Install starting files

Require mongoose

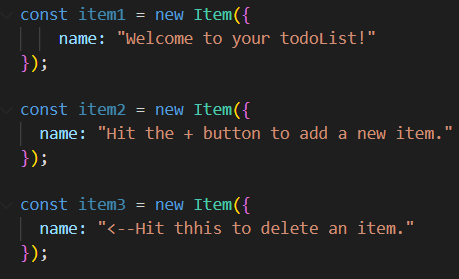


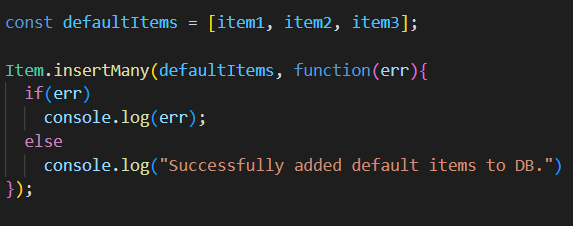
Delete the items and workItems array. We will use mongoose instead.



Now we need to pass our items array to list.ejs, because we deleted it. We want to pass our items that are inside our items collection.

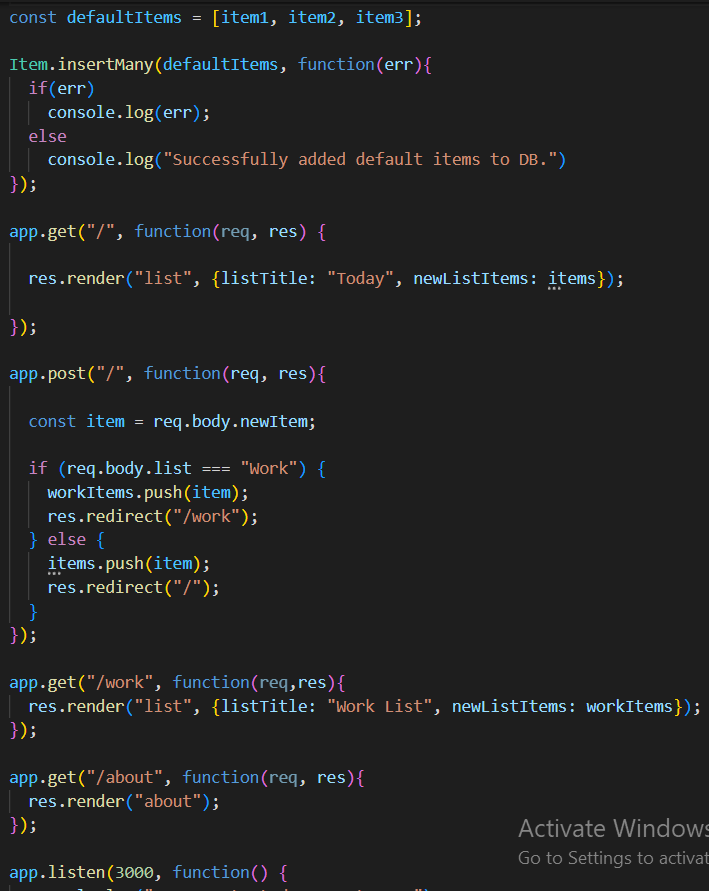
To do that, we need some default items.

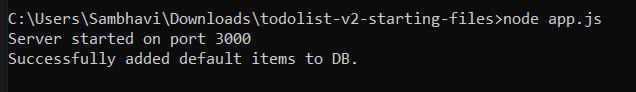


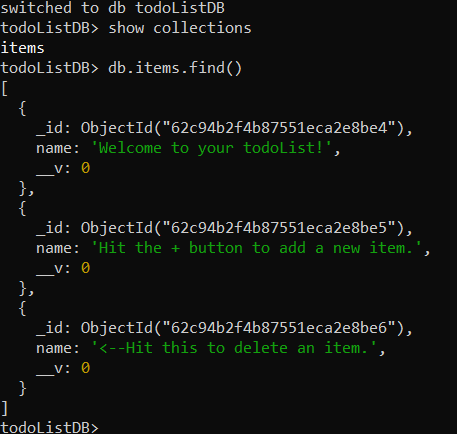


Complete code:





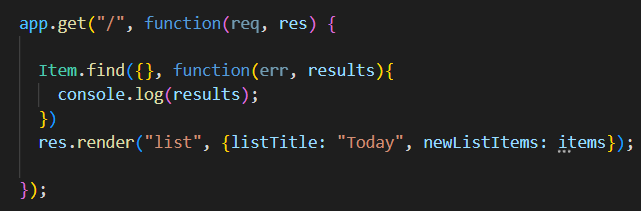


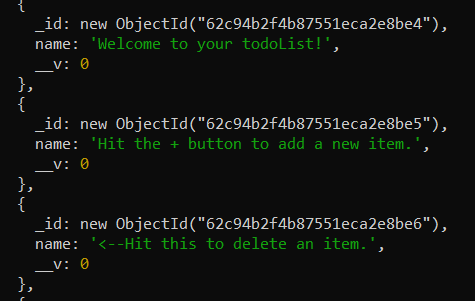


Rendering database Items to the todoList:

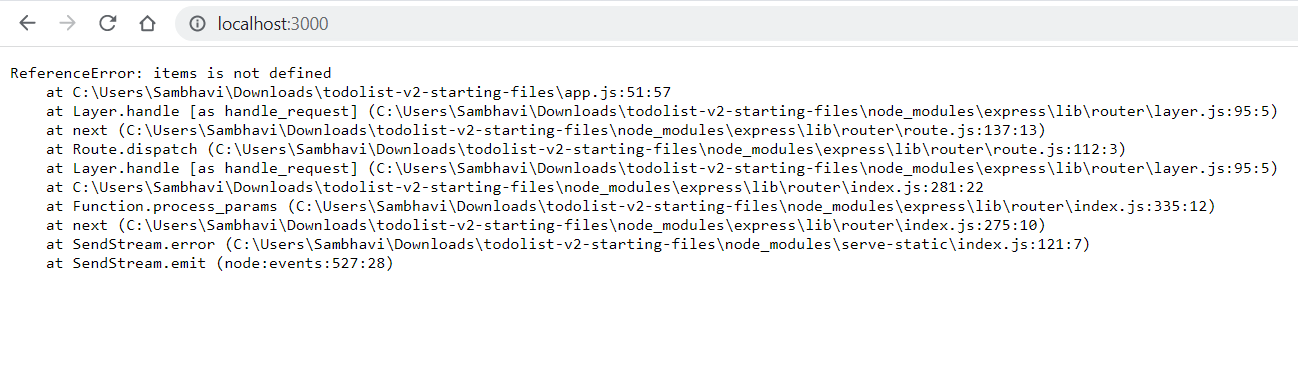
Like we did inside our database, how can we do db.fruits.find() inside our app.js, so that we can actually send it over to our list.ejs to get rendered in our todoList ?

Mongoose.find() method



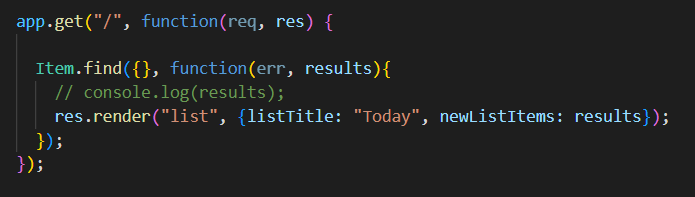


We do get our items logged, but our website is broken

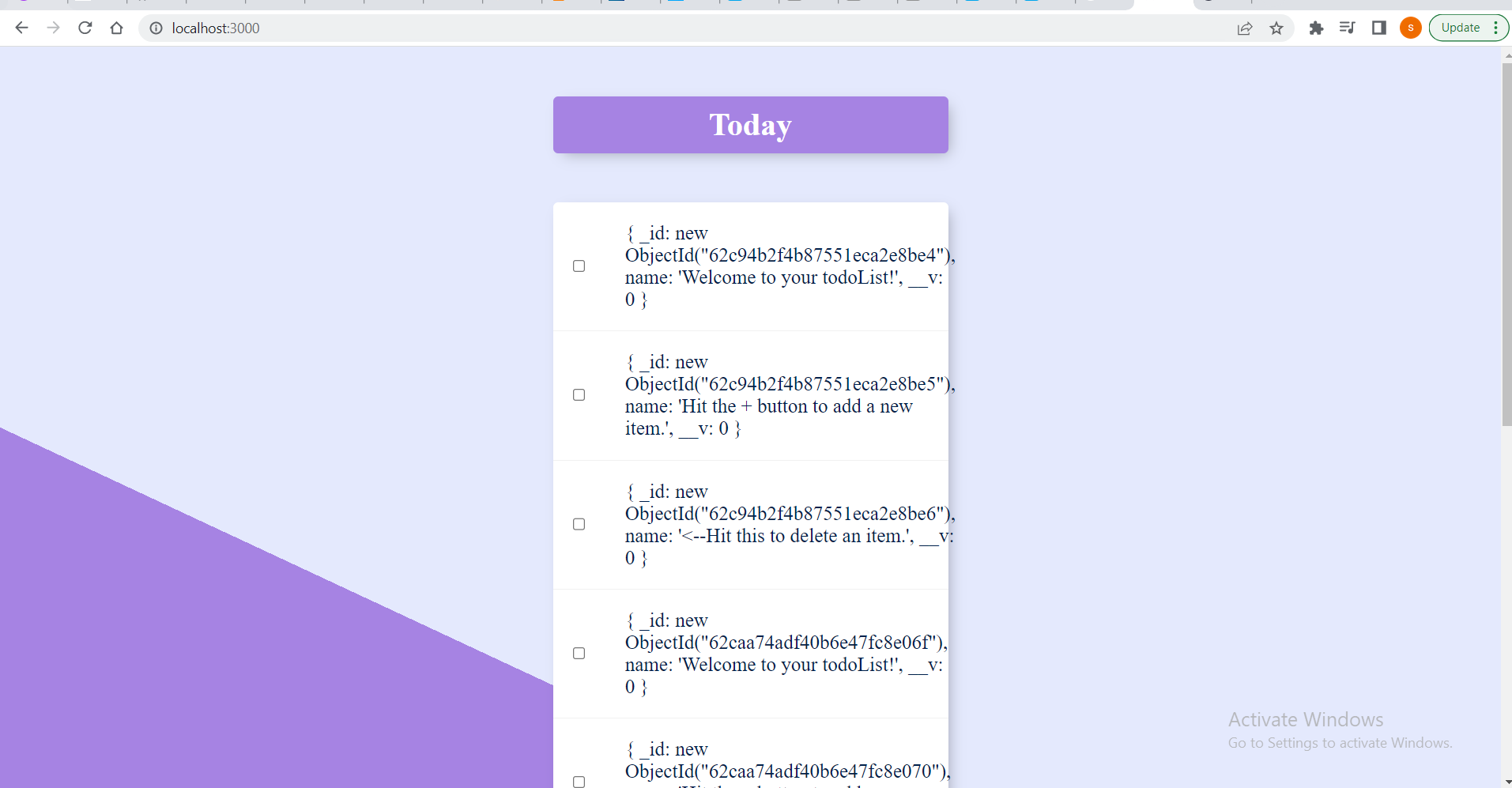


Becoz it could not find the items array.

