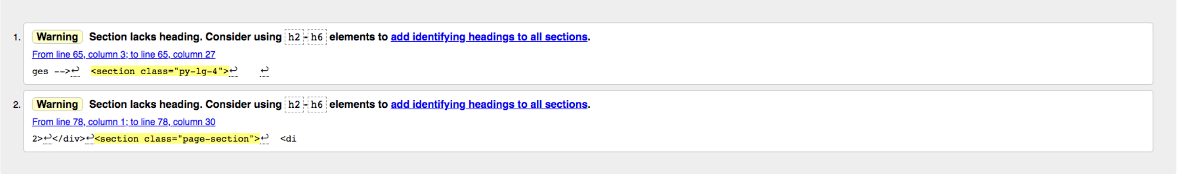
Automated Testing

W3 HTML VALIDATOR

As the code using Flask is not rendered, the way I tested the code was to use google developer tools – and view source. Copy the code into the HTML validator.

Home Page (logged out) html

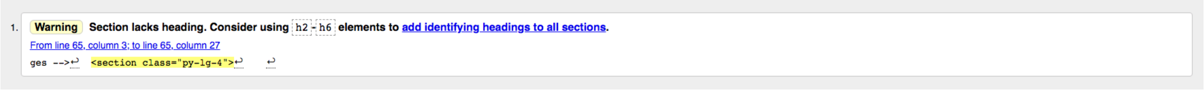
The homepage passed with only 2 warnings, which are fine as part of the original library template.



1. Warning Section lacks heading. Consider using `h2` - `h6` elements to add identifying headings to all sections.
   From line 65, column 3; to line 65, column 27
   `ges -->↩  <section class="py-lg-4">↩    ↩`
2. Warning Section lacks heading. Consider using `h2` - `h6` elements to add identifying headings to all sections.
   From line 78, column 1; to line 78, column 30
   `2>↩</div>↩<section class="page-section">↩  <di`

Log in Page

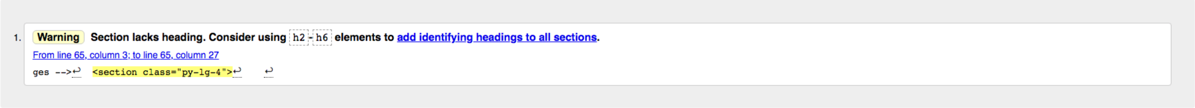
The log in page passed with only 1 warning, which is fine as part of the original library template.



1. Warning Section lacks heading. Consider using `h2` - `h6` elements to add identifying headings to all sections.
   From line 65, column 3; to line 65, column 27
   `ges -->↩  <section class="py-lg-4">↩    ↩`

New User Page

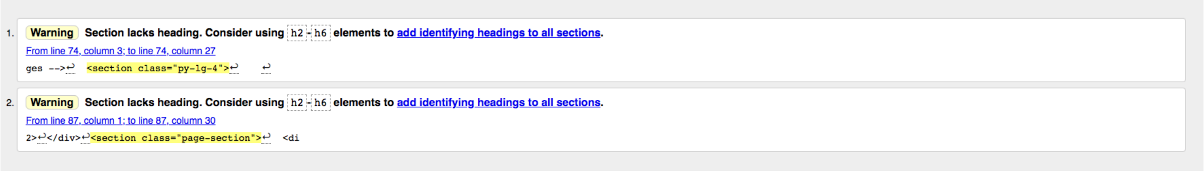
The new user page passed with only 1 warning, which is fine as part of the original library template.



1. Warning Section lacks heading. Consider using `h2` - `h6` elements to add identifying headings to all sections.
   From line 65, column 3; to line 65, column 27
   `ges -->↩  <section class="py-lg-4">↩    ↩`

Home Page (logged in)

The homepage passed with only 2 warnings, which are fine as part of the original library template.

