Proj5 Part 3: Error checking

I chose to do error checking in the database and additionally in the HTML. In HTML, I used minlength and maxlength to force a user to input the correct values. In the database, I added a CHECK constraint “9DIGIT” to the column SSN. This checks to make sure the SSN is greater than 99999999, forcing a 9-digit number. With the HTML, the application stops the user from entering an incorrect value by preventing access to the database until the value entered is proper. I believe the application is greatly benefitted from the HTML constraint, because it will not allow a user to add anything incorrect to the database, ensuring that the user cannot progress unless their data is entered correctly. Handling the error on the front-end seems to have a greater benefit due to the user seeing that there is a problem. If it was handled ONLY through the database, it may not return to the user the specific problem, leading to more questions. With an in-code or front-end error check, the user is properly informed on what the issue is so that they do not make the mistake. If I chose to do it in my Java application, the repository could contain an if statement that connects the user to a separate page giving them the error message if employee.SSN < 100000000. In mySQL, adding a check constraint will cause an error and display it on the database that there was a constraint error so the table could not be updated.