So instead of console.log, lets pass it to our render method



So now, we don’t have any errors but

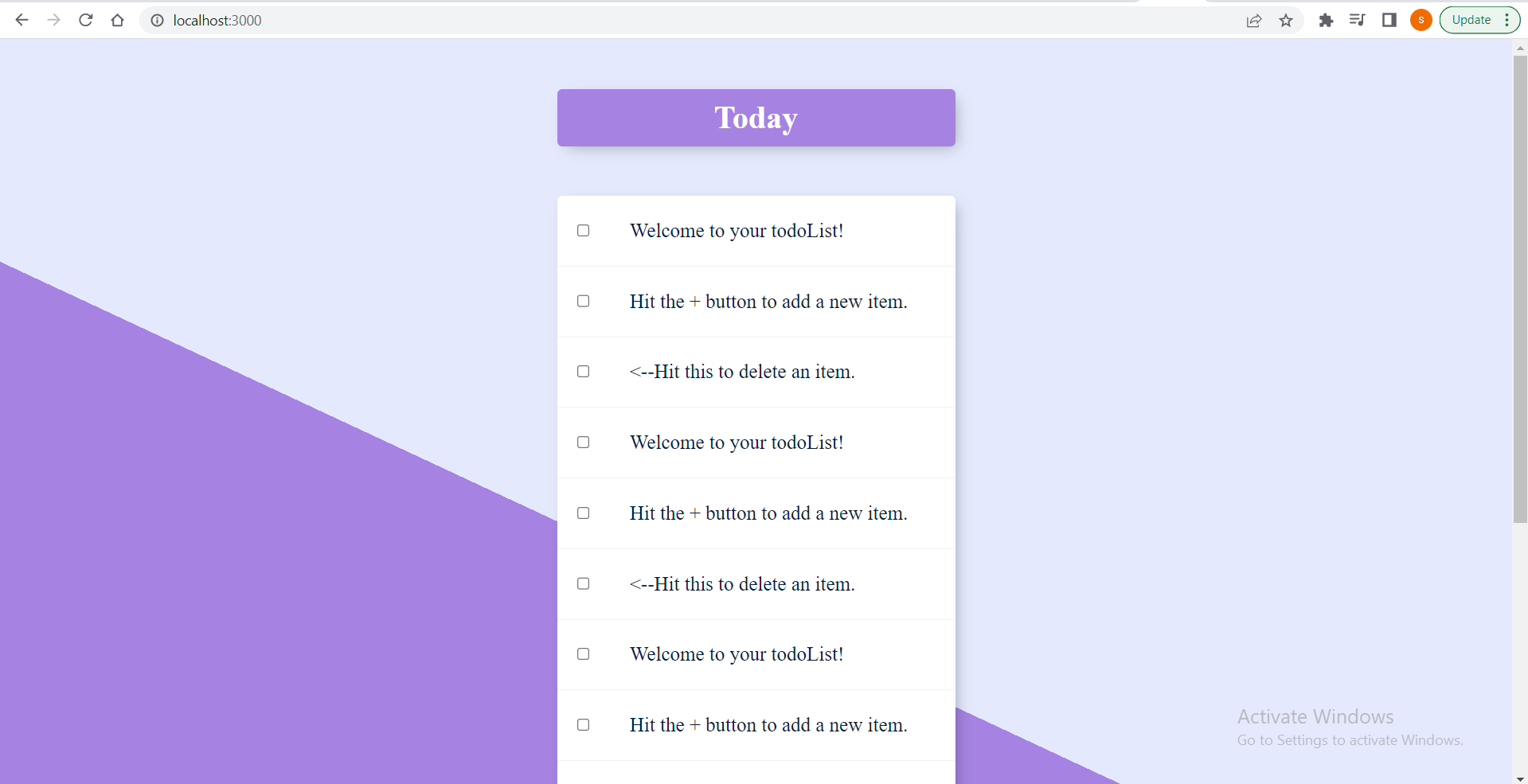


The entire objects are being passed.

We just want their names. And everytime we run our app.js we insert copies of the same list. Because insertMany function is being run every time.



Inside views/list.ejs, just tap and log the name field of the documents.

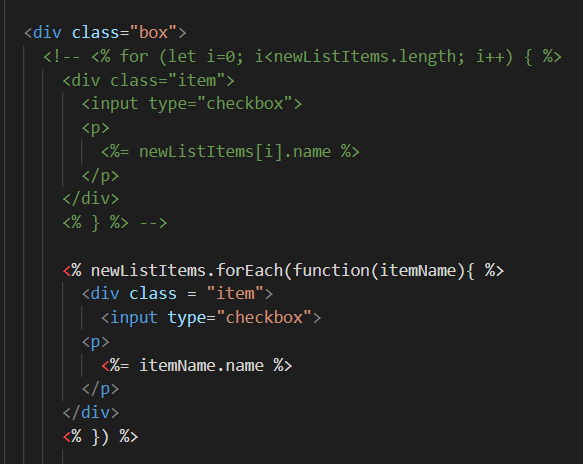


😊

But it still has many copies of the same list,

Challenge:

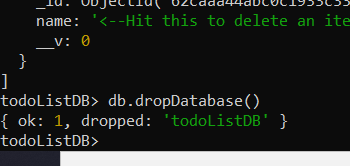
Use forEach loop inside list.ejs

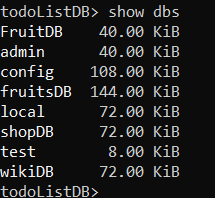


The insertMany function runs everytime, we just have to check if our defaultItems array is empty, if yes then only call function.

First we need to clear our database

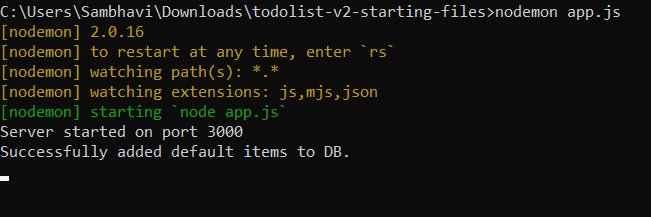
Go inside shell



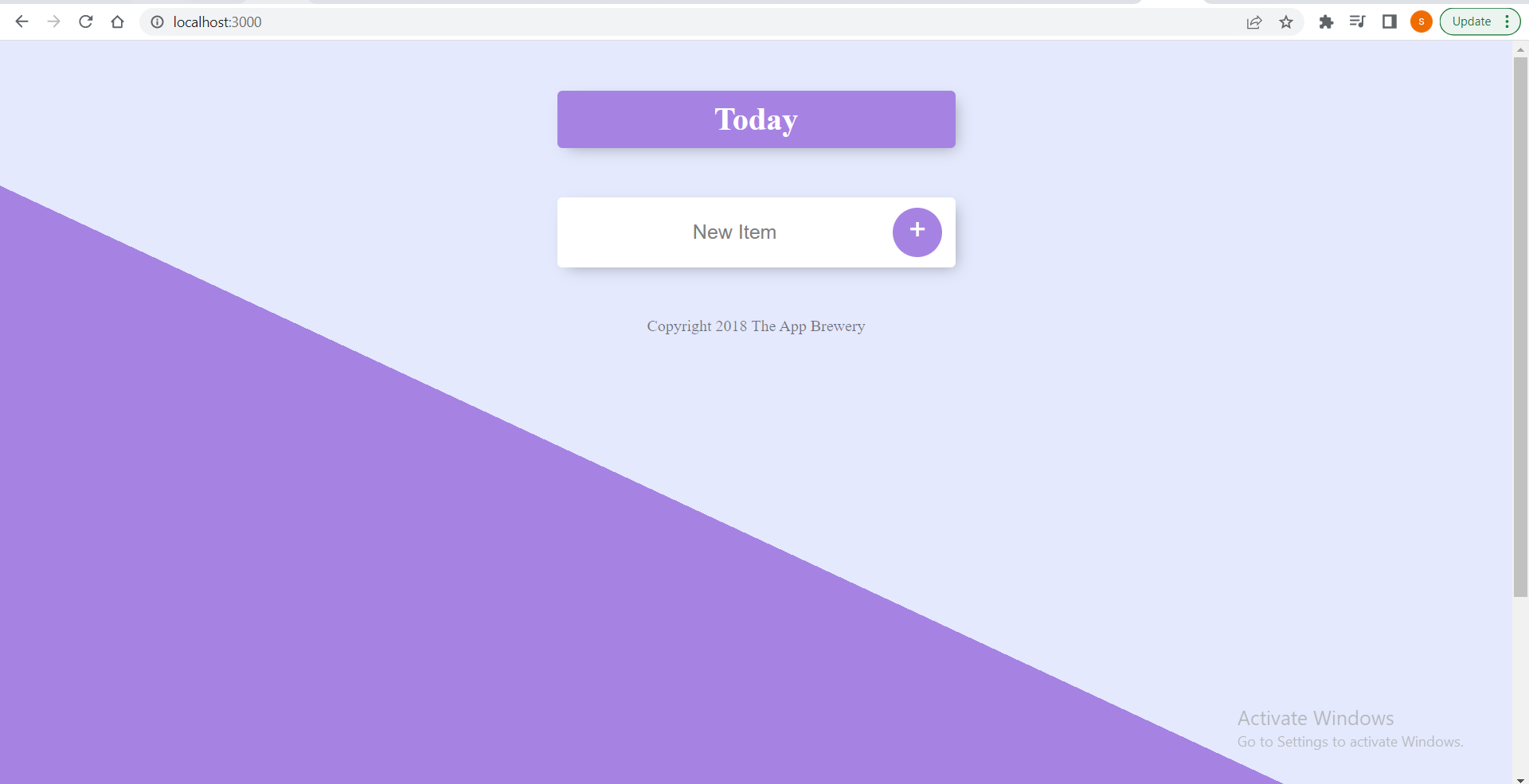


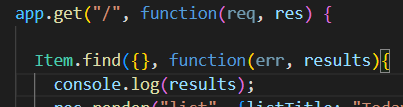
Now we don’t have our DB

Run app.js

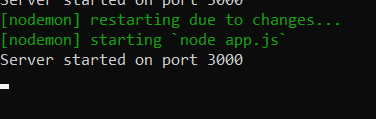


Comment our insertMany function,

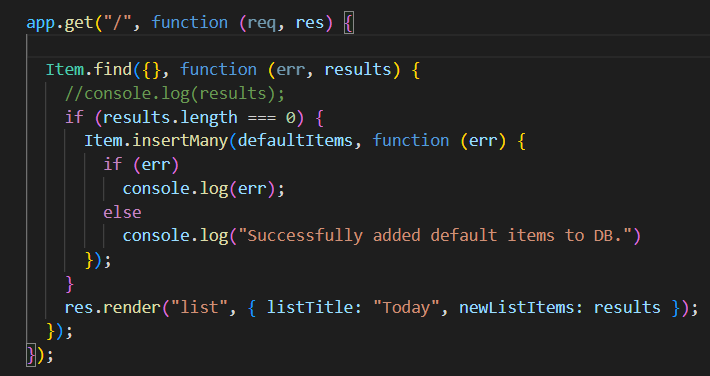




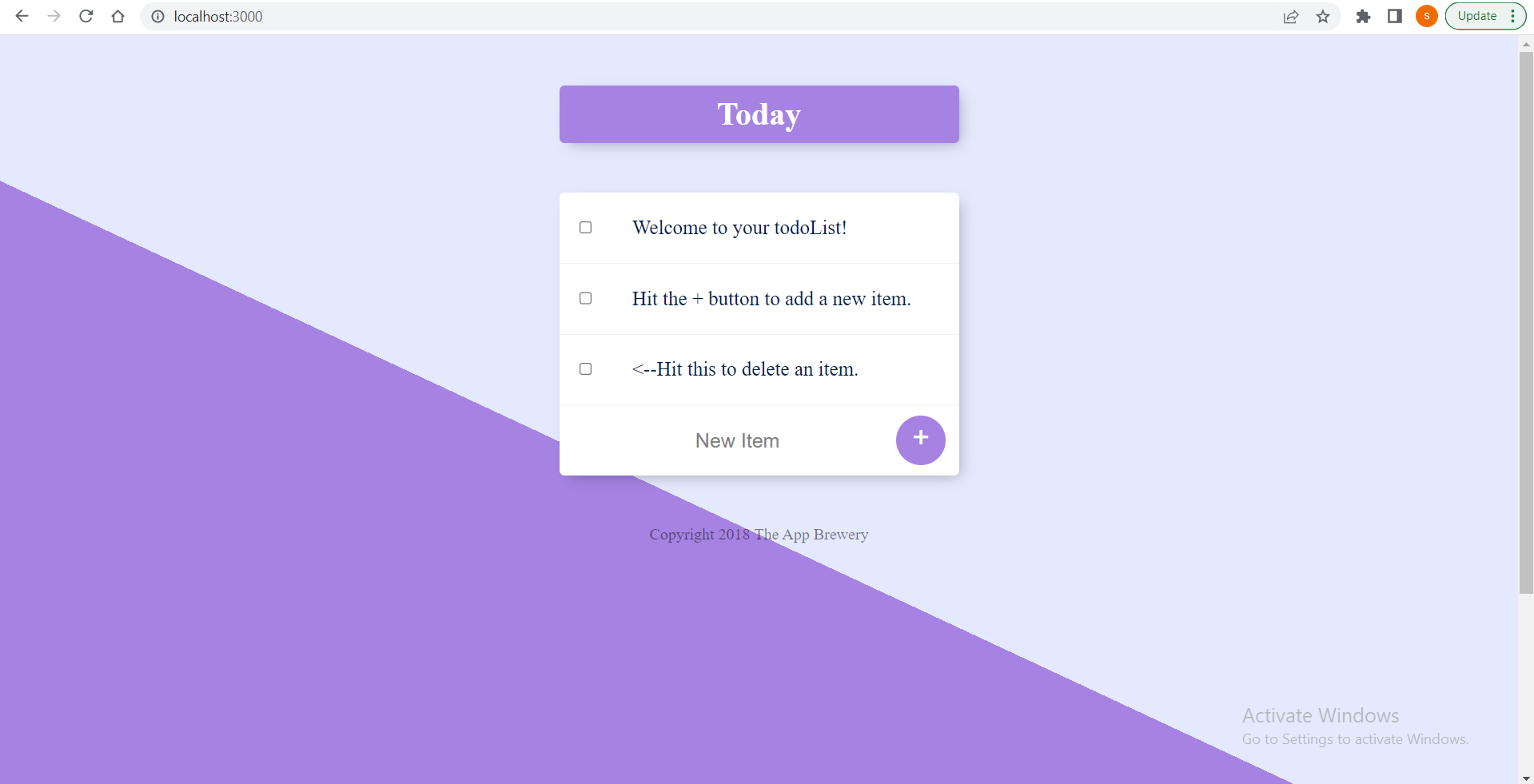
On console.log(), we get empty array results



