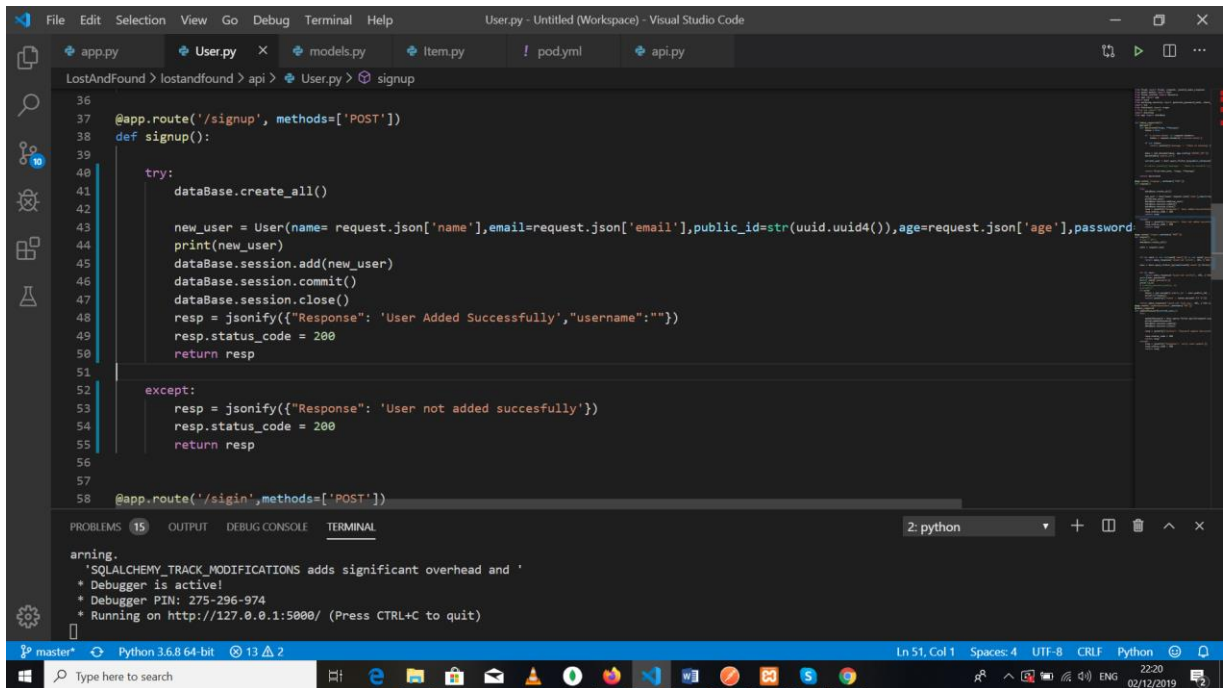


Lost and found

I have given a task to make a backend using Restful APIs for a start-up who are making a lost and found system which have many feature I have done this using following approaches :-

