# Web Worksheet 11: Edit and Update Products

## References:

* Using Python with Sqlite3: <https://docs.python.org/3/library/sqlite3.html>
* Code for Worksheet 10

**Pre-requisites for Lab**

* Completed ProductsApp folder and files from previous Lab Web Worksheet 10:
  + server.py
  + catalogue.db
  + products.html
  + form1.html
  + static folder with images file

**Objectives:**

The objective of this lab is to be able to select a product from the product list and edit the product details.

**Task 1:Edit the products.html template such that the name of the product becomes a hyperlink that can link to a web form for editing the product details.**

1. Modify the content inside the <div class="text title"> as follows:

<div class="text title">

**<a href="/edit/{{ row['Name'] }}">{{ row["Name"] }}</a>**

</div>

1. Add a view function with the following signature in the server.py:

@app.route("/edit/<product>")

def edit\_product(product):

return product

When Jinja processes the template the hyperlink's url will be replaced as /edit/<product name>. When the hyperlink is clicked, the function edit\_product will be invoked with the <product\_name> passed as argument.

Perform the following test before we implement the actual code:

* When the product lists is display, the product name becomes a hyper-link
* When you click on the hyperlink, you should see a new page with just the product name
* You are now able pass an argument into a view function using a url.

**Task 2: Create the product edit web form for editing the details of the product**

1. Locate the product\_edit.html in the templates folder. It has already been created for you.
2. The product\_edit form will display the current values for the product details. You can edit these values and submit them for update in the database.
3. Study the contents of the product\_edit.html file, notice that
   1. you need to pass in an argument named data which is a dictionary object that contains the details of the product that you want to edit.
   2. the form has two submit buttons, with values "update" and "delete".
   3. the url for processing the form is /update/{{ data['name] }}.
4. Modify the edit\_product(product) function as follow:
   1. Open a database connection
   2. Execute a SQL query to return only 1 product,
   3. The code is as follow:

| @app.route("/edit/<product>")  def edit\_product(product):  try:  con= open\_DB("catalogue.db")  cur = con.execute("SELECT \* FROM Products WHERE Name=?", (product,))  row=cur.fetchone()  except Exception as e:  print(str(e))  finally:  con.close()  return render\_template("product\_edit.html",data=row) |
| --- |

1. Test that the edit product page shows the correct product details when you select the product from the product list.

**Task 3: Implement the update \_product function.**

1. After the user has filled in the product details to be modified, including the image file, he/she can click on the update button which will invoke the update\_product(product) function.
2. If the user wants to delete the product, he/she will click on the delete button which will also invoke the update\_product(product) function.
3. Add the function update\_product(product) in server.py.
4. The code is as follows:

@app.route("/update/<product>", methods=['POST'])

def update\_product(product):

file\_name=""

if "image" in request.files:

file\_obj = request.files["image"]

file\_name = file\_obj.filename

file\_obj.save("static/images/"+file\_name)

con = open\_DB("catalogue.db")

if request.form["submit"] == "update" and file\_name == "":

cur=con.execute("UPDATE Products SET Name=?, Description=?, Price=? WHERE Name=?",

(request.form['name'], request.form['description'], request.form['price'],product)

)

elif request.form["submit"] == "update" and file\_name != "":

cur=con.execute("UPDATE Products SET Name=?, Description=?, Price=?, Image=? WHERE Name=?",

(request.form['name'],request.form['description'],request.form['price'],file\_name, product)

)

elif request.form["submit"] == "delete":

cur=con.execute("DELETE FROM Products WHERE Name = ?",(product,))

con.commit()

con.close()

return redirect("/")