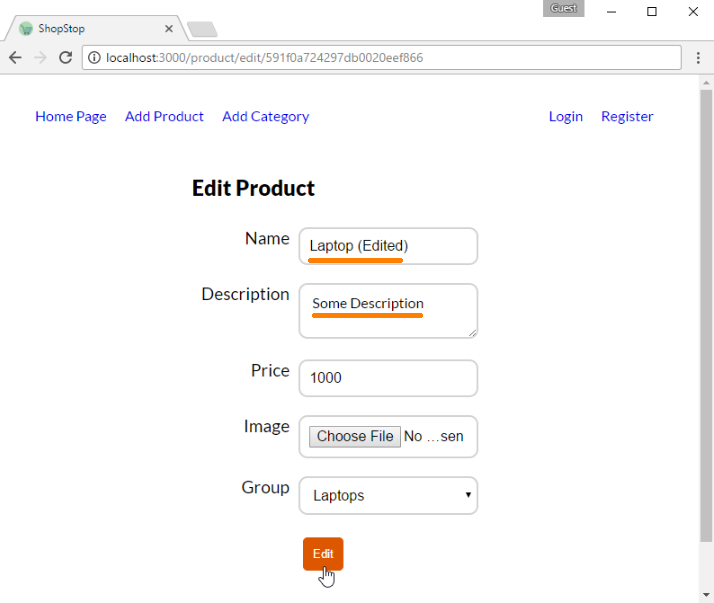
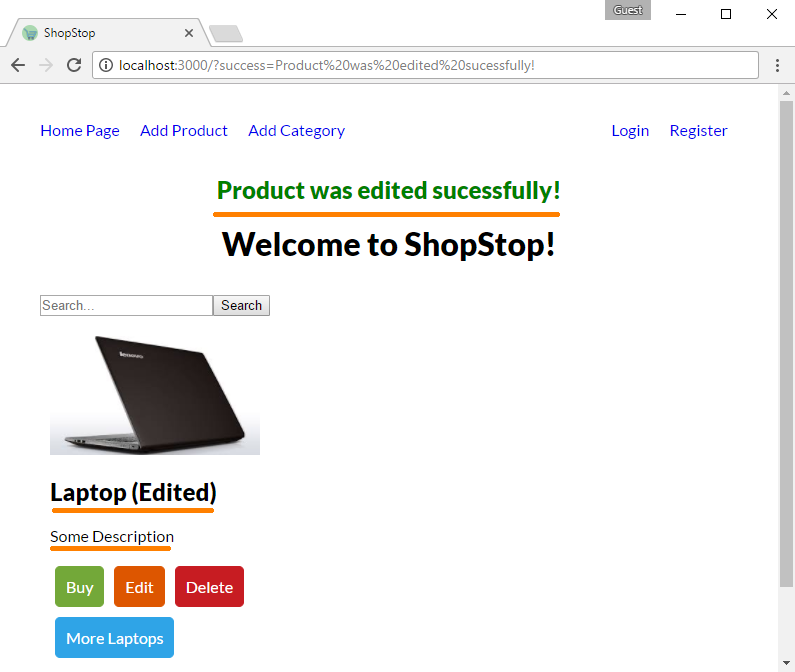
Go to **home handler** and update it to look like this:

|  |  |
| --- | --- |
|  | **Remove** any **route** and **http** **checks** and **extract** every single if-case to a **function**. |
|  |  |
|  |  |

## Edit Product





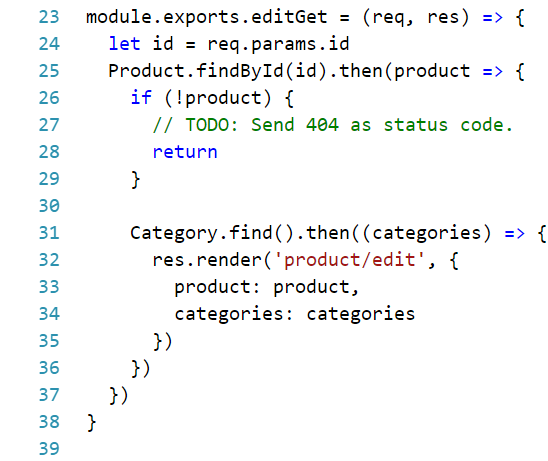
Before diving into writing the logic behind editing product we should consider two key things:

1. Product's picture should be **edited** if **only** a new one is **uploaded** (e.g. if we edit only title there is no need of uploading same picture)
2. With changing product's category we should **remove** **the product's reference** in **old category** and **add** it to the **new** one.