So only when our array is empty, are we going to insert

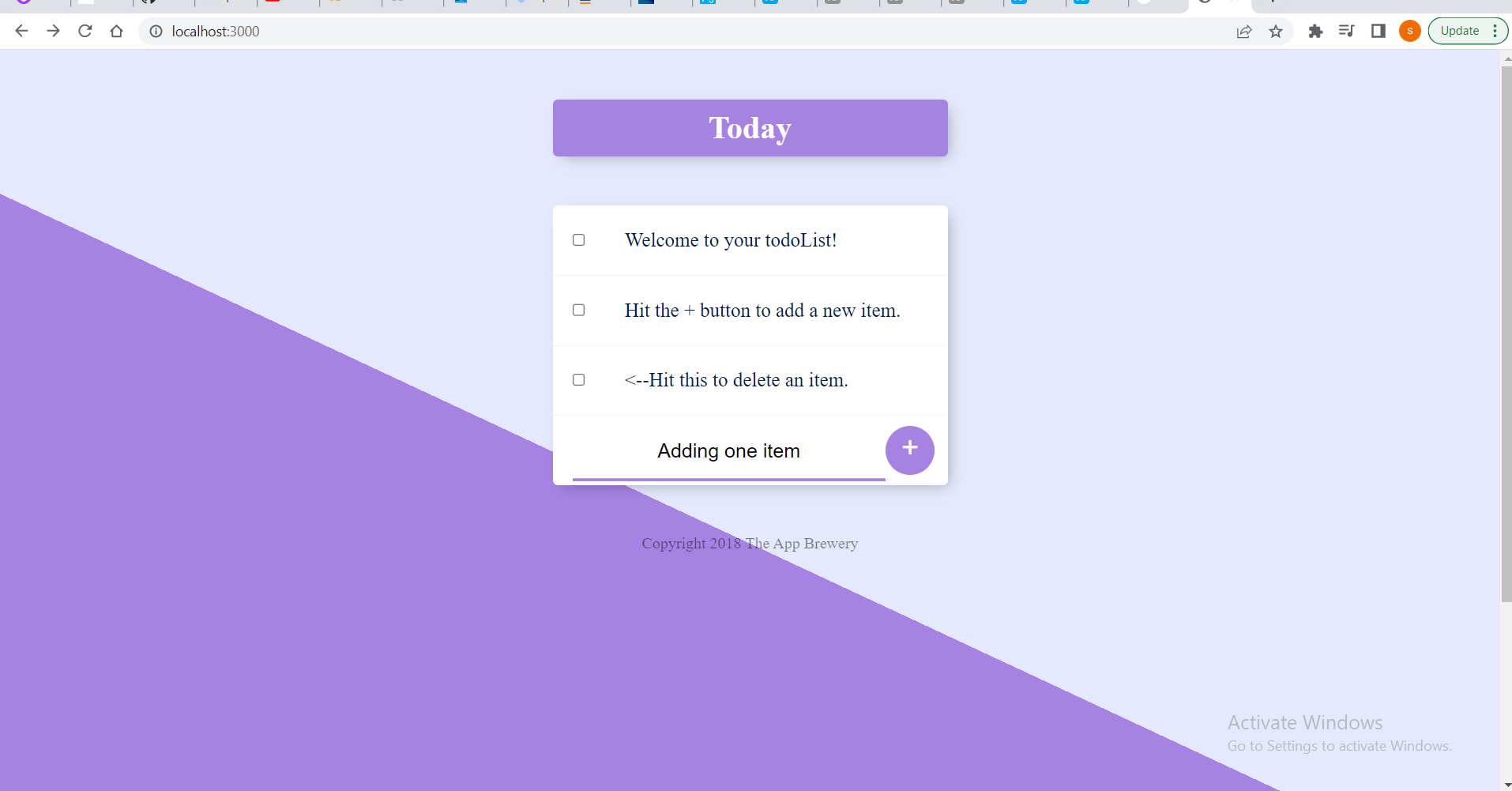


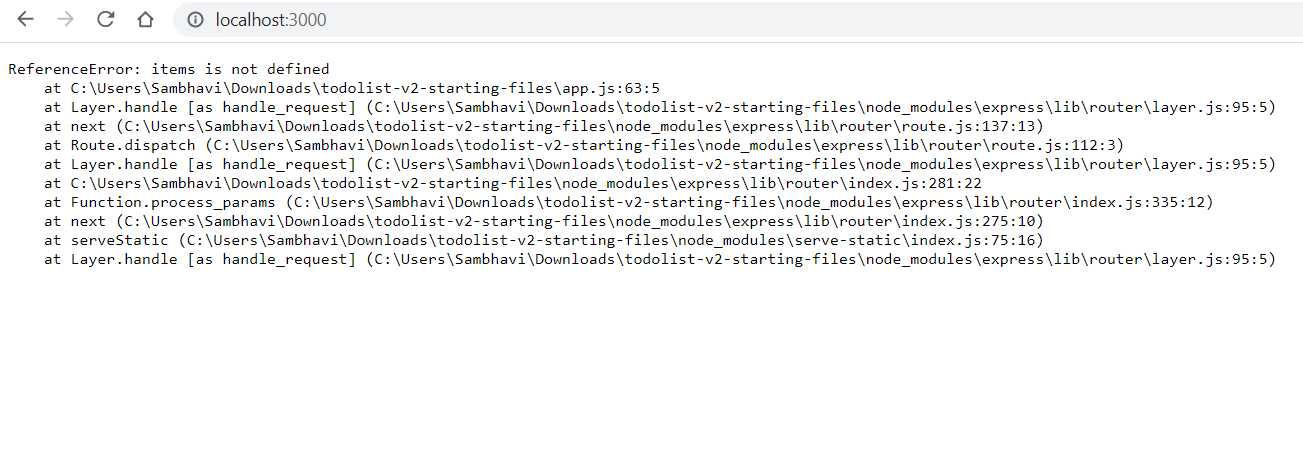
Put insertMany code inside if statement



No matter, how many times we re run, it still has 3 default times 😊

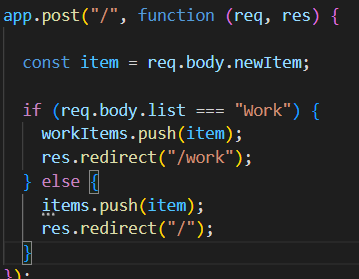
But if we try to add a new item to our list,





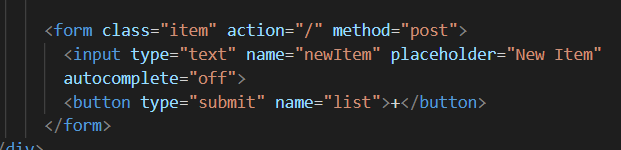
Our app crashes,

Because



This items array does not exist

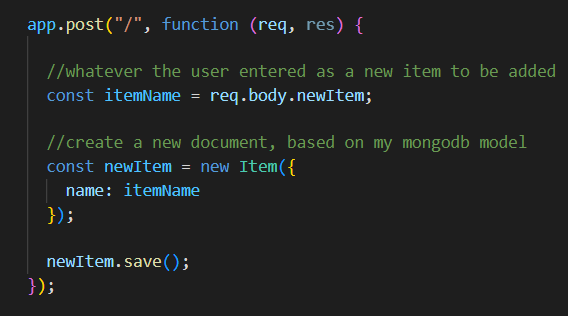
This happens in our post route, because that is when we try to submit our form

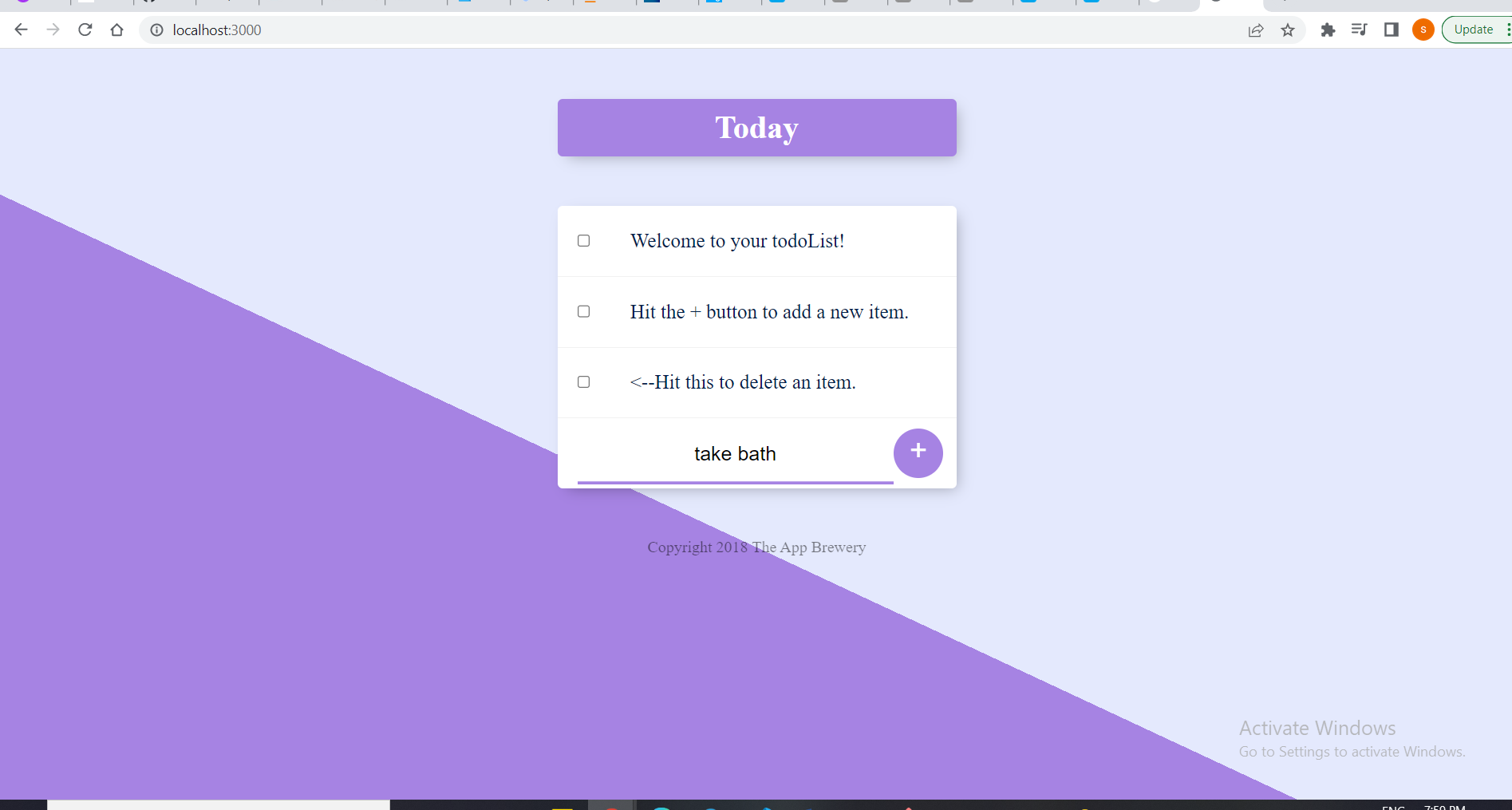


By clicking on the + button.

How can we do this using MongoDB instead.

Deleting whats inside the post function and writing a fresh body for our post request.

