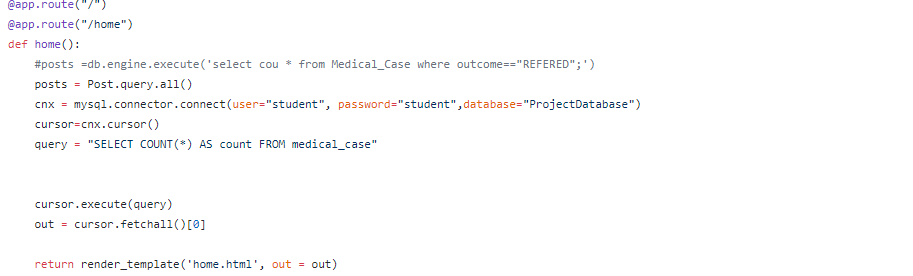
1. Include one query using Flask-SQLAlchemy filter or filter\_by.  
     
   **Implemented in the referral route to achieve the “Referals” page where all of the cases that have been refered out are listed (filtered from the Medical\_Cases table.**
2. Include at least one SELECT using an aggregate function.  Use regular SQL for this.



1. Include at least one SELECT using a compound condition using regular SQL, and also the equivalent of a compound select using Flask-SQLAlchemy.

We tried to add one of these queries but it did not show anything for the data that we have. However, we did do a join in #4 that is similar.

1. Include at least one JOIN query using SQL, and also one using Flask-SQLAlchemy.  
     
   **Implemented in the “cases” route by joining the Physician table, Medical\_Case table and Hospital table.**
2. Include at least one subquery.  Regular SQL.  Excellence points if you also use Flask-SQLAlchemy.

6. Use a form to collect user data, as shown in our CRUD labs

**We used the user and registration functionality from the CRUD labs**

7. Populate a field on a form or table from the database. This would most likely be for your update, and you can model this directly off of our examples in class

**Patient Form fills the patient\_id form with the next viable id. Case Form does the same with case id.**

**Case Form includes two select fields for existing patient ids and existing procedure ids**

8. Check for empty data fields. You can use the built-in validations for this

**Used built in validations.**

9. Implement referential intergrity. Demonstrate what happens when it is violated. Or, if you constructed your program so that it can't be violated, demonstrate how it references a primary key and prevents a vioaltion of referential intergrity. For example, if employee has a foreign key deptNo that references the primary key deptNo in the relation Department. If you populate a drop-down box with existing department numbers, this will prevent the user from entering an invalid department number, thereby enforcing referential integrity. We did something similar in Lab-4-c. If you just have a text box, the user can enter a department number that doesn't exist, and you can demonstrate that this will cause a referential integrity error. Either method is fine, but be clear on what you are trying to achieve and demonstrate.

**If you try to add a record with an existing id, it will not allow it.**