- **User's registrations**

Is done by applying following implantation's

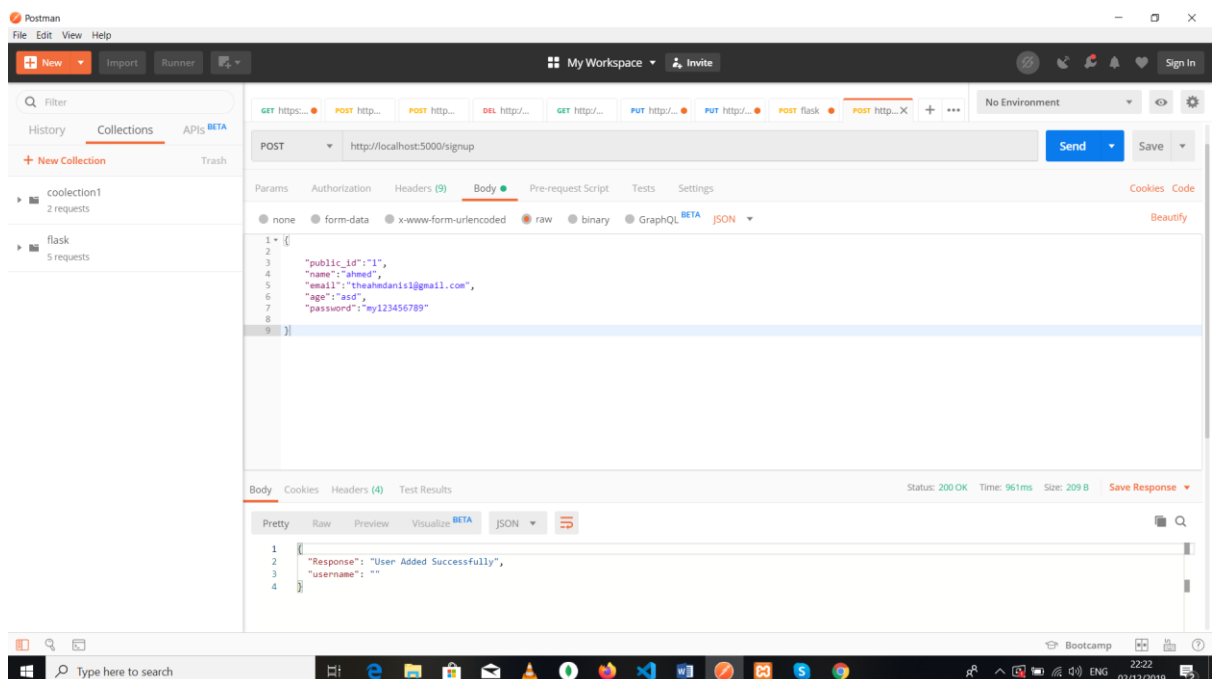


```
36
37 @app.route('/signup', methods=['POST'])
38 def signup():
39
40     try:
41         DataBase.create_all()
42
43         new_user = User(name= request.json['name'],email=request.json['email'],public_id=str(uuid.uuid4()),age=request.json['age'],password
44         print(new_user)
45         DataBase.session.add(new_user)
46         DataBase.session.commit()
47         DataBase.session.close()
48         resp = jsonify({"Response": "User Added Successfully","username":""})
49         resp.status_code = 200
50         return resp
51
52     except:
53         resp = jsonify({"Response": "User not added succesfully"})
54         resp.status_code = 200
55         return resp
56
57
58 @app.route('/signin',methods=['POST'])
```

PROBLEMS (15) OUTPUT DEBUG CONSOLE TERMINAL

Warning.
'SQLALCHEMY_TRACK_MODIFICATIONS' adds significant overhead and '
* Debugger is active!
* Debugger PIN: 275-296-974
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

And the output the code is as follow



Postman

GET https... POST http... POST http... DEL http... GET http... PUT http... PUT http... POST flask POST http... X + ... No Environment

POST http://localhost:5000/signup

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON Beautify

```
1 {
2   "public_id": "1",
3   "name": "ahmed",
4   "email": "theahmedanis@gmail.com",
5   "age": "asd",
6   "password": "my123456789"
7 }
8
9 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 961ms Size: 209 B Save Response

Pretty Raw Preview Visualize BETA JSON Beautify

```
1 {
2   "Response": "User Added Successfully",
3   "username": ""
4 }
```

- **User login**

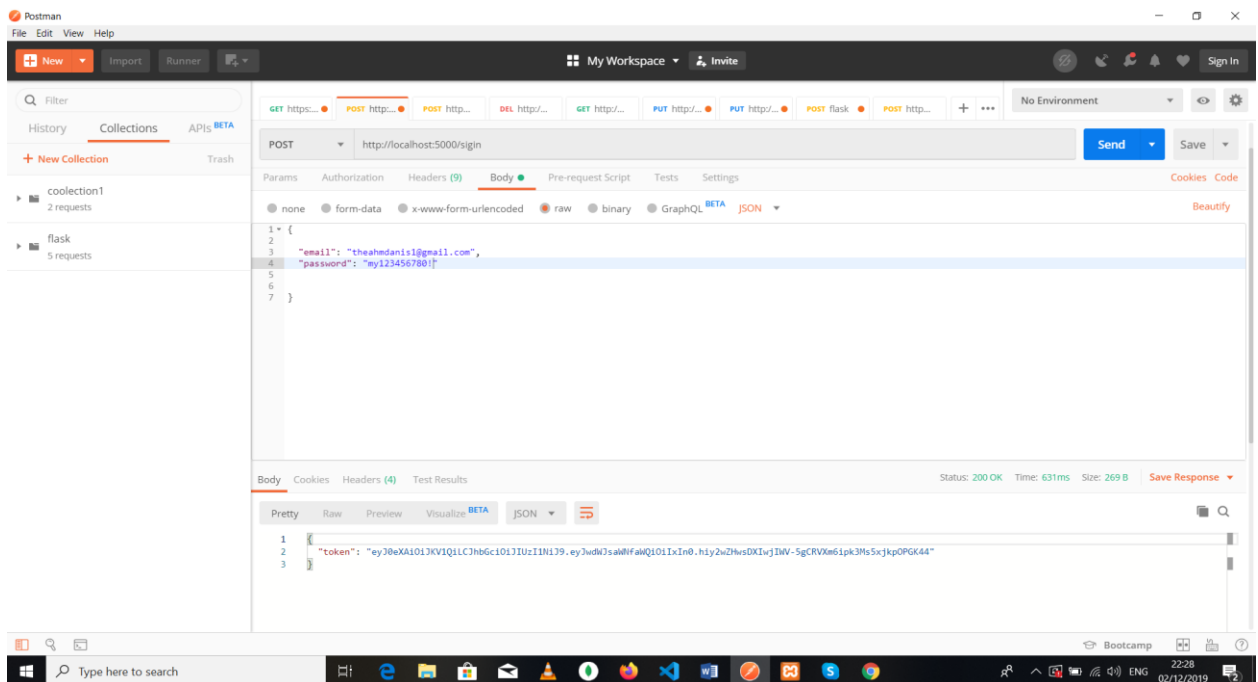
User login is done by following implemtnaion