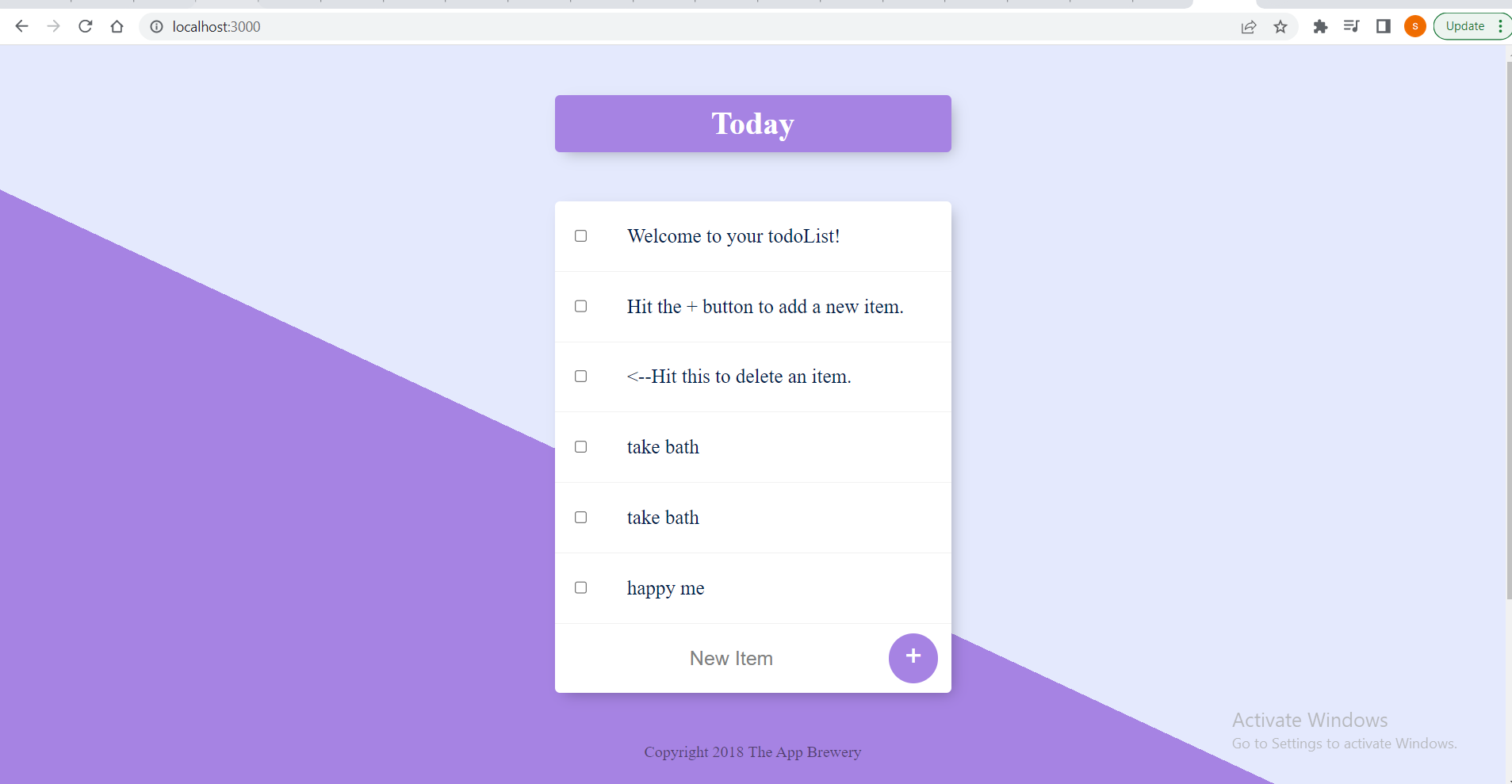


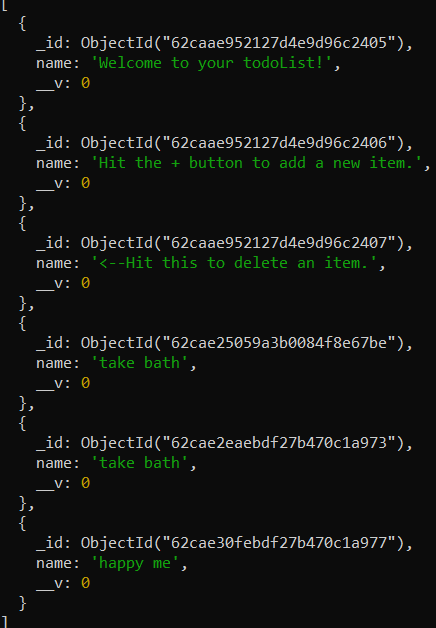


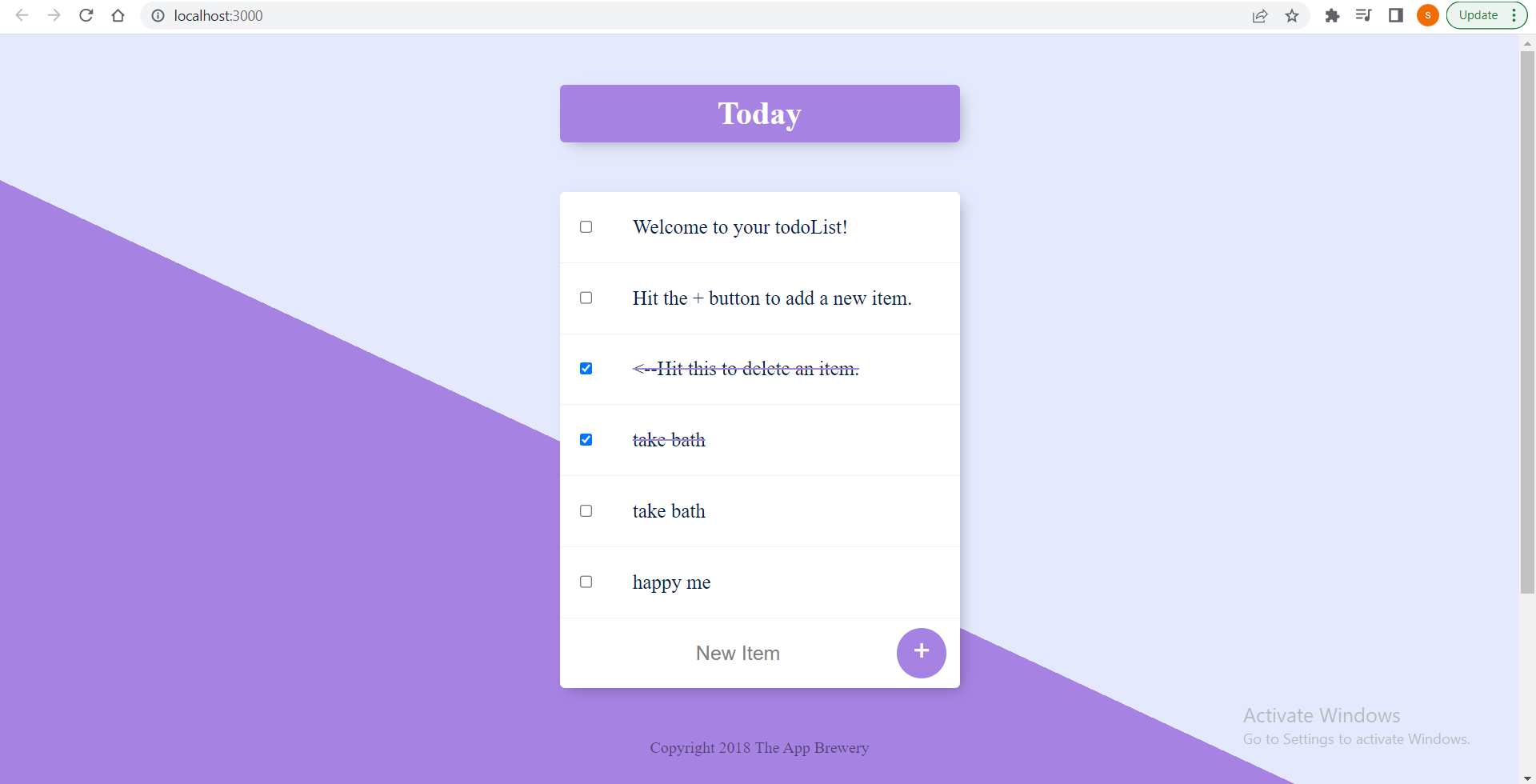


Yeee!! New item got added into my collection!!

In order to show it in our website, we just need to res.redirect to our homepage

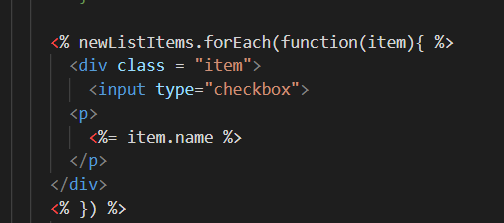




Deleting items:

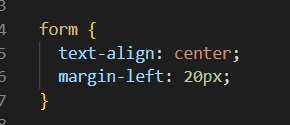
Right now, it just strikes it off, but it doesn’t get removed from the database.

So we want the particular item to be deleted when we click on the checklist button left of it. To do that, we need to remove that particular item from the items collection

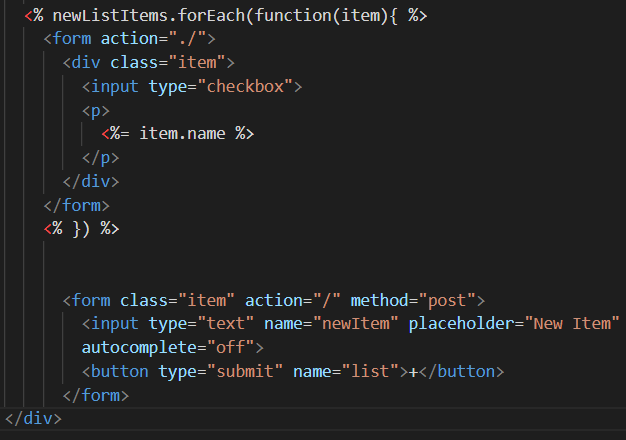


Currently, we have a input checkbox and inside it we have an item.name inside a paragraph. If we want it to send some data, we need to have a form and a post route inside it.

When we refresh, the styling looks a bit different. The reason for this is, in our stylesheet

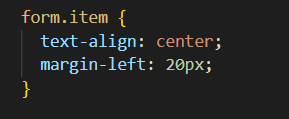


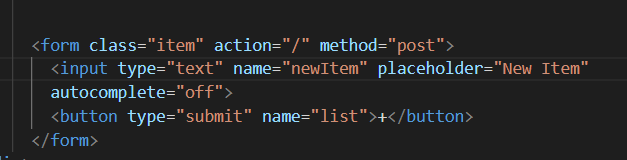
So this gets applied to both the forms



But we want it for only the bottom form, and not the one above.

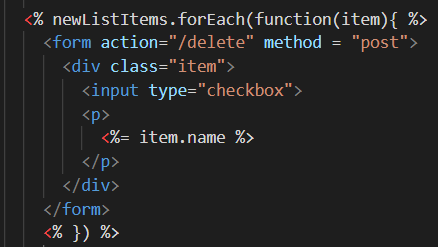
So, we need to apply our CSS to the forms that have class ‘item’.



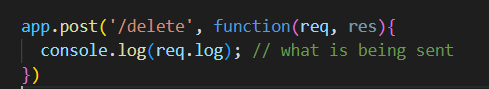


Now this post request was for adding items

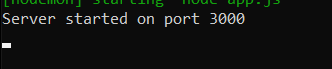
And the new form that we created, is for deleting checked items.

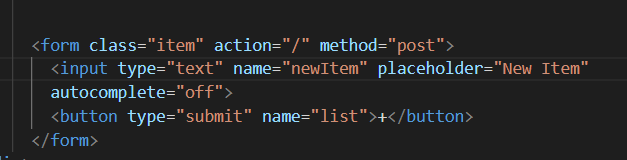


Inside app.js



Now if we check our console, nothing is logged

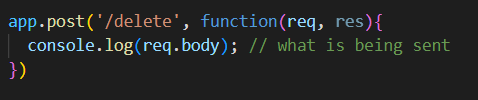


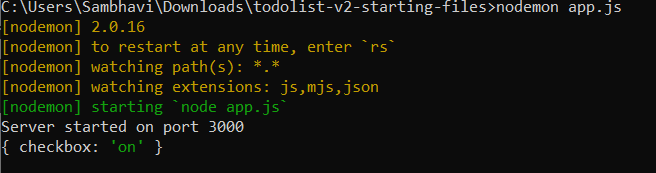
This is because a submit button, when clicked, makes the post request to whatever route, we specified like here 

But in this form, we don’t have such buttons to submit. We want that whenever a checkbox is clicked, the post request should be made.

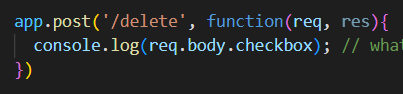
For accessing the input, we also need to give it a name

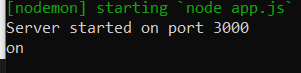






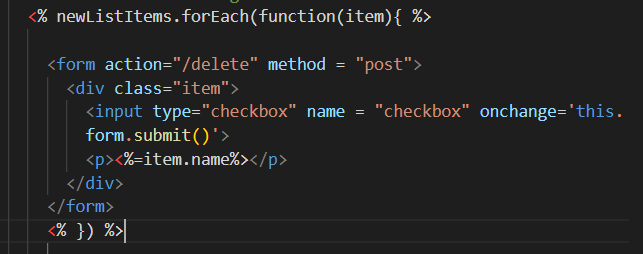
This is what’s being sent as a request.





