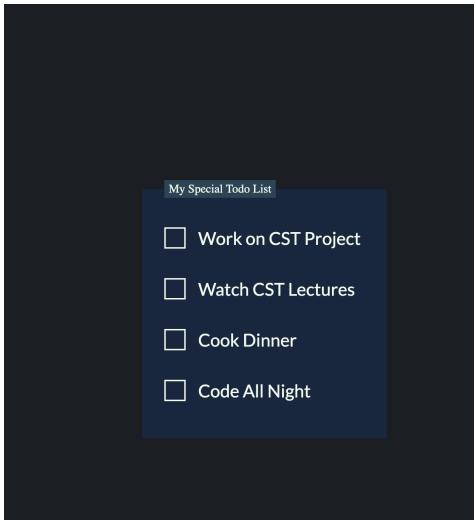
## Assignment 3: Persisting Your Todo List:

Recall the starter.zip file you downloaded for last week's Assignment. When you ran the application and visited localhost:3000, you would see something similar to below:



The problem with the data above is it is completely static data. This is evident when you go to index.ejs in the views/pages folder, where you will see this data hard-coded into the HTML.

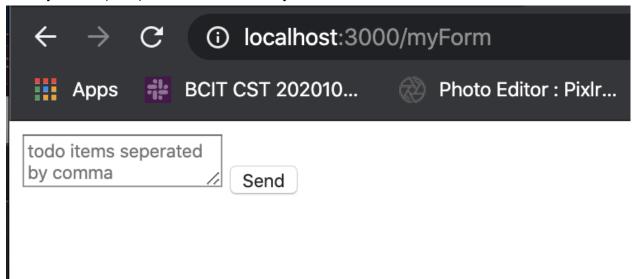
For this assignment, you must do the following:

Turn the static content from the list into a persisted Todo list (a todo list stored in the database). You may use MongoDB or MySQL to do so. Either is fine, the choice is yours.

You must ensure the following:

When you visit the main / route, it should show your Todo List in an empty state (since there shouldn't be any todos in the database).

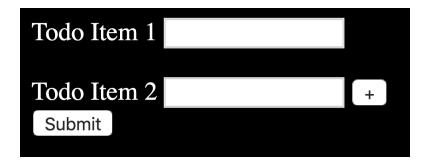
When you visit (GET) the localhost:3000/myForm URL:



Rather than seeing the textarea above, you must see the following:



Each time the user clicks the + button, a new input should be created with the label of Todo Item <Number> next to it. For example, if the user clicks the + button, it show now look like this:



When the user clicks send, you must insert all items the user typed into the database. After adding them to the database, the user should be sent back to their Todo list, and should be able to see the items they added.

On your main todo list page, you must also have the following:

2 inputs at the very top of the page that have a p tag above it saying: "Enter the todo item you want to modify, and the new name you want to give it".

A user is able to type the name of a todo item into the first input, and for the second input, they can type the new updated name they want to give that todo item. Below these inputs should be 2 buttons: **Update**, **Delete**.

If you press Update, it should take the Todo Item they typed in the first input, along with the new name they typed in the second input, and perform a **PUT** operation which should update the database with the newly updated name. Ensure that case-sensitivity is not on, meaning a user can type "my todo item" or "MY TODO ITEM" and in both cases, it will match "my todo item" within the database. Once the update is complete, you should be taken to your main todo list.

If you press Delete, it should look up the Todo item in the database that matches the name they typed in, and delete that Todo item. After deleting it, you should be taken back to your main todo list.

You may handle errors or cases in which the Todo Item doesn't exist in whatever way you feel is appropriate.

If you are using MongoDB, please use Mongo Atlas and include your connection string in your deliverable so we can test out your application. You may delete your database as soon as you receive your grade. Your code and connection string will not be shared with anyone other than myself or Bhupinder.

End of Assignment. Please submit the assignment to Learning Hub (D2L) when you have completed it.