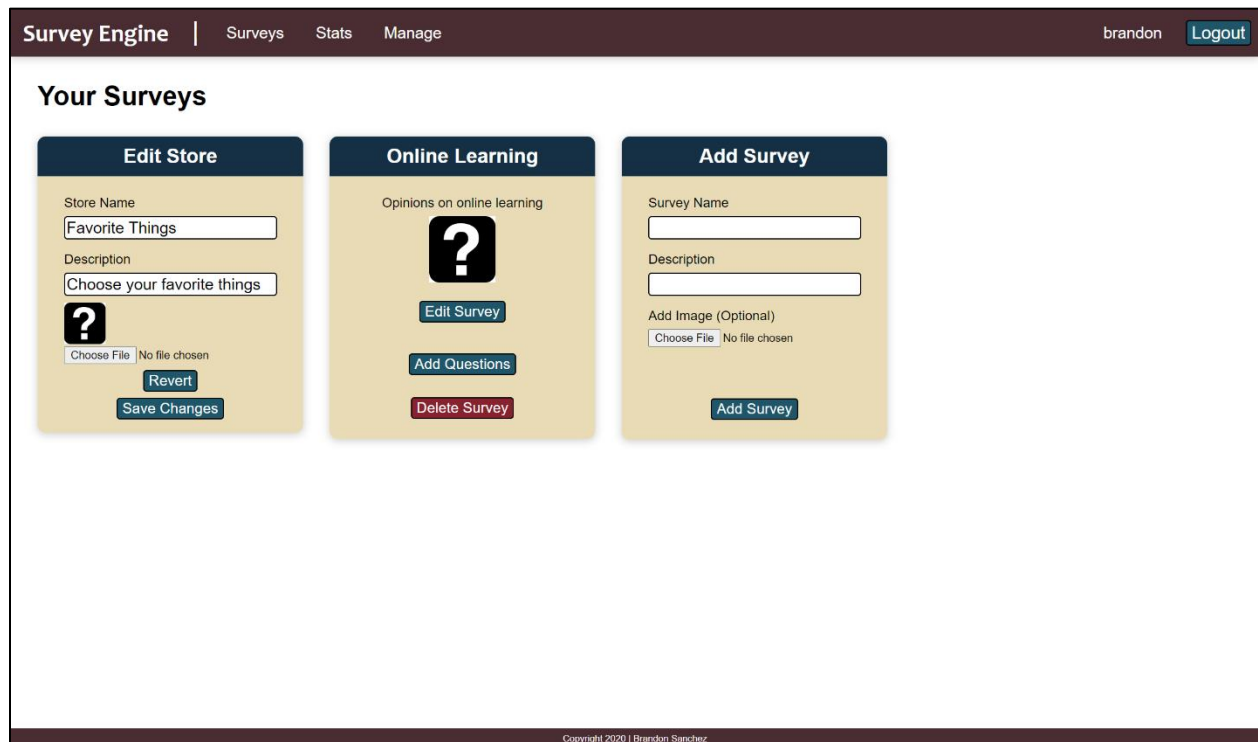

Once the user logs in, they are greeted with all the surveys uploaded to the database by users. From now on, the header is the same for all the pages. Their username and the logout button are displayed in the top right.
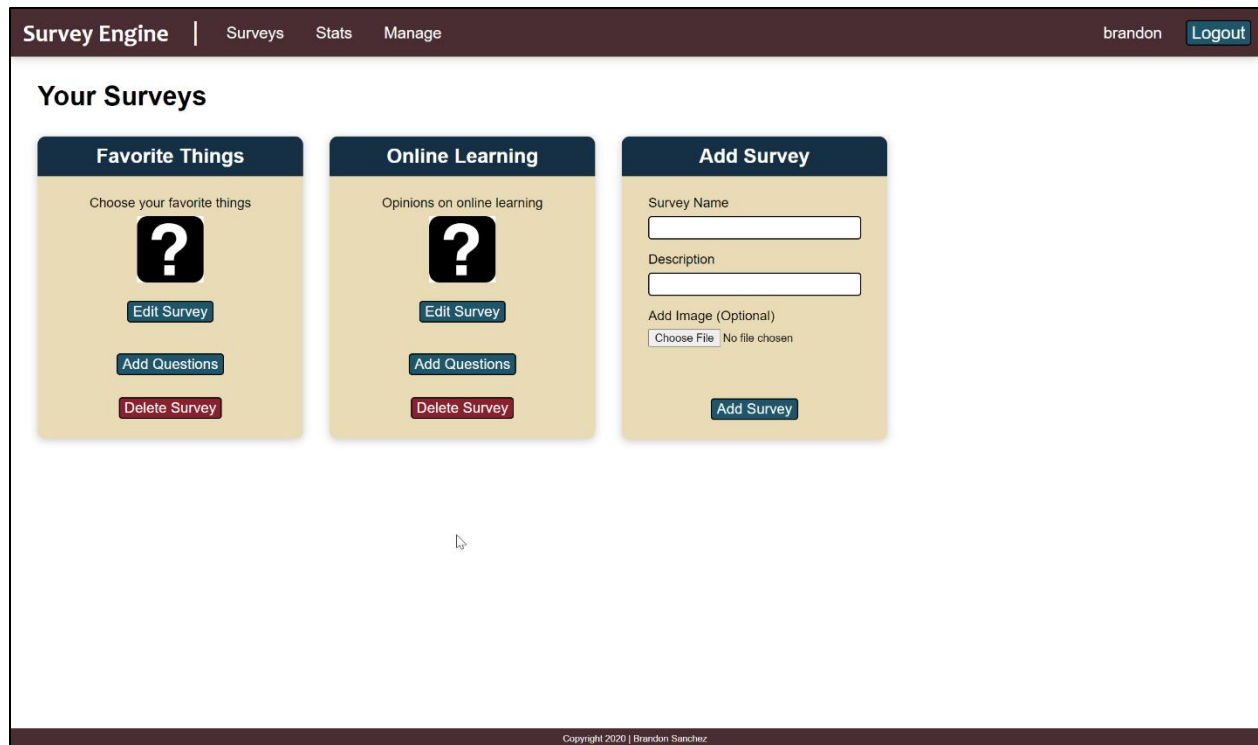
If the user selects manage at the top, they are taken to a list of their surveys with the ability to add more surveys or edit their existing surveys.

If the user presses edit survey, then they can edit anything about their survey entry.

If the user presses add questions on their survey card, then they can view all the questions currently in their survey and add more questions.

