### a cs350 rudiment

# Super Search 150xp

There are two options for this rudiment. You can do Super Search which is described here or you can do a project of your own design. The requirements are:

1. Your app has a complete login/user system
   1. A page where you can create a new user account
   2. A login feature
   3. A logout feature
2. A person should stay logged into the site until they logout or they close the browser.
3. Your app should have a search feature which is dependent on who is logged in. For example, in super search when a user searches for a store the results are dependent on the logged in user’s zipcode.

If you are implementing your own app, run the idea by me.

## Super Search

See the youtube video:[React Super Search](https://youtu.be/u4wZosG841s).

You web app should implement similar features.

[The initial SQL File](https://raw.githubusercontent.com/zacharski/supersearchclient/master/supersearch.sql)

Here are the things you need to do and implement:

### 1. Database

The SQL file has some insert statements to help you get started. However, it doesn’t have code to create a database or any tables. You need to add those.

Here is one problem with the current design. The structure of the store table looks like

|  |  |  |  |
| --- | --- | --- | --- |
| name | type | address | etc…. |
| Duck Donuts | Coffee, donuts | 1223 Jefferson Davis Hwy |  |
| 25 30 Expresso | coffee | 400 Princess Anne |  |
| La Posta | Mexican | 2410 Calle De San Albino |  |
| Double Eagle | Steakhouse, American | 2355 Calle De Guadalupe |  |

Now we would like to have a store have multiple types. For example, Duck Donuts is listed on Yelp as donuts, coffee, breakfast and Double Eagle is listed as Steakhouse, American. I would like you to add this feature to our database and have at least 10 items that have multiple types.

**The database needs to be in 3NF**

### 2. General Search

You should implement a search feature that works similarly to the video. We should be able to search for:

1. The name of a store (i.e., Starbucks)
2. The type of a store (i.e., Mexican)
3. The name of a movie (i.e., Wonder Woman)
4. All the movies by typing movies.

### 3. Login

A login page is provided in the repository. You need to finish the implementation. Once a person logs in, it should be persistent across pages. Add a visual indicating who is logged in -- If Ann is logged in we should display ‘Ann” somewhere on the page.

### 4. Logout

There should be a working logout feature.

### 5. New User

You need to create an ‘add new user’ page for the website.

6. Security

The database user your code uses should have the minimal permissions necessary to complete its job. All passwords should be hashed.

7. Super Search

Once a person is logged in any search should be limited to their zipcode.For example, if my zipcode is 88005 and I search for Starbucks, I should only see results for 88005.

8. Minimal SQL Injection Protection

Make sure you can successfully search for Peet’s and Nobody’s Watching.

9. Running in the Cloud

Your application must be running on a cloud-based web server such as Google Cloud.

### demo

Please demo in class. Before you start the demo please hand me a 3x5 card with the title ‘Sessions’, your section number and your full name legibly printed:

Sessions

Ann Hedberg

Section 1