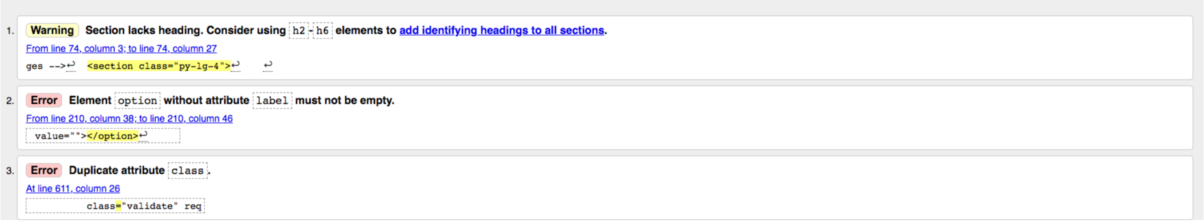


1. Warning Section lacks heading. Consider using `h2` - `h6` elements to add identifying headings to all sections.
   From line 74, column 3; to line 74, column 27
   `ges -->↩  <section class="py-lg-4">↩    ↩`
2. Warning Section lacks heading. Consider using `h2` - `h6` elements to add identifying headings to all sections.
   From line 87, column 1; to line 87, column 30
   `2>↩</div>↩<section class="page-section">↩  <di`

Add post page

The add post page failed with 2 errors, and 1 warning. The warning is fine as part of original library template. But the errors needed fixing.

I removed the duplicate attribute. However, for value=”” this has to be kept like this, in order to display a blank dropdown menu. I checked multiple websites suggesting to add a space or &nbsp; but this threw errors. According to my tutorials this is the correct procedure, so I left the error – as in this case it is required.



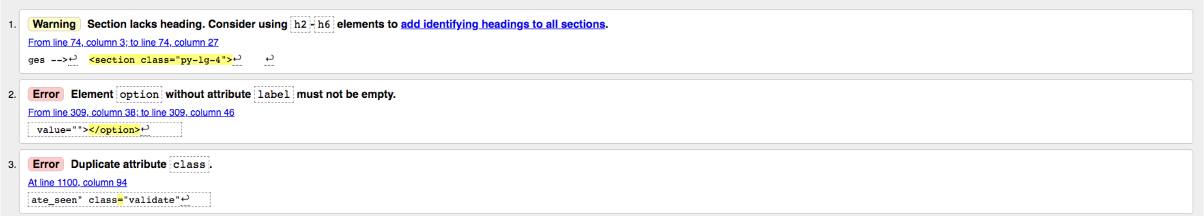
```
1. Warning  Section lacks heading. Consider using h2 - h6 elements to add identifying headings to all sections.
   From line 74, column 3; to line 74, column 27
   ges -->↩  <section class="py-lg-4">↩    ↩

2. Error  Element option without attribute label must not be empty.
   From line 210, column 38; to line 210, column 46
    value=""></option>↩

3. Error  Duplicate attribute class.
   At line 611, column 26
               class="validate" req
```

Edit post page

The edit post page has the same errors and warnings as ‘add post page’ above, therefore I have removed the duplicate value only, as others do not need changing – they are correct.



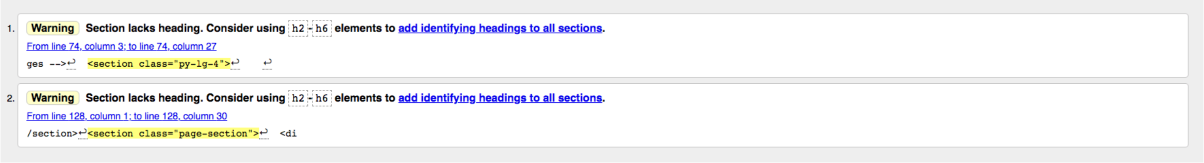
```
1. Warning  Section lacks heading. Consider using h2 - h6 elements to add identifying headings to all sections.
   From line 74, column 3; to line 74, column 27
   ges -->↩  <section class="py-lg-4">↩    ↩

2. Error  Element option without attribute label must not be empty.
   From line 309, column 38; to line 309, column 46
    value=""></option>↩

3. Error  Duplicate attribute class.
   At line 1100, column 94
   ate_seen" class="validate"↩
```

Account Settings Page

The account settings page passed with only 2 warnings, which are fine as part of the original library template.