```
File Edit View Go Debug Terminal Help User.py - Untitled (Workspace) Visual Studio Code
app.py User.py x models.py Item.py ! pod.yml api.py
LostAndFound > lostandfound > api > User.py > sign
55 |         return resp
56 |
57 |
58 | @app.route('/signin', methods=['POST'])
59 | def signin():
60 |     DataBase.create_all()
61 |     auth = request.json
62 |     if not auth or not str(auth['email']) or not auth['password']:
63 |         return make_response('Could not verify', 401, {'WWW-Authenticate': 'Basic realm="Login required!"'})
64 |
65 |     user = User.query.filter_by(email=auth['email']).first()
66 |     if not user:
67 |         return make_response('Could not verify2', 401, {'WWW-Authenticate': 'Basic realm="Login required!"'})
68 |     a=str(user.password)
69 |     b=str( auth['password'])
70 |
71 |
72 |     if a==b:
73 |         token = jwt.encode({'public_id' : user.public_id }, app.config['SECRET_KEY'])
74 |         |
75 |         return jsonify({'token' : token.decode('UTF-8')})
76 |
77 |     return make_response('Could not find you', 401, {'WWW-Authenticate': 'Basic realm="Login required!"'})
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: python + □ ✕ ^

be disabled by default in the future. Set it to True or False to suppress this warning.
*SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and '
* Debugger is active!
* Debugger PIN: 275-296-974
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

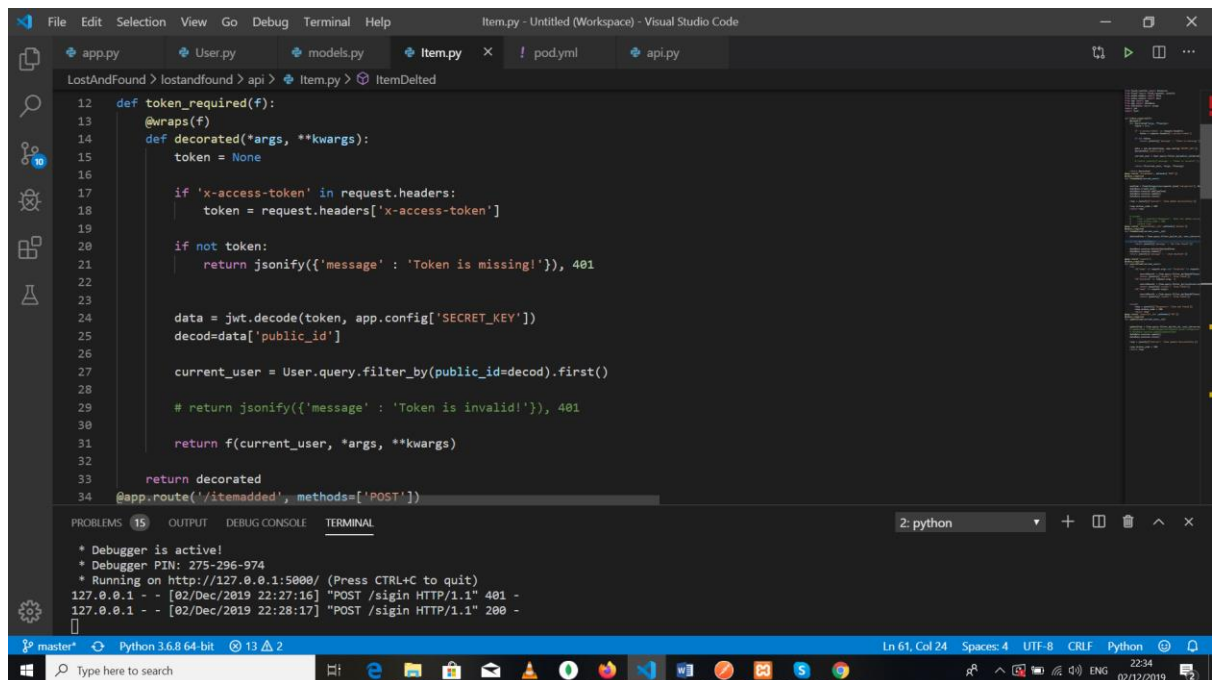
And the output is :