If the user presses add responses on their question card, then they can view all the responses to that question and add more responses.

If the user clicks the 'take survey' button on the surveys page, they will be greeted with the survey in this format. The user must select an answer for each question. Once done, they press submit. They will be taken back to the surveys page with a thank you message.

Your Surveys

Favorite Things

Choose your favorite things
Number of Responses: 1

View Stats

Online Learning

Opinions on online learning
Number of Responses: 0
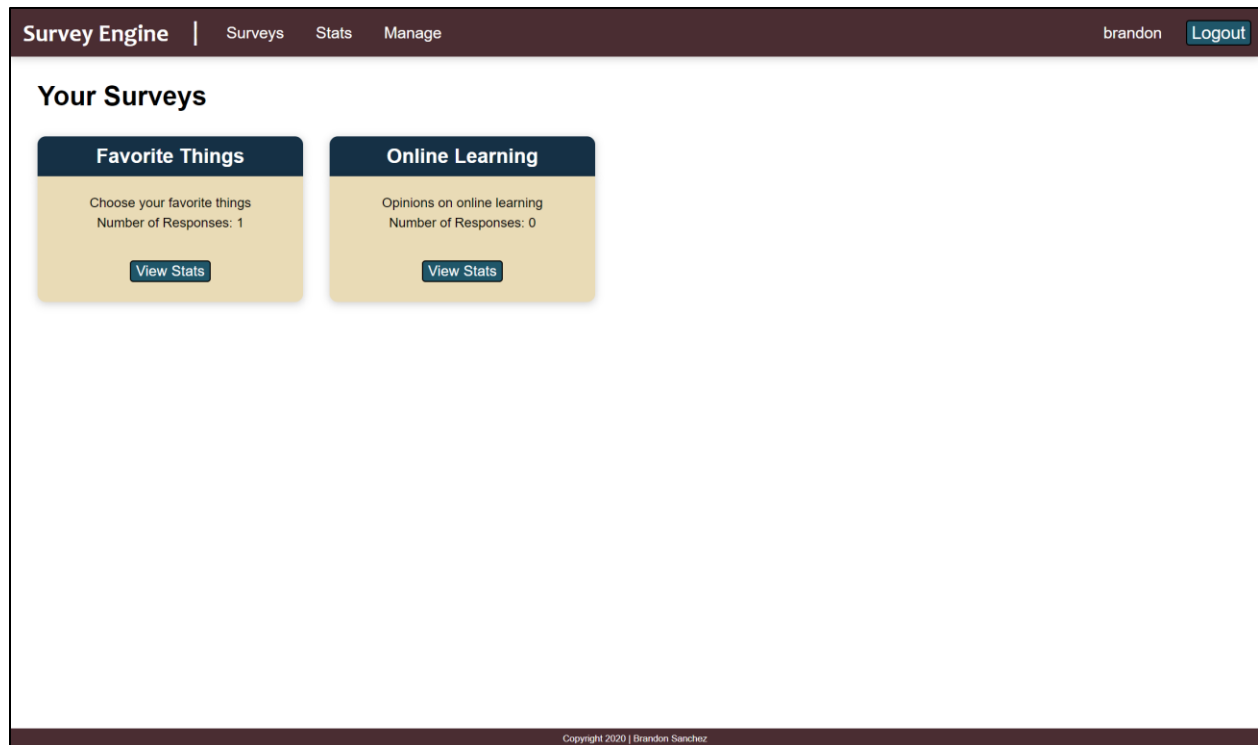
View Stats

