

Omar Alkendi

M. Lehr, PhD

CIS/CSC-17B

13 December, 2019

Individual Project: Survey Engine (Kendi Survey)

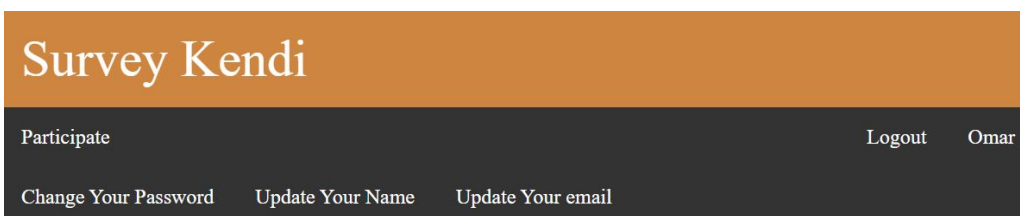
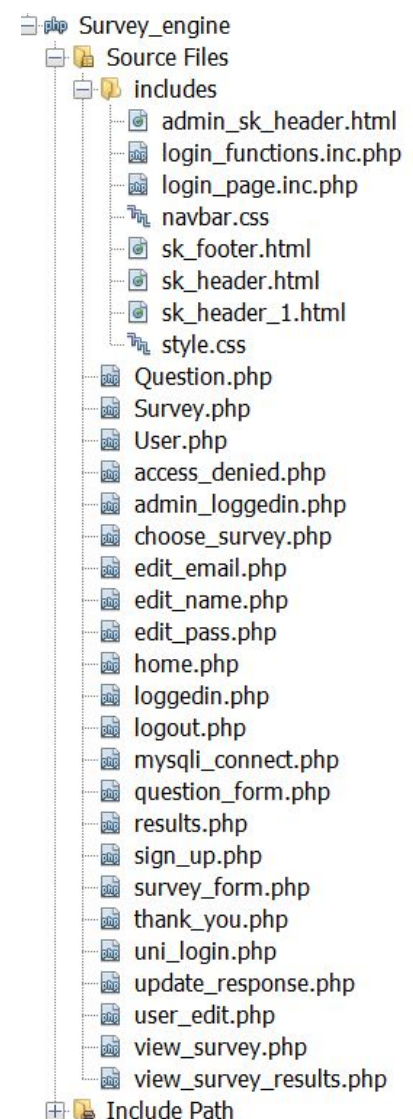
Introduction:

Kendi Survey is a messy yet somewhat functional survey engine that allows the admin to create a survey and add questions to it. A user can then participate in any survey—more than one time, unfortunately—by simply logging-in and selectin a survey.

Kendi Survey takes care of calculating the results from the surveyees responses and returns the total responses for each answer.

MVC:

Kendi Survey lacks an MVC design, but features numerous PHP pages that serve the purpose of modifying the view to the user. Since the survey incorporates multiple forms, not including the user login and updates, it was quite difficult to adopt an MVC design for the project. Nonetheless, the presence of headers and footers on every page will make the transition between different pages quite smooth.



Objects:

There are a total of 3 PHP classes in the project.

Question	Survey	User
+ qstn: string + cat: string + ans: string*	+ title: string + desc: string + qstn: Question*	+ user_id: int + first_name: string + last_name: string + email: string + password: string
+ set_cat(string): void + set_qstn(string): void + add_ans(string): void + get_cat(): string + get_qstn(): string + get_ans(int): string + display(): void + getDB(): dbc + insert_qstn(): void + insert_into_survey(): void + insert_ans	+ set_title(string): void + set_desc(string): void + add_qstn(Question): void + get_title(): string + get_desc(): string + get_qstn(int): Question + display(): void + getDB(): dbc + create_survey(): void + take_survey(): void + update_results(): void + show_results(): void	+ create_user(fn, ln, e, p, dbc): void + check_pass(user_id, pass, dbc): bool + check_email(user_id, email, dbc): bool + update_name(fn, ln, user_id, dbc): void + update_pass(user_id, np, dbc): void + update_email(user_id, ne, dbc): void + getDB(): dbc

PHP Classes:

1. User.php: A class that assists the user in creating an account to save the user's score. It also allows the user to change their name, password, and email address. Moreover, it can check the login credentials to determine if a user with the entered credentials exists.
2. Question.php: The question class allows the admin to create a multiple choice radio-buttons questions with at least two answer. To add the question to the survey, the survey must be initialized in advance.
3. Survey.php: The survey class initializes a survey for questions to be added to later on.

Note: While the classes are accessed directly, their use is limited to the methods (functions) that they have.

Reading/Writing Files, Local Storage, and JSON:

NA: No concept under this category was used in making Survey Kendi

Databases and MySQL:

A MySQL database was utilized to hold three enums and one two tables:

1. enum_answer: Each answer is assigned a unique id to avoid repetition.
2. enum_question: Each question is assigned a unique id to avoid repetition.
3. enum_survey: Each survey is assigned a unique id to avoid repetition.
4. entity_user: Holds a record of all users, admins and non-admins.
5. survey: A cross reference between enum_answer, enum_questio, and enum_survey.

The screenshot shows the Survey Kendi web application. At the top, there is a header with the title "Survey Kendi" and navigation links: "Create Survey", "Add Questions", "Results", "Logout", and "Values". Below the header, the main content area displays a survey titled "Fav?". The survey consists of three questions, each with a list of options and their respective entry counts:

- Food?**
 - Pizza (5 Entries)
 - Salad (1 Entries)
 - Sushi (1 Entries)
 - Shawarma (1 Entries)
 - Taco (1 Entries)
- Color?**
 - Black (3 Entries)
 - Grey (4 Entries)
 - White (1 Entries)
 - NOV (1 Entries)
 - All (1 Entries)
- Animal?**
 - Cat (6 Entries)
 - Dog (1 Entries)
 - Horse (1 Entries)
 - Lion (1 Entries)
 - Snake (1 Entries)

Form Validation:

Form validation was implemented in the sign_up.php page to prevent the user from creating an account without the proper format for:

1. First Name: Must be in English letters only.
2. Last Name: Must be in English letters only.
3. Password: Must be 6~20 characters length, include one upper-case, one-case, and one special character.
4. Email: Must follow the general format for an email (eg. a@b.com)
5. All surveys must have a title and a description.
6. All questions must have a definition, a category, and at least two answers

User-Admin Login:

The uni_login.php is a login page for both users and admins, and will redirect each of the to their proper page. Admins can create and view the results of surveys. Users, on the other hand, can only participate.

Cookies, Sessions, and Securing Pages:

Sessions are used on the top of every single page to determine whether or not the user has logged-out. Hence providing the proper header element and the correct redirections.

Code for Survey Kendi are available [here](#).