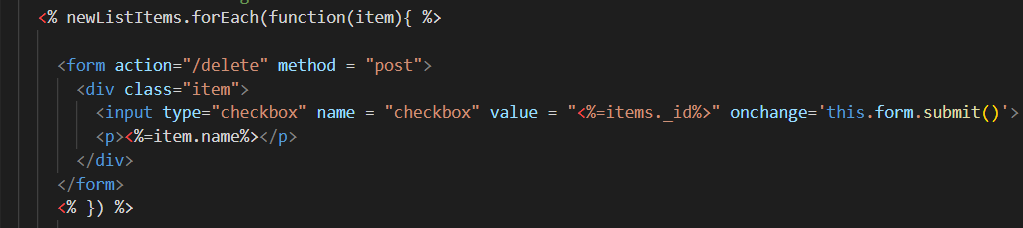
Now we get the value of the checkbox input(inside req.body)

But we want the item name that was checked off.

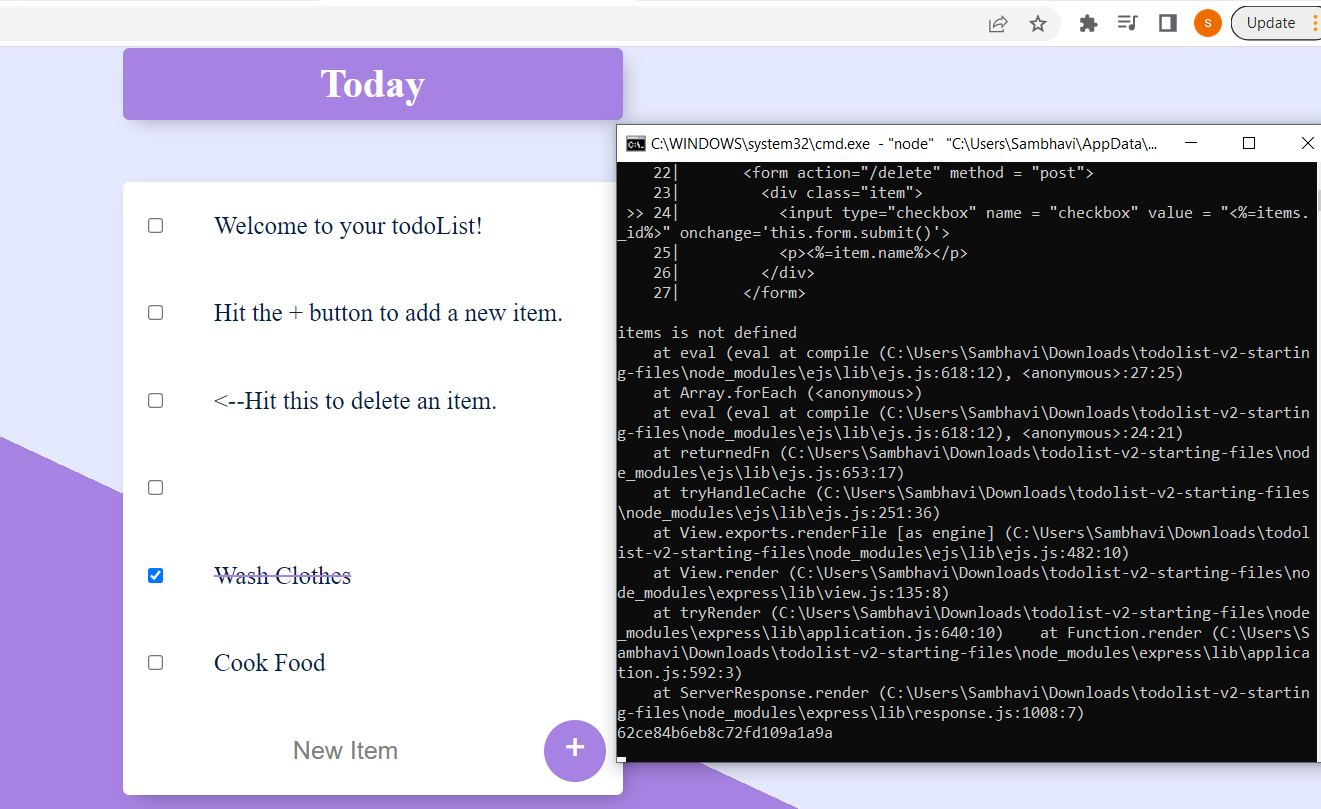


So our forEach loop is going to render all the items in the items collection as checkboxes. So we can also assign a value to our checkboxes based on the item’s id.

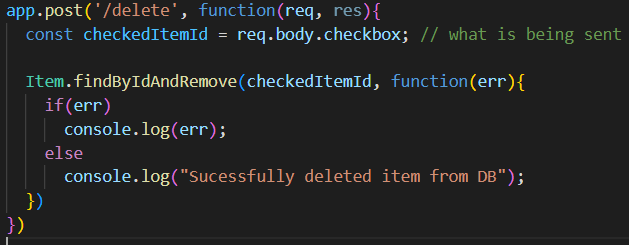
Add value attribute

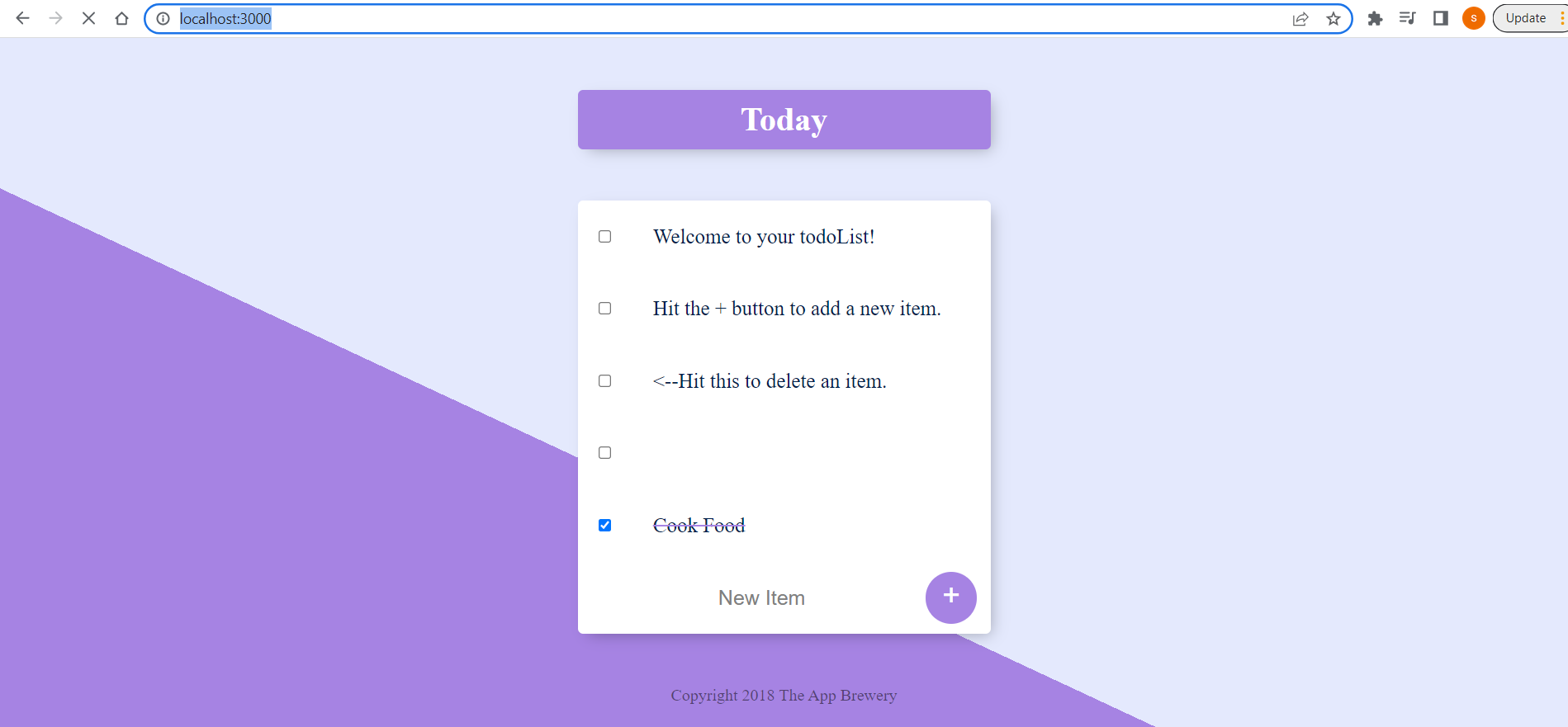


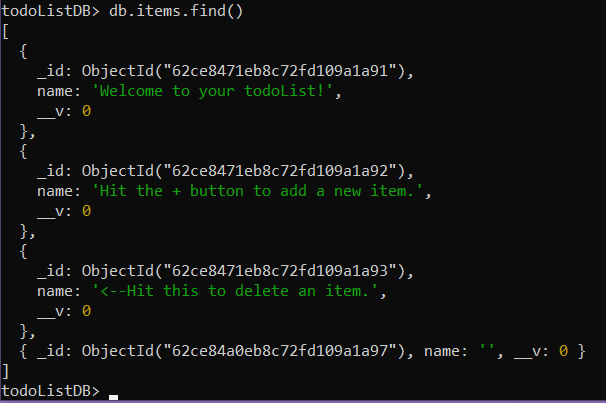
Remember, when we pass newListItems over, we are actually passing an array of mongo documents. Each of these documents/items that we’re looping through has a name and an id



We can the checked item’s id is being logged.



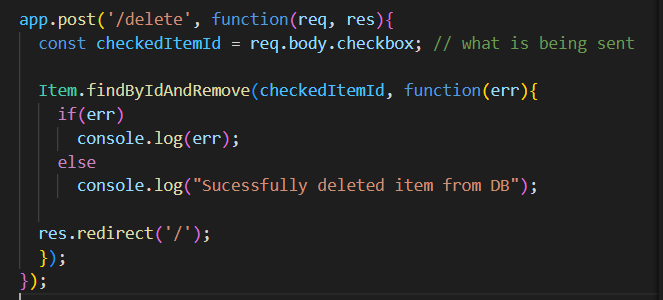




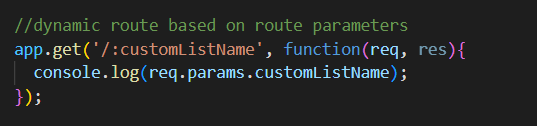
And deleted!!!! 😊

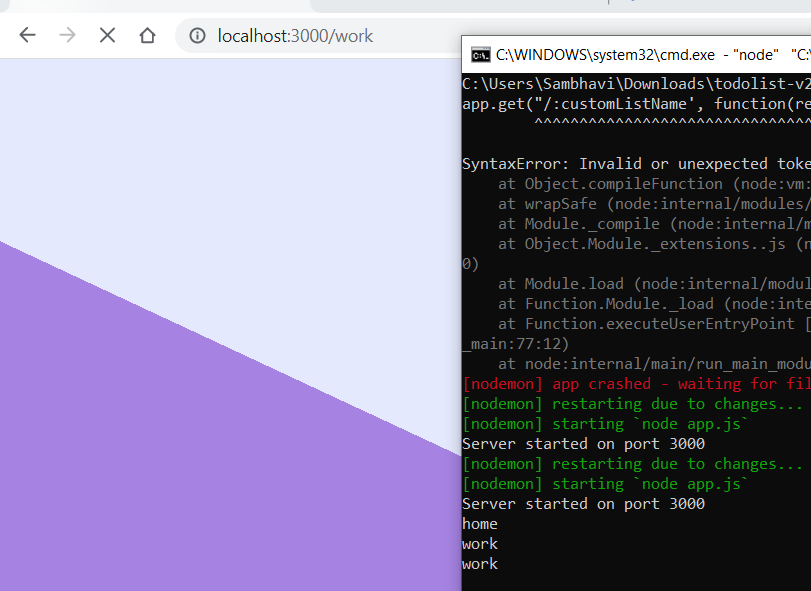
Just res.redirect for the changes to be reflected without a refresh.