With keeping this arguments in mind we are ready to create our **product/edit** **view**:

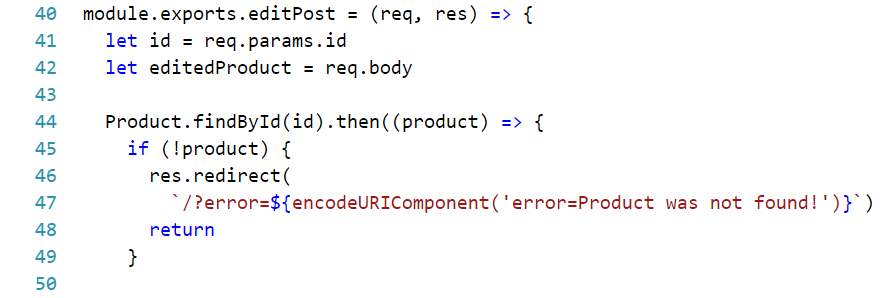
|  |
| --- |
|  |

Now in the **product** **handler** and add action for visualizing the edit form:



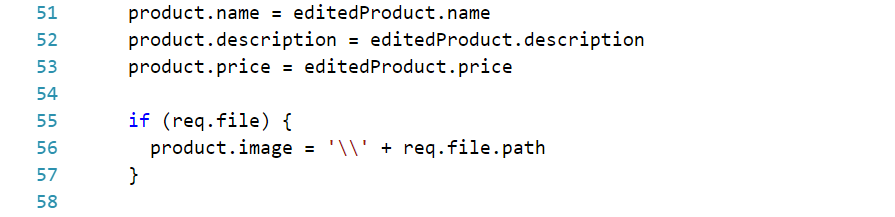
\*Hint: [here](https://stackoverflow.com/a/42370052/5591237) is how you could send status code

This was the easy part, now we have to implement the real logic behind editing a product:



Note the usage of [encodeURIComponent](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/encodeURIComponent) – it will simply **escape** any **invalid** url **characters** (like "/" or " ")

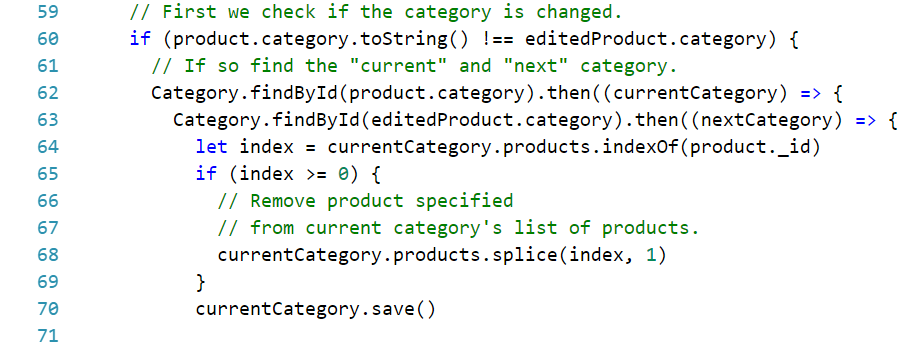
Then we continue with editing the product object:

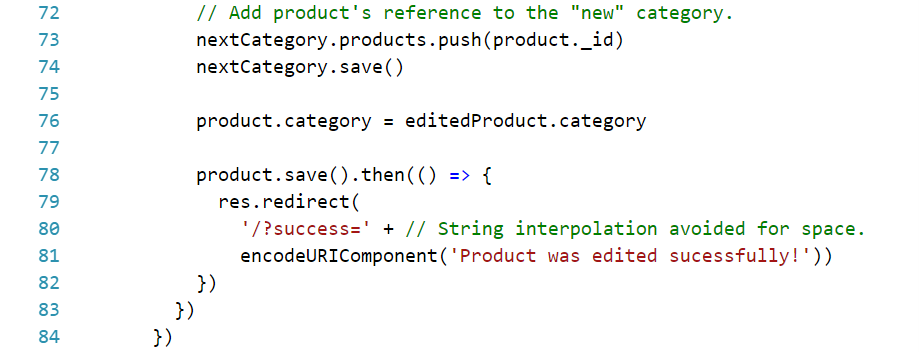


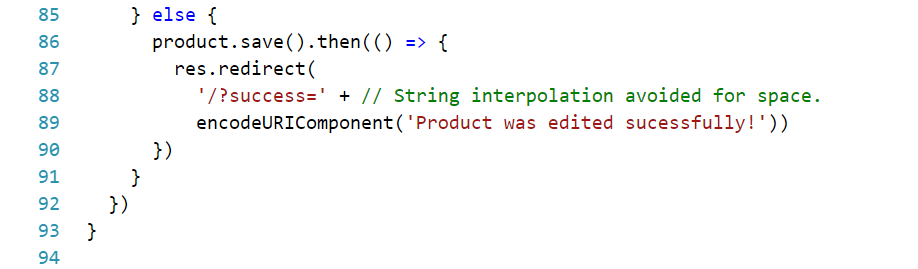
We just **re-assign** **values** passed **from** the **form** in order to **update** the **current** **product** object.

The next part is little messy though. The reason is because we have to deal with the relation between product and category.

Whenever we **change** some **product's** **category** we should go to that **old category** and **remove product's reference** from it and last but least in the **new category** we should **add product's reference**. So here is a suggestion of how it could be done:







One last thing before we move on – **routing**:



Use the following html:

|  |
| --- |
|  |

**\*Note** that only the **GET** action should be implemented. When we have users we will come back for this one.

**In the next part** we will implement the functionality behind **buying a product** and we will cover up **user authentication**.