If the user creates a survey, they can view the stats by clicking the stats link in the header.

Survey

What is your favorite color?
Red (Frequency = 0, 0.00%)
Yellow (Frequency = 0, 0.00%)
Green (Frequency = 0, 0.00%)
Blue (Frequency = 1, 100.00%)
Orange (Frequency = 0, 0.00%)

What is your favorite food?
Italian (Frequency = 0, 0.00%)
Mexican (Frequency = 1, 100.00%)
Chinese (Frequency = 0, 0.00%)
American (Frequency = 0, 0.00%)

What is your favorite car brand?
Chevrolet (Frequency = 0, 0.00%)
Toyota (Frequency = 0, 0.00%)
Honda (Frequency = 1, 100.00%)
Ford (Frequency = 0, 0.00%)

If the user presses 'view stats' on their survey, they can see the frequency of each answer and the percentage relative to the total number of survey responses.

Sample IO

1. **MVC**

   The controller in this case is all the include .php files that manipulate the database and send it to the web pages. The view is all the normal .php files that show things on the screen. The model is the MySQL databases.

2. **Objects**

   Each include.php is stored in their own file to help compartmentalize the code through objects.

3. **Storage**

   When a user uploads an image of their store or items, it is stored on the server in a file called uploads and the directory name is stored in the database. The code for this is found in addstore_inc.php and additem_inc.php.

4. **SQL**

   Almost all the code in the includes file uses php to connect to the database. The relational chart for the database is at the bottom of this document.

5. **Form Validation**

   The login page uses form validation so users can only enter letters and numbers in for their username. The regular expression can be found at signup_inc.php line 16.

6. **Login**

The login system code can be found in index.php and signup.php for the actual visual web pages. For working with the database, it can be found in login_inc.php, logout_inc.php, and signup_inc.php.

7. **Sessions**

   Sessions are started in header.php which is attached to the beginning of every page. This stores the user id of the user logged in, their balance, their cart total, and many more things.

Things I Learned

I completed this project after the storefront program, so I had some experience working on a website. Due to the similarity of the two programs, I copied what I had from the storefront program. This was a really good time saver, but it led to some issues. The manage view is not suited for surveys. Also, for the sake of time, I left many of the variable and file names the same so they do not make sense in regards to the survey engine.

Entity Relationship Diagram

`bsanchez_se_users`

`user_id` int(11) NOT NULL AUTO_INCREMENT

`username` varchar(45) NOT NULL

`password` varchar(1000) NOT NULL

`bsanchez_se_surveys`

`survey_id` int(11) NOT NULL AUTO_INCREMENT

`user_id` int(11) NOT NULL

`name` varchar(45) NOT NULL

`description` varchar(45) NOT NULL

`img_dest` varchar(100) NOT NULL

`num_responses` int(11) NOT NULL DEFAULT 0

`bsanchez_se_responses`

`response_id` int(11) NOT NULL AUTO_INCREMEN

`question_id` int(11) NOT NULL

`response` varchar(100) NOT NULL

`num_responses` int(11) NOT NULL DEFAULT 0

`bsanchez_se_questions`

`question_id` int(11) NOT NULL AUTO_INCREMENT

`survey_id` int(11) NOT NULL

`question` varchar(100) NOT NULL