It generates Jwt token.

- **Authentication**

For Adding item the system ensure that user is logged in and genuine user for applying this I have make a function name required token



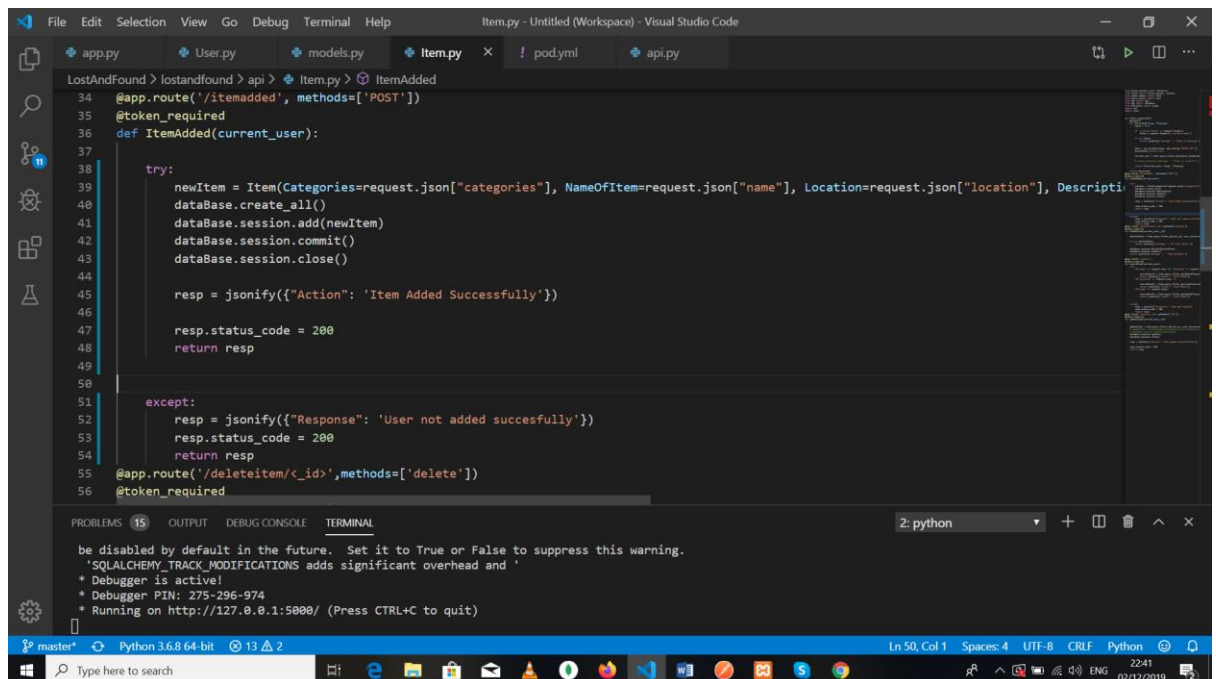
```
12 def token_required(f):
13     @wraps(f)
14     def decorated(*args, **kwargs):
15         token = None
16
17         if 'x-access-token' in request.headers:
18             token = request.headers['x-access-token']
19
20         if not token:
21             return jsonify({'message': 'Token is missing!'}), 401
22
23         data = jwt.decode(token, app.config['SECRET_KEY'])
24         decodedata['public_id']
25
26         current_user = User.query.filter_by(public_id=decodedata['public_id']).first()
27
28         # return jsonify({'message': 'Token is invalid!'}), 401
29
30         return f(current_user, *args, **kwargs)
31
32     return decorated
33
34 @app.route('/itemadded', methods=['POST'])
```