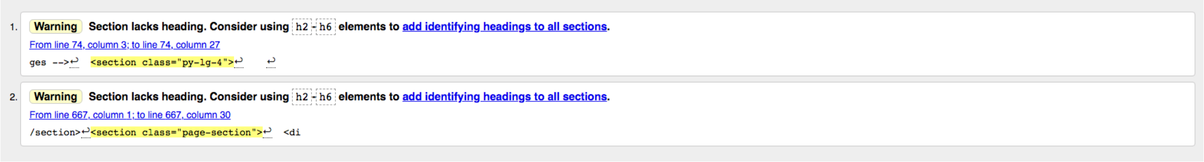


```
1. Warning  Section lacks heading. Consider using h2 - h6 elements to add identifying headings to all sections.
   From line 74, column 3; to line 74, column 27
   ges -->↩  <section class="py-lg-4">↩    ↩

2. Warning  Section lacks heading. Consider using h2 - h6 elements to add identifying headings to all sections.
   From line 128, column 1; to line 128, column 30
   /section>↩<section class="page-section">↩  <di
```

Modify Cars Page

The account settings page passed with only 2 warnings, which are fine as part of the original library template.

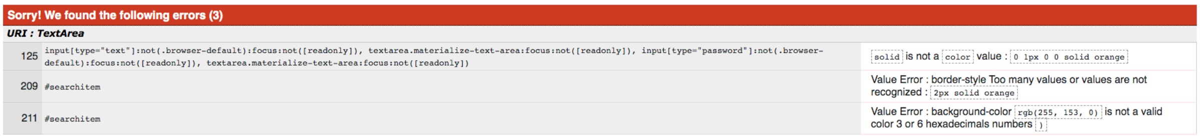


```
1. Warning  Section lacks heading. Consider using h2 - h6 elements to add identifying headings to all sections.
   From line 74, column 3; to line 74, column 27
   ges -->↩  <section class="py-lg-4">↩    ↩

2. Warning  Section lacks heading. Consider using h2 - h6 elements to add identifying headings to all sections.
   From line 667, column 1; to line 667, column 30
   /section>↩<section class="page-section">↩  <di
```

W3 CSS VALIDATOR

Style.css

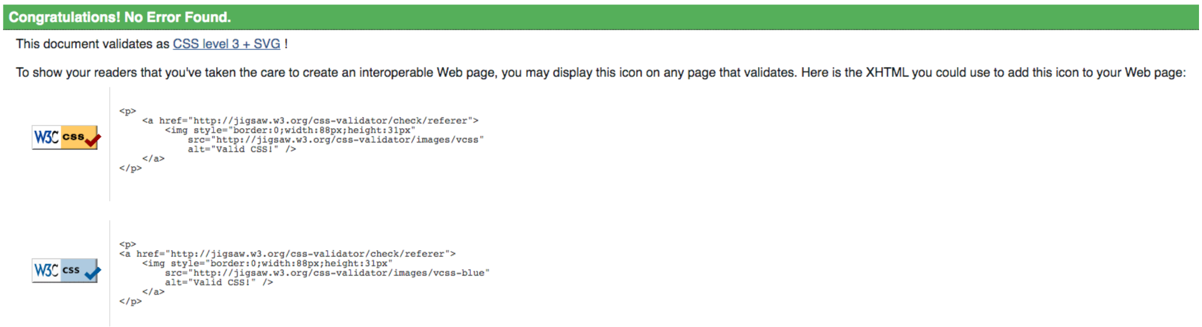


Sorry! We found the following errors (3)

URI : TextArea

125	`input[type="text"]:not(.browser-default):focus:not([readonly]), textarea.materialize-text-area:focus:not([readonly]), input[type="password"]:not(.browser-default):focus:not([readonly]), textarea.materialize-text-area:focus:not([readonly])`	`solid` is not a `color` value : `0 1px 0 0 solid orange`
209	`#searchitem`	Value Error : border-style Too many values or values are not recognized : `2px solid orange`
211	`#searchitem`	Value Error : background-color `rgb(255, 153, 0)` is not a valid color 3 or 6 hexadecimals numbers `)`

The style.css had a few errors, which have been corrected and now passes, see below.



Congratulations! No Error Found.

This document validates as CSS level 3 + SVG !

