**INST377 Final Report**

**Title**

Food for Terps

**Team Members**

Alex Acuna, Alicia Curcuru, Christina Hall, Emily Hwang, Lydia Spurrier

**VCL account/URL to AWS Server**

Folder with all our files

* <http://ec2-34-230-66-70.compute-1.amazonaws.com/foodforterps/>
* Username: root
* Password: spurrierly

**GitHub**

* <https://github.com/spurrierly/FoodForTerps>

**Information Problem**

The information problem we are trying to solve is suggesting restaurants to those within the College Park area as there is a large variety that people can choose from but they might not know what is offered or where they should eat based on their dietary and monetary restrictions. While some services, like Yelp and Google, provide and suggest restaurants based on customer reviews and also filter depending on some preferences, they currently lack in providing filters for all dietary restrictions, which is the main problem and need we have discovered.

**Strategies/Solutions**

The solution to finding restaurants in only the College Park area that meet people’s criteria is to create a form that they are able to fill out with their preferences. From there, it checks their preferences to a database of local eateries and delivers an output of the suggested restaurants that meet their criteria.

**Rationales and Justifications (on system design and technology)**

We decided to structure our site as a user input form which would then return restaurant options based on the form data. Our form was designed with ease of use in mind, but also takes in account the possibility of user error or malicious attacks and does not allow for free text entry. Instead we utilize drop down boxes, radio buttons, and sliders. By only letting the users select buttons that can lead them to their answer the output and process should not break and always deliver an output. Also, for the accessibility and user-friendliness of the website, when a user’s mouse hovers over the dietary restriction options, the color of the options change color. With this friendly display, the user will have better experience of selecting the right option quickly.

**Final System solving the Problem**

Our final system solves the problem of finding restaurants in the College Park area based on a few popular deciding factors, including dietary needs, price point, cuisine type, or overall rating. By allowing the user to make choices based on these categories, our site provides them possible restaurants in the College Park area that meets the desired criteria. Possible use cases could include finding a restaurant that also has vegan options for your newly vegan friend, finding the cheapest but best rated bar, or trying a new cuisine genre. Also, as mentioned earlier, since the suggested restaurants are all in the College Park area, the users won’t have to worry about needing to, on their own, filter out the restaurants too far away from the area.

**Challenges (faced and their impact on the final design**)

Challenges confronted include incorporating the rating system into the html form, however, it was successfully added after researching online and through trial and error. We understood that we wanted a star rating system but did not know how to create one. In the long run, after all of the research, it boiled down to writing html for five buttons with labels and then going with it to create css that would reimage them to appear as stars. We considered to put half star ratings, simply because other sites do use that and it can be helpful. But to maintain our scope and not make the project too difficult, we stuck to the five buttons that allows us to use the rating scale like we wanted. Another challenge we had was having to find an API that had all the information we needed. We weren’t able to find an API within such a short period of time so we just created a sql file that has some list of restaurants and its data. Since not all restaurants in the College Park area is listed, the users will be restricted and will only be recommended to the restaurants in our database.

**Future Work**

Future work includes incorporating a larger list of restaurants to be seen in the results after a user inputs their information. This could be done either through finding an API that fulfills all the requirements or if time allows then manually inputting all the data into the sql file we have made. Furthermore, the results would also include location, hours, whether they take reservations or not, review comments, and other extra information that would be helpful in reviewing a restaurant.