Terminal output:

```
* Debugger is active!
* Debugger PIN: 275-296-974
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [02/Dec/2019 22:27:16] "POST /signin HTTP/1.1" 401 -
127.0.0.1 - - [02/Dec/2019 22:28:17] "POST /signin HTTP/1.1" 200 -
```

- **Item Addition**

After authentication of user system allow end user to add item

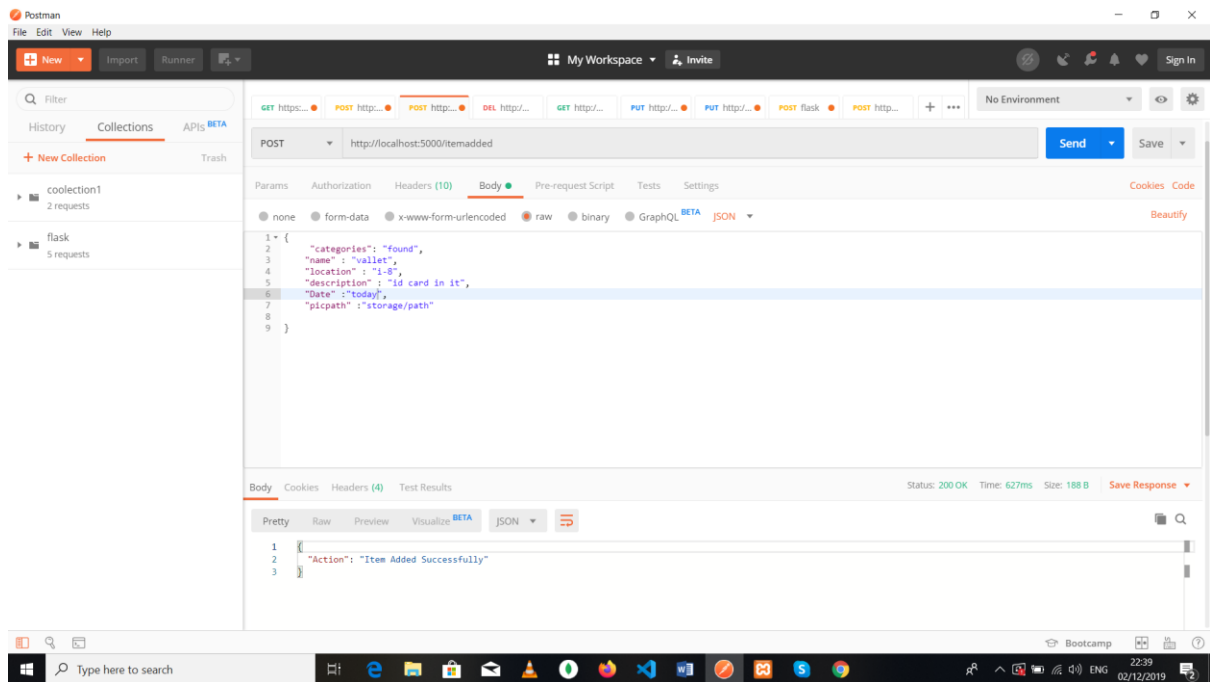


```
34 @app.route('/itemadded', methods=['POST'])
35 @token_required
36 def ItemAdded(current_user):
37
38     try:
39         newItem = Item(Categories=request.json["categories"], NameOfItem=request.json["name"], Location=request.json["location"], Description=request.json["description"])
40         DataBase.create_all()
41         DataBase.session.add(newItem)
42         DataBase.session.commit()
43         DataBase.session.close()
44
45         resp = jsonify({"Action": 'Item Added Successfully'})
46
47         resp.status_code = 200
48         return resp
49
50     except:
51         resp = jsonify({"Response": 'User not added succesfully'})
52         resp.status_code = 200
53         return resp
54
55 @app.route('/deleteitem/<id>', methods=['delete'])
56 @token_required
```