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

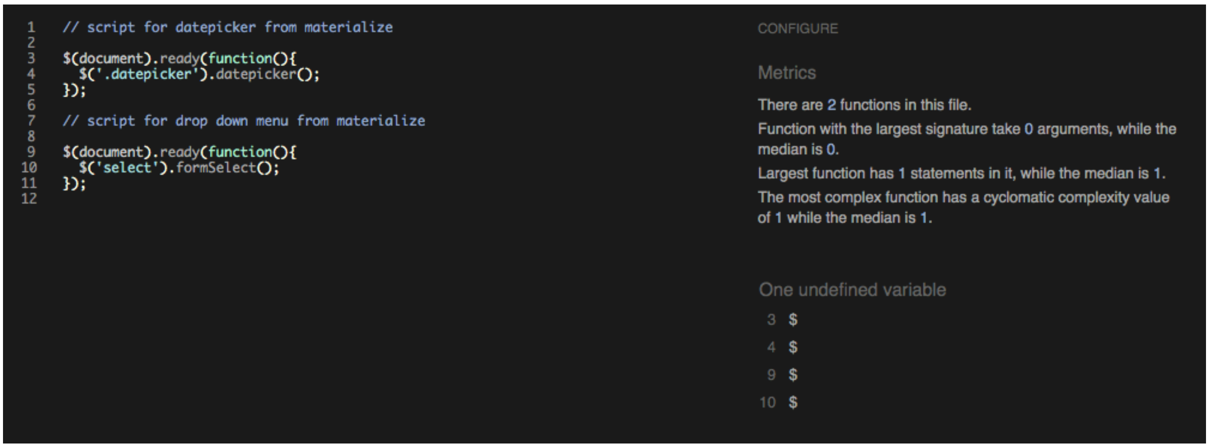
```
<p>
    <a href="http://jigsaw.w3.org/css-validator/check/referer">
        <img style="border:0;width:88px;height:31px"
            src="http://jigsaw.w3.org/css-validator/images/vcss"
            alt="Valid CSS!" />
    </a>
</p>
```

```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referer">
    <img style="border:0;width:88px;height:31px"
        src="http://jigsaw.w3.org/css-validator/images/vcss-blue"
        alt="Valid CSS!" />
    </a>
</p>
```

JS HINT VALIDATOR

Js.script

There isn't too much Javascript / JQuery – but it passes with no issues.



```
// script for datepicker from materialize

$(document).ready(function(){
  $('.datepicker').datepicker();
});

// script for drop down menu from materialize

$(document).ready(function(){
  $('select').formSelect();
});
```

CONFIGURE

Metrics

There are 2 functions in this file.

Function with the largest signature take 0 arguments, while the median is 0.

Largest function has 1 statements in it, while the median is 1.

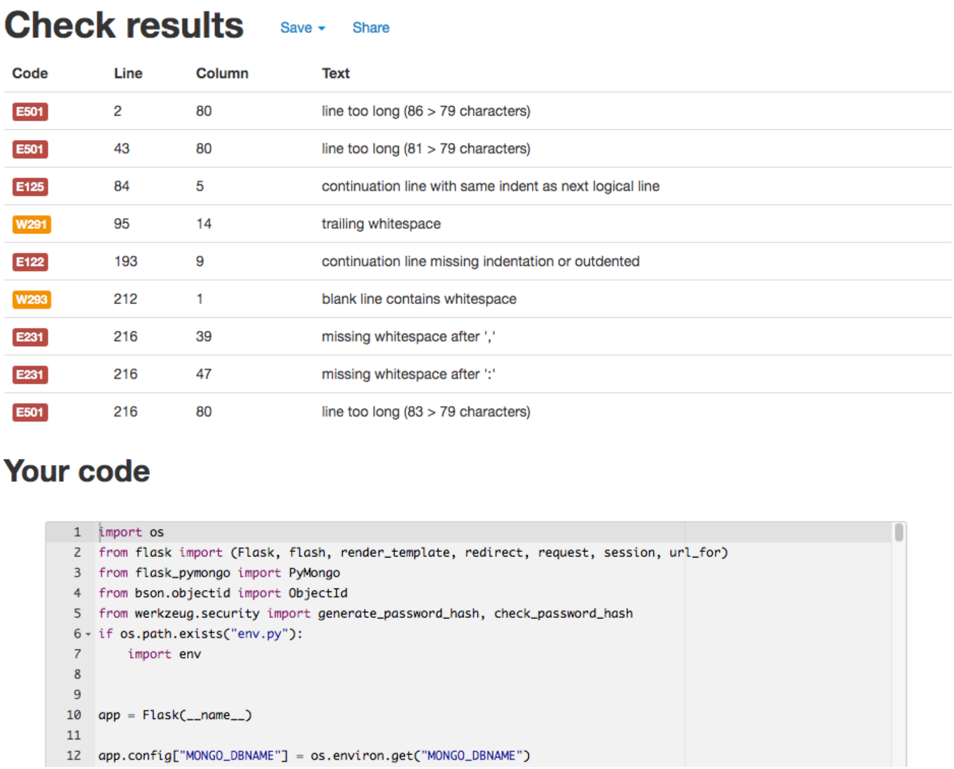
The most complex function has a cyclomatic complexity value of 1 while the median is 1.