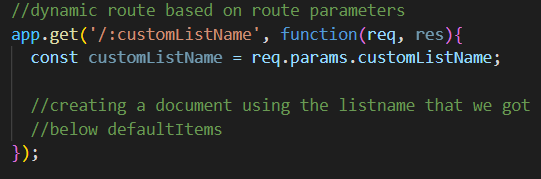
Complete code

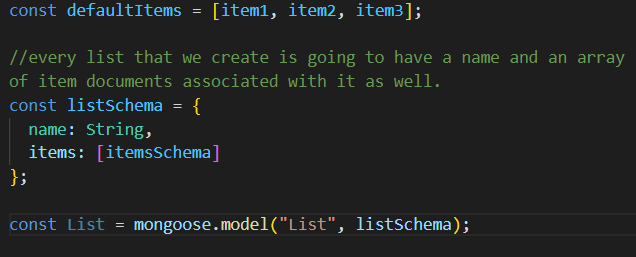


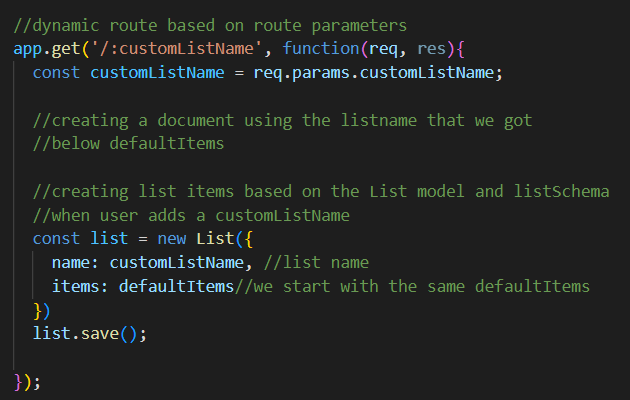
What if we wanted more than one list. We want to create new lists on the fly, if I type localhost/home, I want to create a brand new page, without me having to define all of the get and post routes. We can use express route parameters to do that.





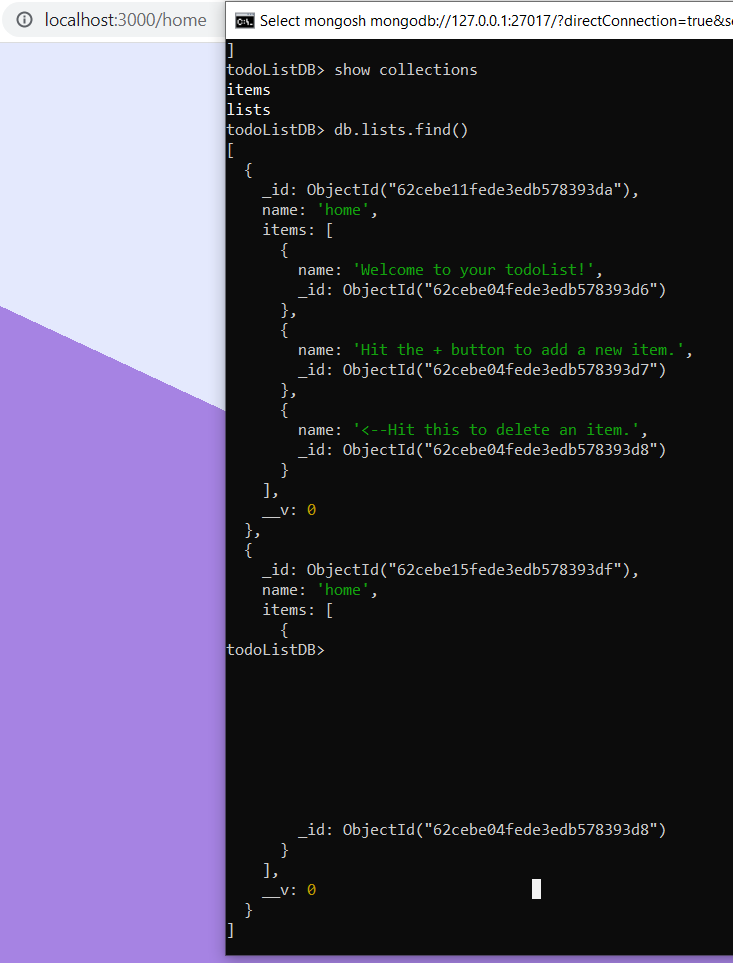


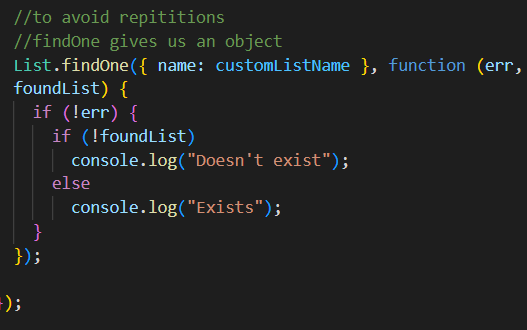




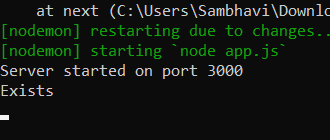
But if we go to the same route i.e type /home again, we will get one more document called home with the same defaultListItems

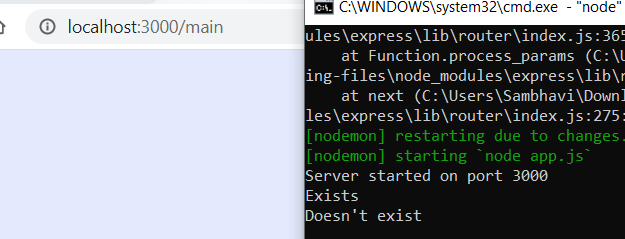
We have to check if a list with the same name already exists in our list, if it does, we wont create a new list, otherwise we will.

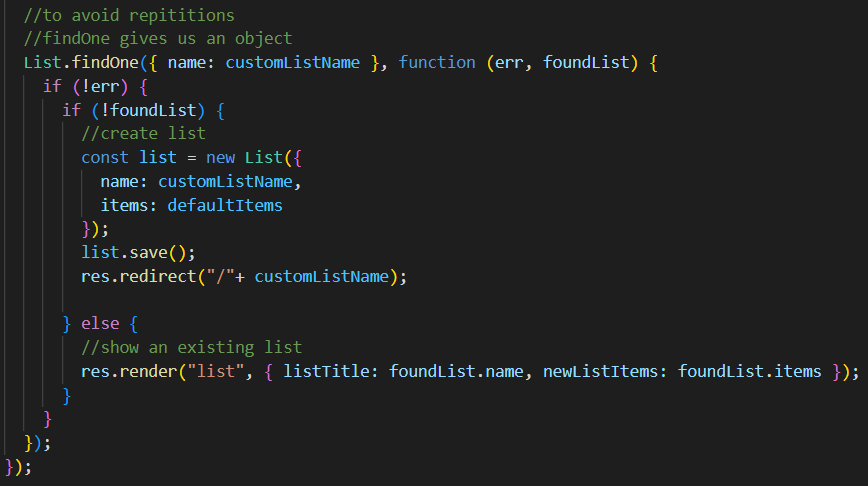


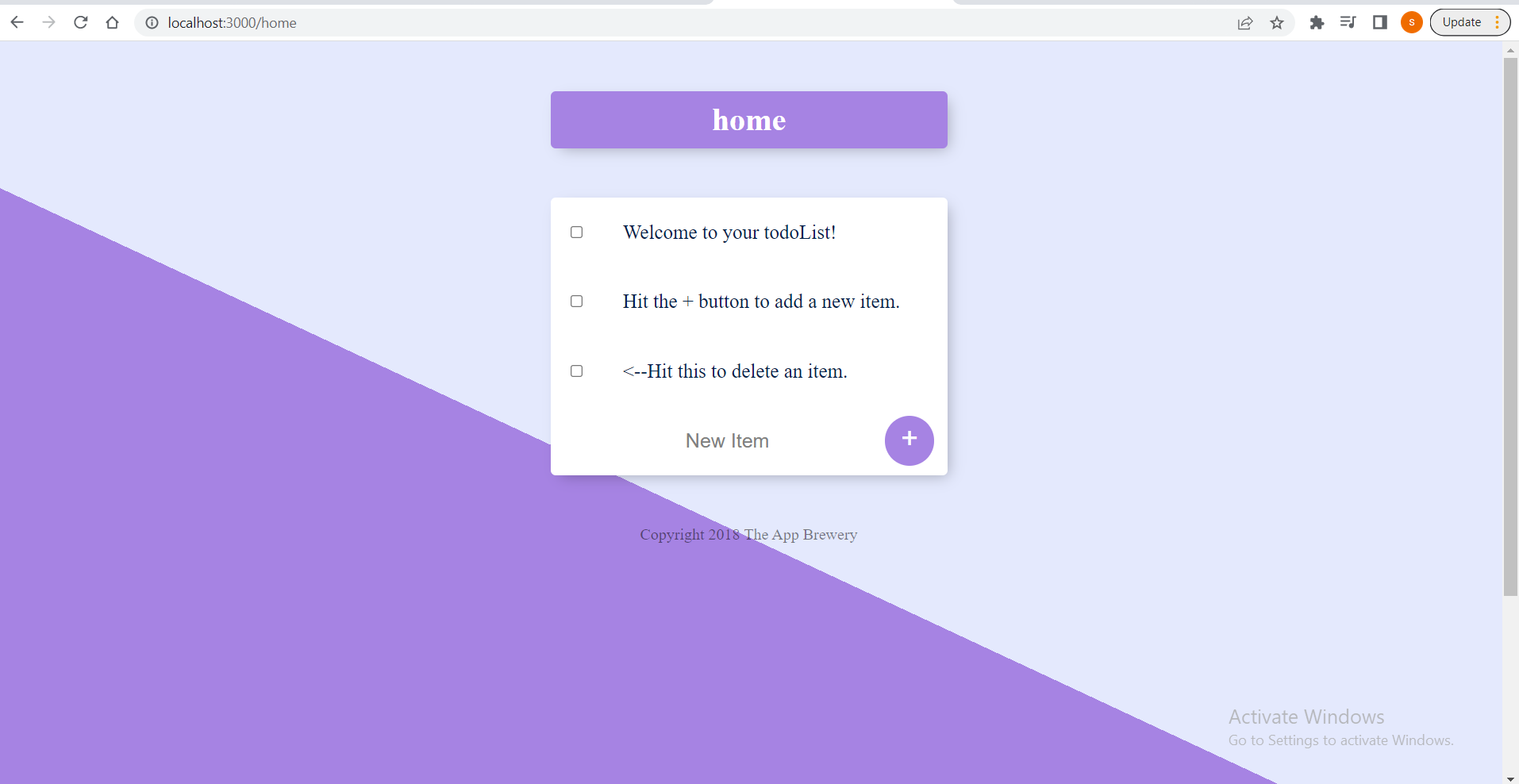




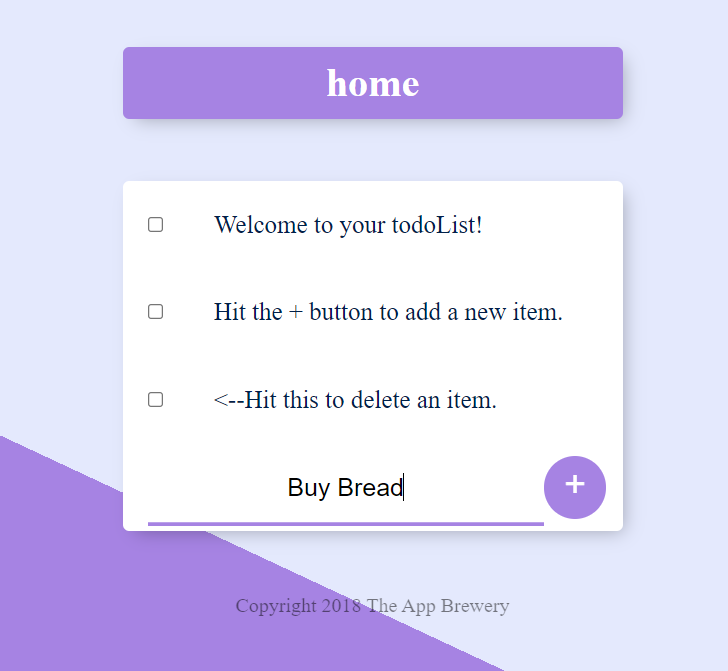


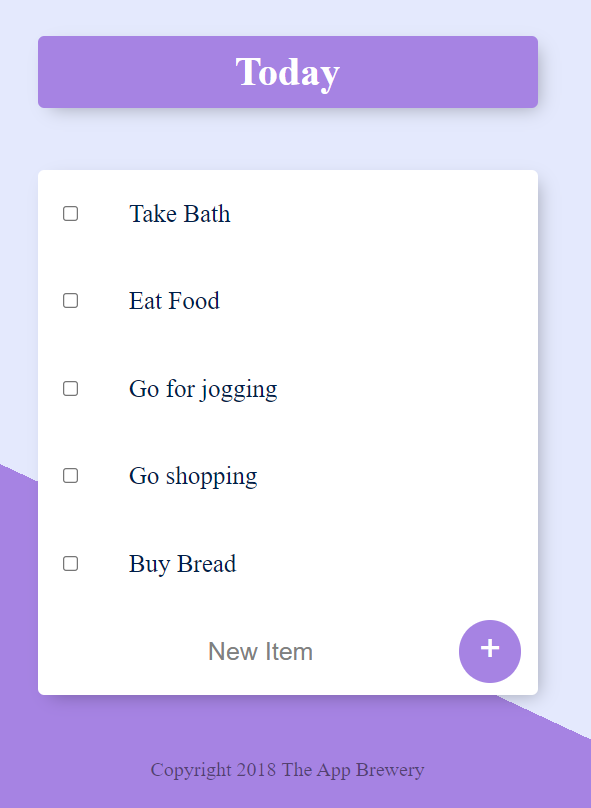






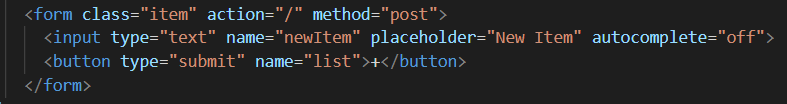
There is a problem, if we try to add something to one of the custom lists,





It gets added to the localhost list(homepage)☹. We get redirected to the route route

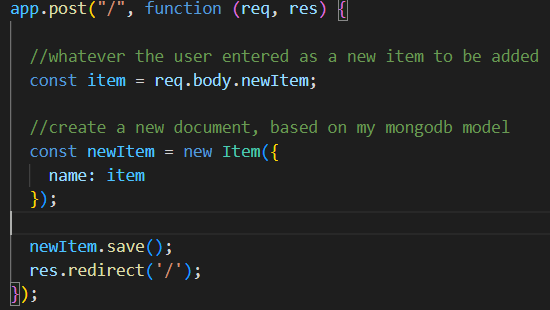
This happens because inside our list.ejs



Whenever we click on the ‘+’ button, it makes a post request to the home route. That is, we are always posting the new item to the root route.