Terminal output:

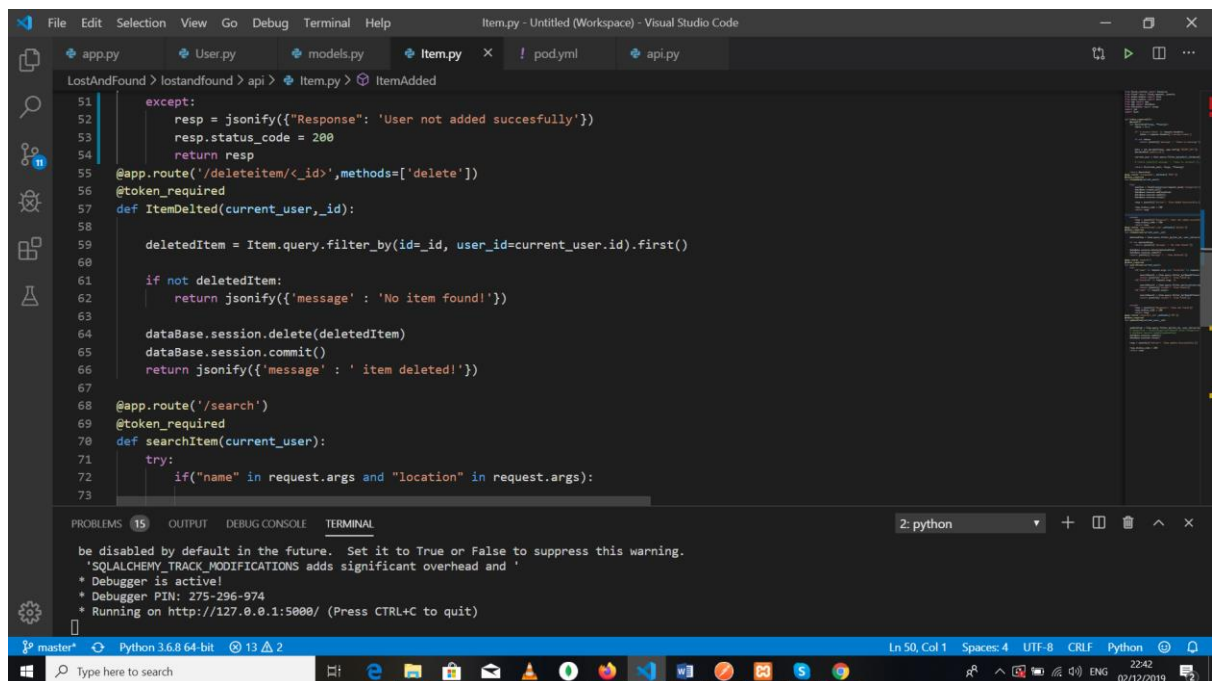
```
be disabled by default in the future. Set it to True or False to suppress this warning.
'SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and '
* Debugger is active!
* Debugger PIN: 275-296-974