One undefined variable

```
3  $
4  $
9  $
10 $
```

PEP8 ONLINE VALIDATOR

Run.py



Check results

Code	Line	Column	Text
E501	2	80	line too long (86 > 79 characters)
E501	43	80	line too long (81 > 79 characters)
E125	84	5	continuation line with same indent as next logical line
W291	95	14	trailing whitespace
E122	193	9	continuation line missing indentation or outdented
W293	212	1	blank line contains whitespace
E231	216	39	missing whitespace after ','
E231	216	47	missing whitespace after ':'
E501	216	80	line too long (83 > 79 characters)

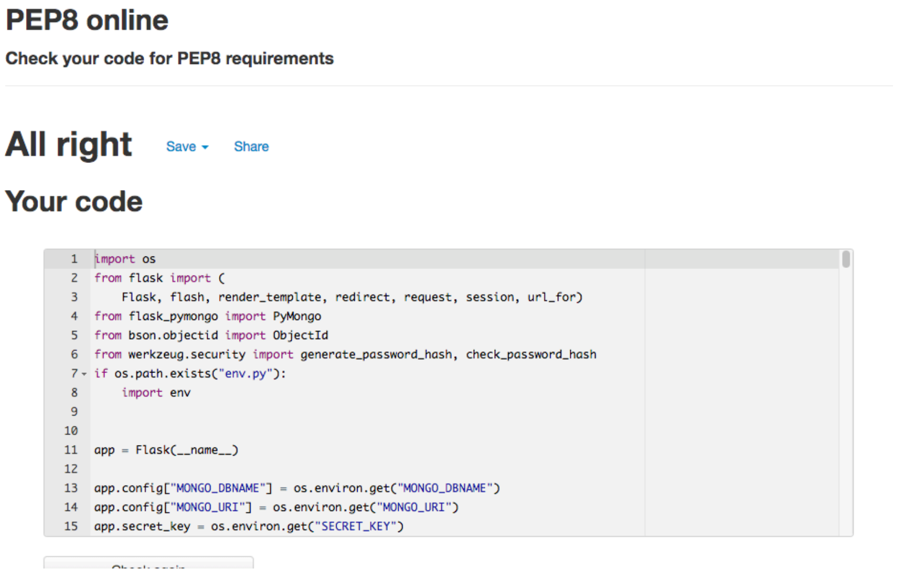
Your code

```
import os
from flask import (Flask, flash, render_template, redirect, request, session, url_for)
from flask_pymongo import PyMongo
from bson.objectid import ObjectId
from werkzeug.security import generate_password_hash, check_password_hash
if os.path.exists("env.py"):
    import env


app = Flask(__name__)

app.config["MONGO_DBNAME"] = os.environ.get("MONGO_DBNAME")
```

There were multiple errors, which have all been fixed, and now the python is PEP8 compliant as per below.



PEP8 online

Check your code for PEP8 requirements

All right

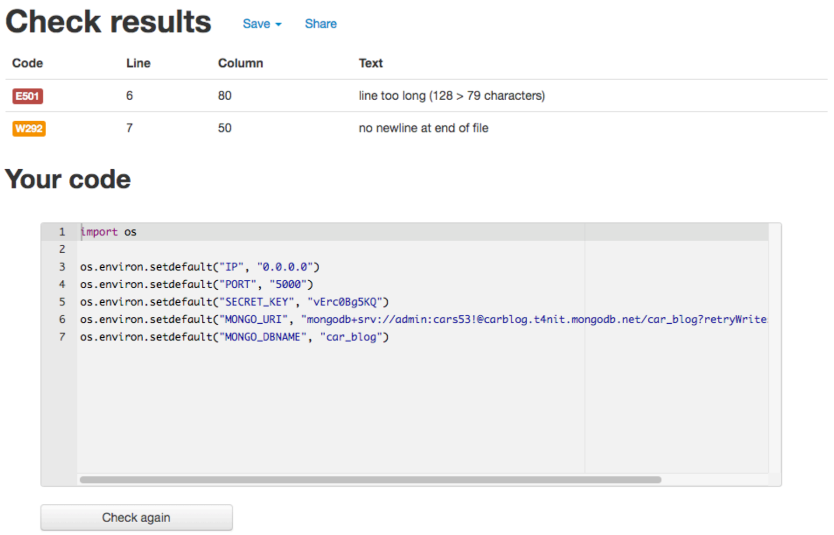
Your code

```
import os
from flask import (
    Flask, flash, render_template, redirect, request, session, url_for)
from flask_pymongo import PyMongo
from bson.objectid import ObjectId
from werkzeug.security import generate_password_hash, check_password_hash
if os.path.exists("env.py"):
    import env


app = Flask(__name__)

app.config["MONGO_DBNAME"] = os.environ.get("MONGO_DBNAME")
app.config["MONGO_URI"] = os.environ.get("MONGO_URI")
app.secret_key = os.environ.get("SECRET_KEY")
```