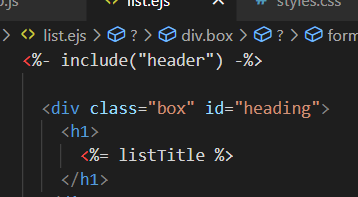
And since this the template file, we use the same template for the home list, the root list, and all other lists. So essentially, every list posts new items to the root route.

Inside the post request,

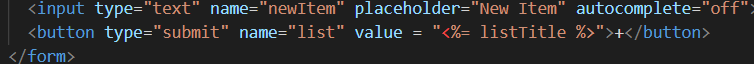


We create a new item, we save it and redirect to the root route.

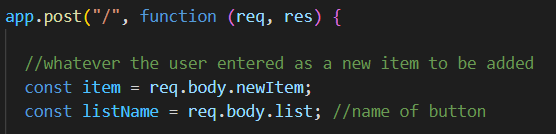
We actually need to pass over the list that is currently being displayed. We have access to it in the listTitle variable

****

And we need to pass it back, when the submit button is triggered.



Now, whenever we submit, we should have access to the ‘newItem’ and the ‘list’

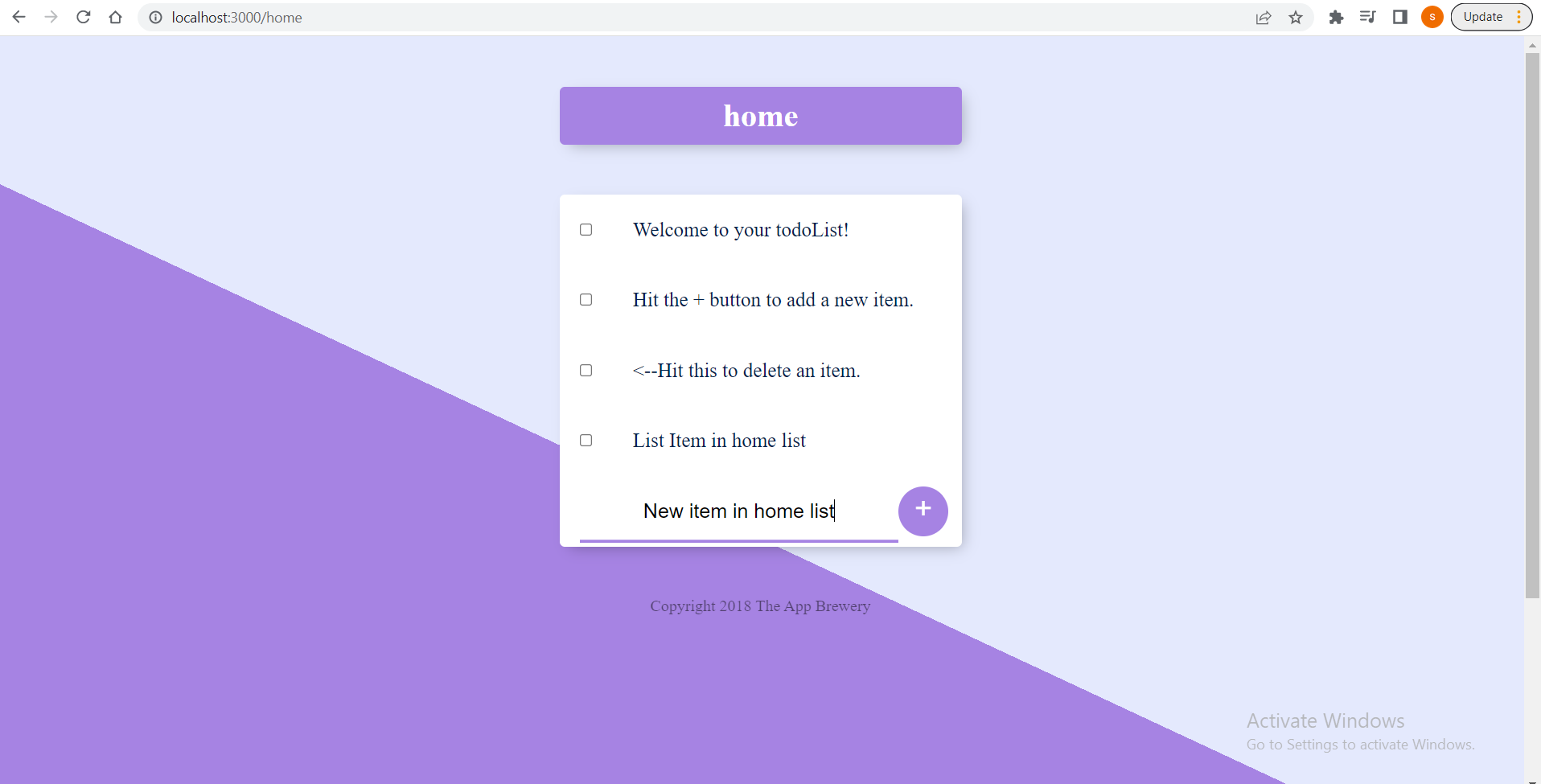


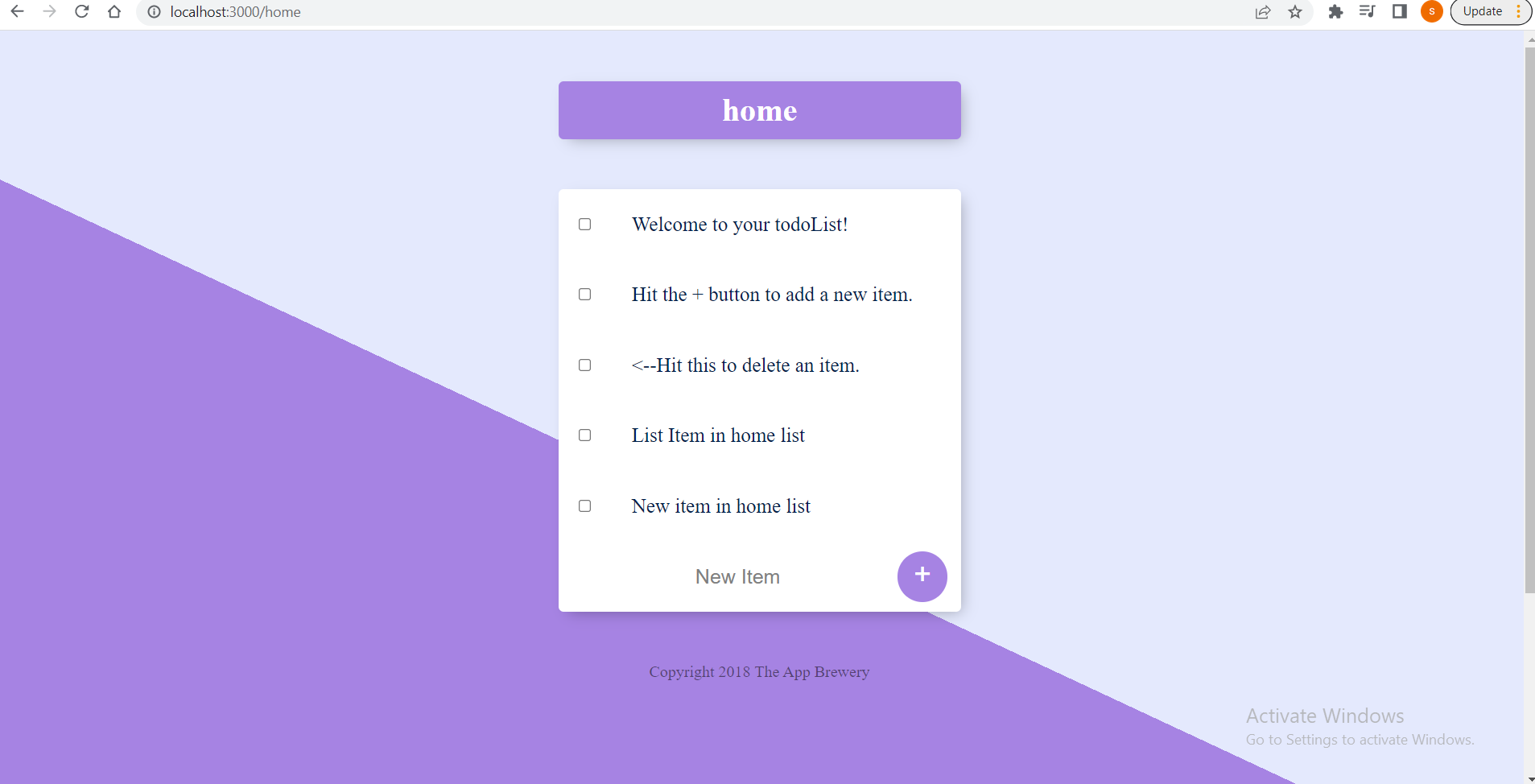
Now we have to handle the part, where we decide which list does our item gets added to.

Now, when the user submits, the newItem is sent to the post request of the root route,



A new item is created, if the post request was made from the root route, the item will be saved into the items collection and redirected to the root route, else if the listName is found, the added item will be pushed to the items collection of the foundList document, saved and then redirected to the list to which we added an item.

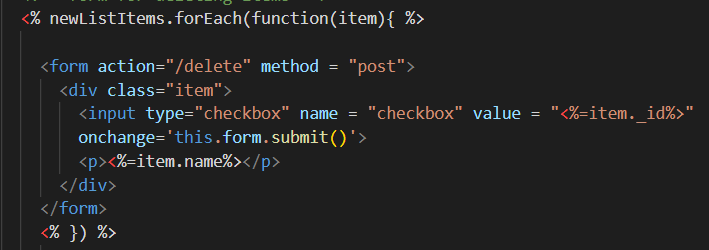




Yes it works 😊

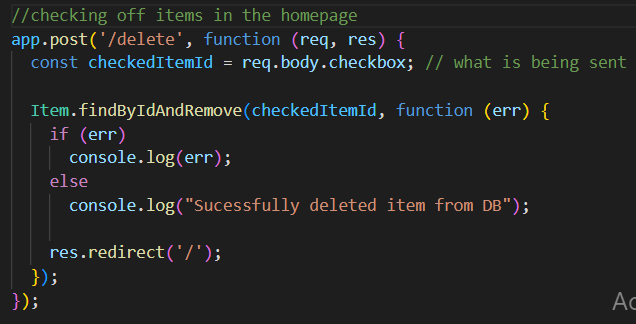
If we try to delete items from any of our customList, it doesn’t get deleted, instead we get redirected to the homepage.

The way we are implementing the delete functionality, is through this form



This form makes a post request to the /delete route and it passes over the item\_id that the user wanted to delete and the form gets triggered, when one of the checkboxes get clicked.

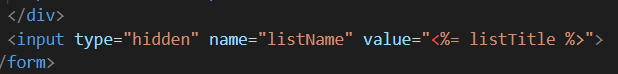
And in that post route,

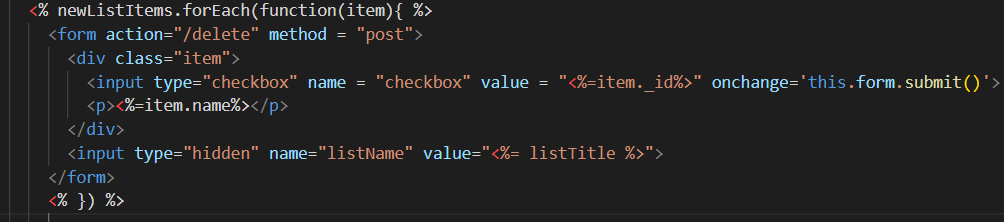


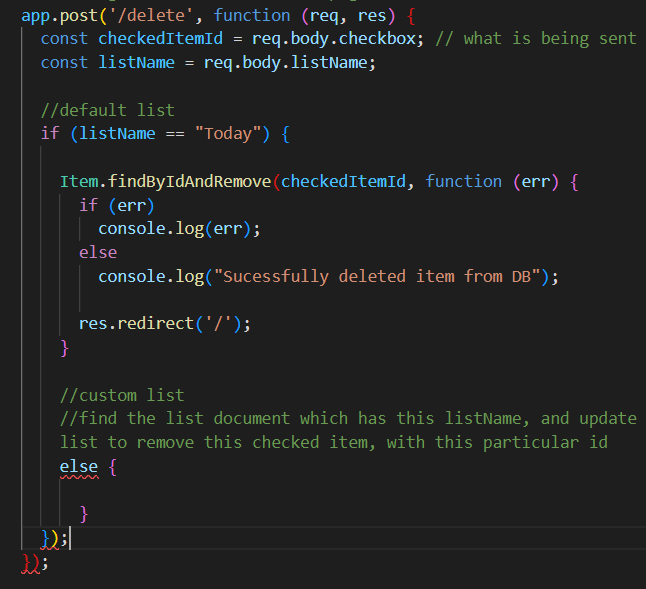
We only look for all of the ‘Item’ but we don’t check which item the list is from, in order to delete it from the correct list. So we need 2 pieces of info when we tap into the /delete route, the id of the checked item and the list it came from.

In this form, we don’t have a button so we can’t pass a value to it like we did previously.

We’re going to use input type = “hidden”. It allows us to include data that can’t be seen or modified by users when a form is submitted, for example the list name.

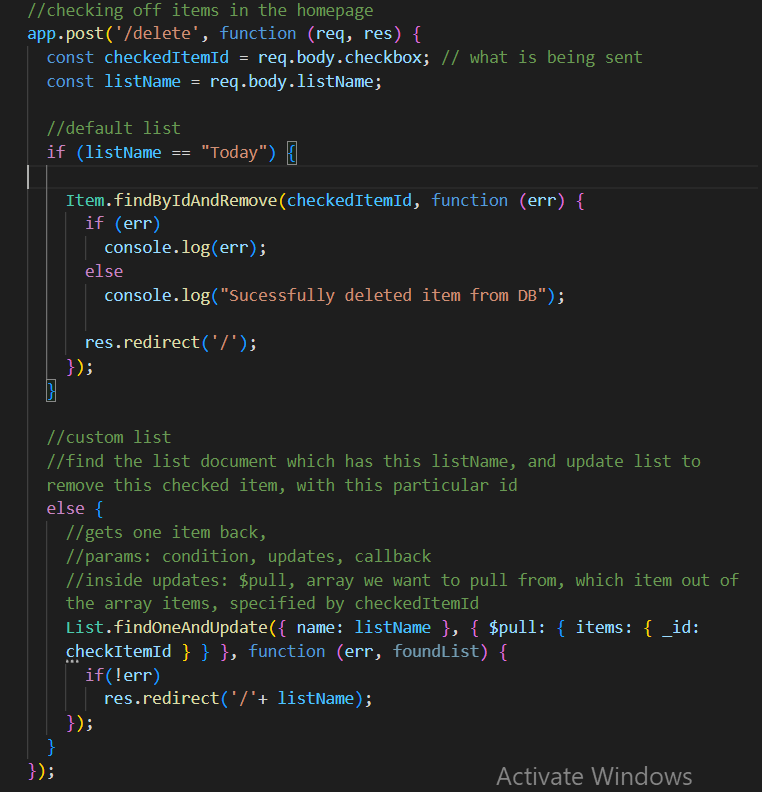




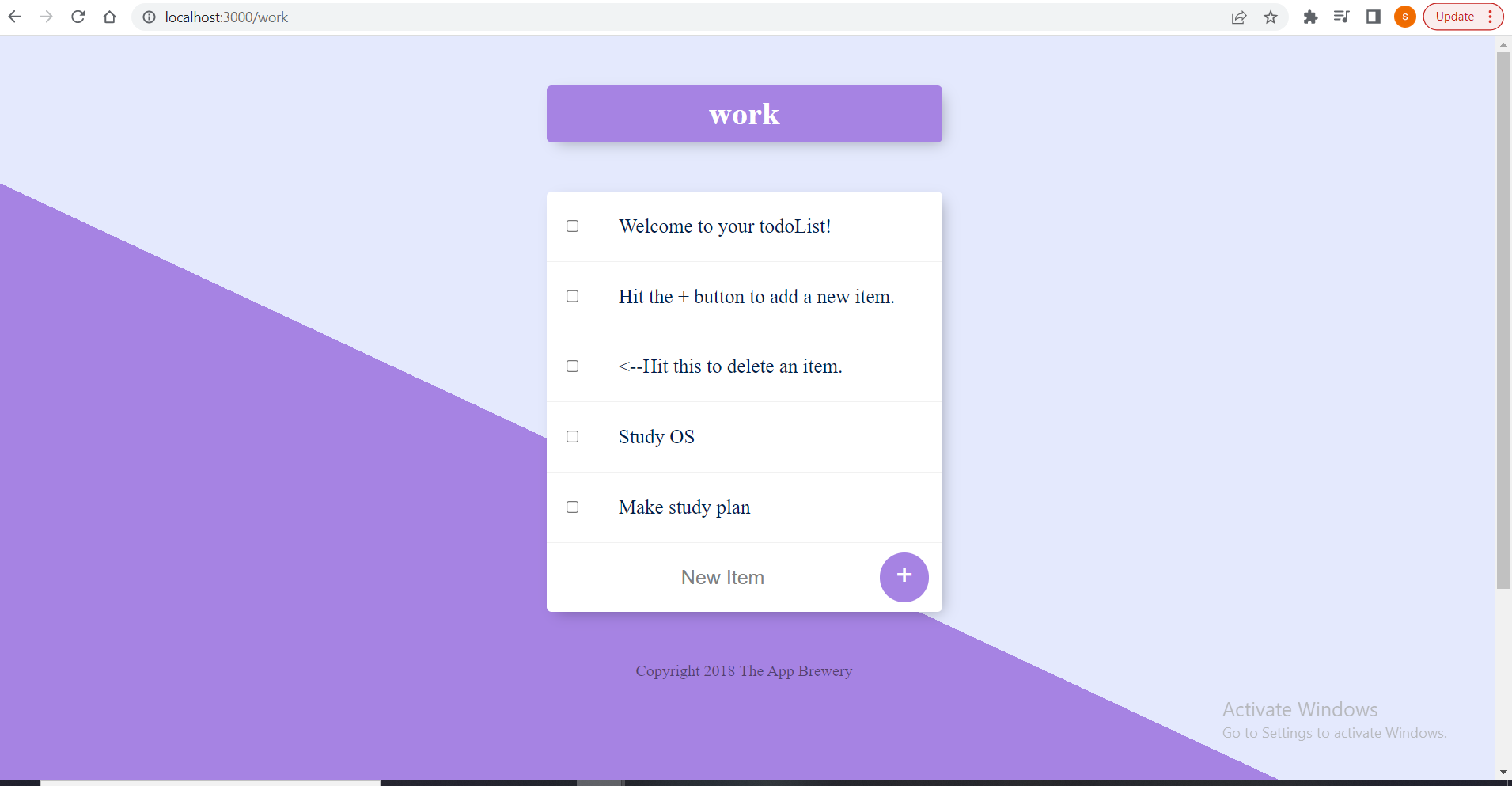


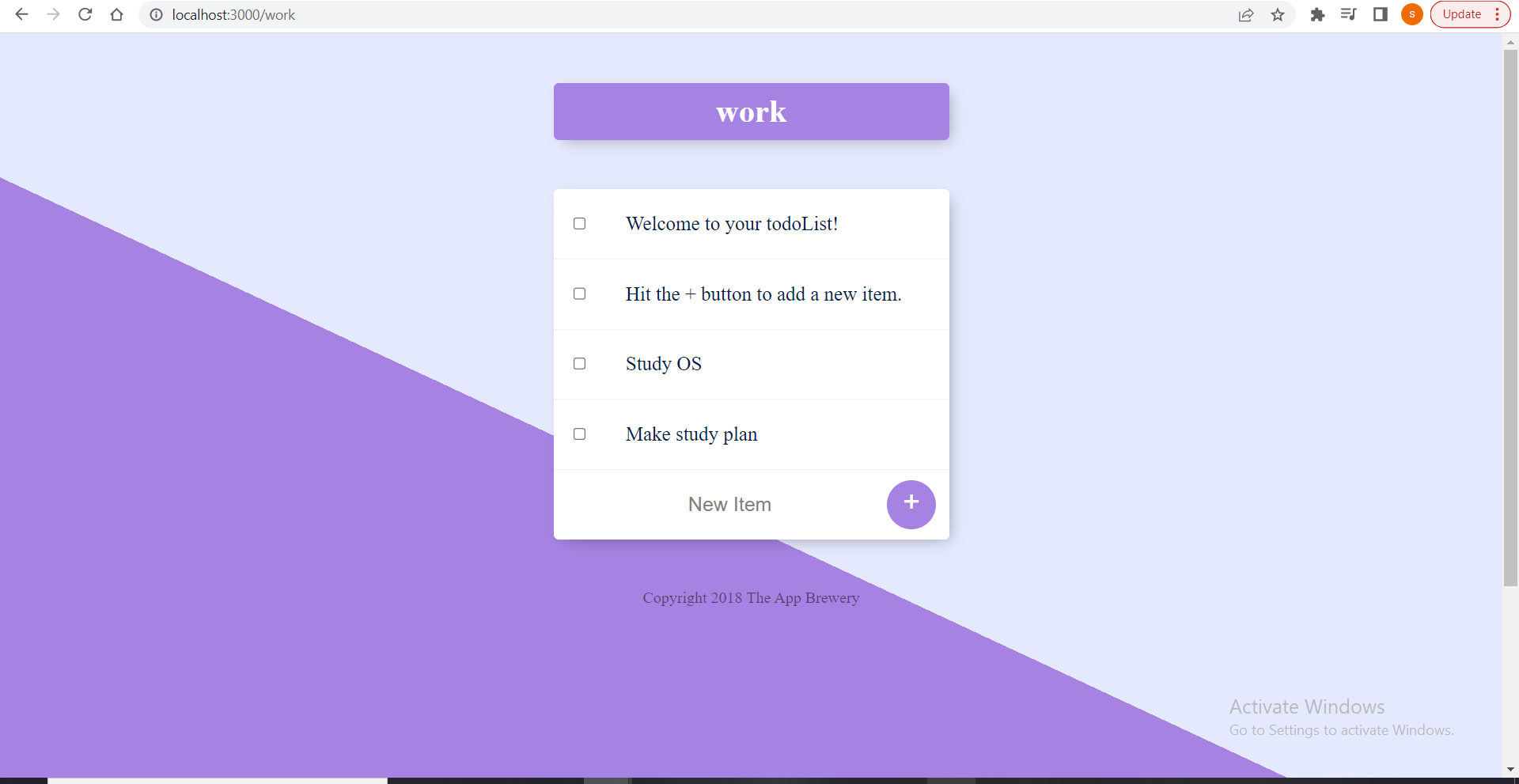
Because, inside our list document, there is an array of item documents, it is a little more complex. Because, we need to find that particular item inside the array of that list document. We have to iterate through the array of this item document to find the item that matches our particular id.

We’ll use the $pull method along with the findOneAndUpdate mongoose method



And it works!!!





We checked the third item and deleted it!

Just one last thing, our routes differentiate between ‘/home’ and ‘/Home’,

So lets remove that.

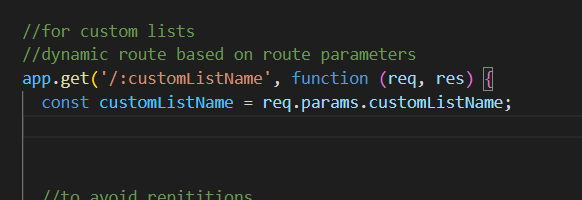
Using lodash

Lets say we always wanted the capitalized version of our list name, we want to change whatever input the user enters into the customListName field, to a title case.

Use lodash

Npm install lodash

\_.capitalize



We’ll use it here.