env.py



Check results

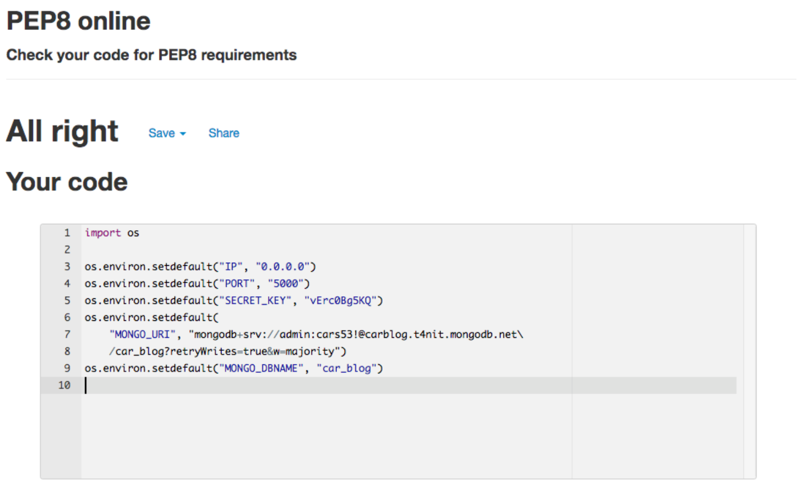
Code	Line	Column	Text
E501	6	80	line too long (128 > 79 characters)
W292	7	50	no newline at end of file

Your code

```
import os

os.environ.setdefault("IP", "0.0.0.0")
os.environ.setdefault("PORT", "5000")
os.environ.setdefault("SECRET_KEY", "vErc0Bg5KQ")
os.environ.setdefault("MONGO_URI", "mongodb+srv://admin:cars53!@carblog.t4nit.mongodb.net/car_blog?retryWrite
os.environ.setdefault("MONGO_DBNAME", "car_blog")
```

For the env.py file, which is not too big, there are a couple of errors. There were no brackets after MONGO_URI so I tried to split the URL using \ as per guidelines.



PEP8 online

Check your code for PEP8 requirements

All right

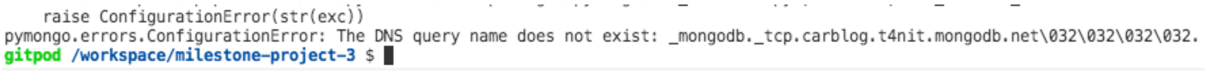
Your code

```
import os

os.environ.setdefault("IP", "0.0.0.0")
os.environ.setdefault("PORT", "5000")
os.environ.setdefault("SECRET_KEY", "vErc0Bg5KQ")
os.environ.setdefault(
    "MONGO_URI", "mongodb+srv://admin:cars53!@carblog.t4nit.mongodb.net\
    /car_blog?retryWrites=true&w=majority")
os.environ.setdefault("MONGO_DBNAME", "car_blog")

```

Although it passes PEP8 Online, it threw the below error and failed to launch.



```
    raise ConfigurationError(str(exc))
pymongo.errors.ConfigurationError: The DNS query name does not exist: _mongodb._tcp.carblog.t4nit.mongodb.net\032\032\032\032.
gitpod /workspace/milestone-project-3 $ 
```

Therefore, I had to revert back to the full URL in order to allow the project to run. Which is the only part that will not pass.