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

The output of this end point is :

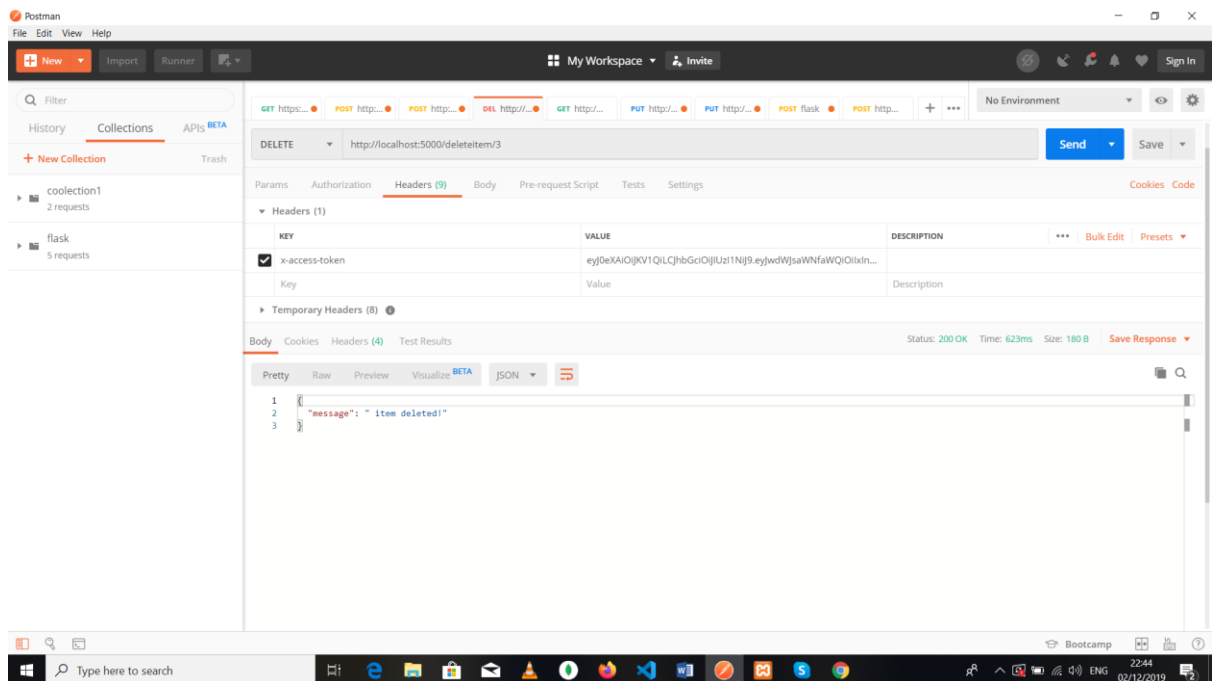


- **Item Deleted**

System allow end user to delete the item which is added by him

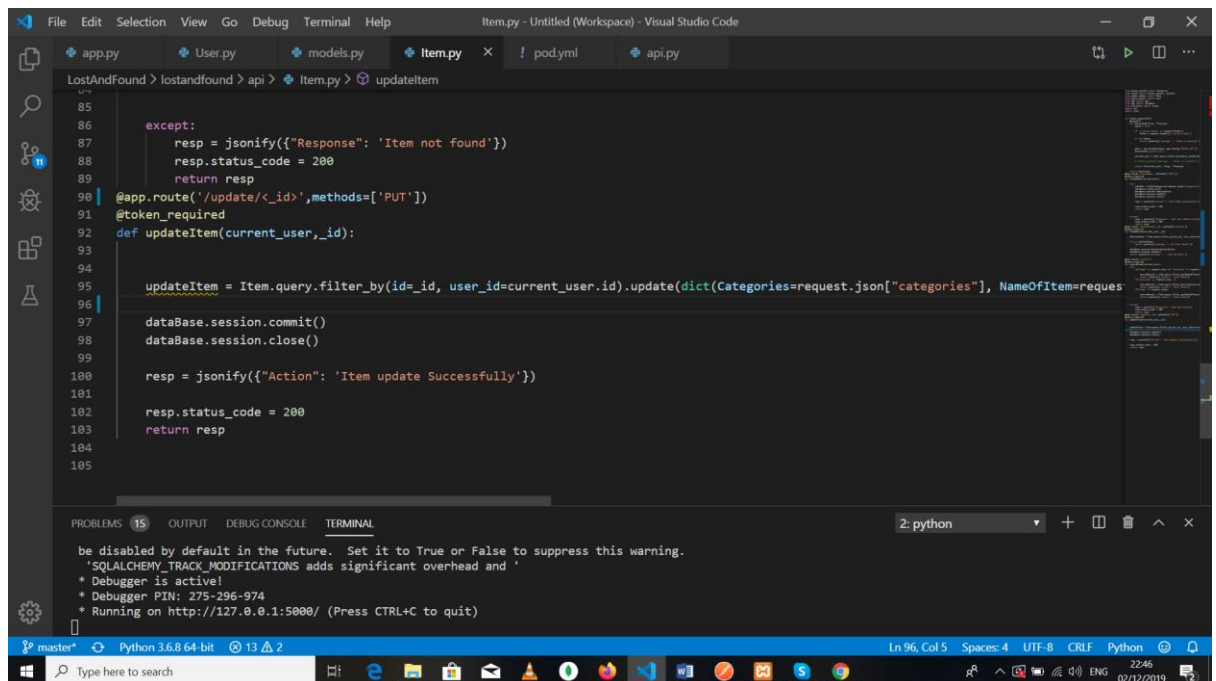


And the output of end point is:

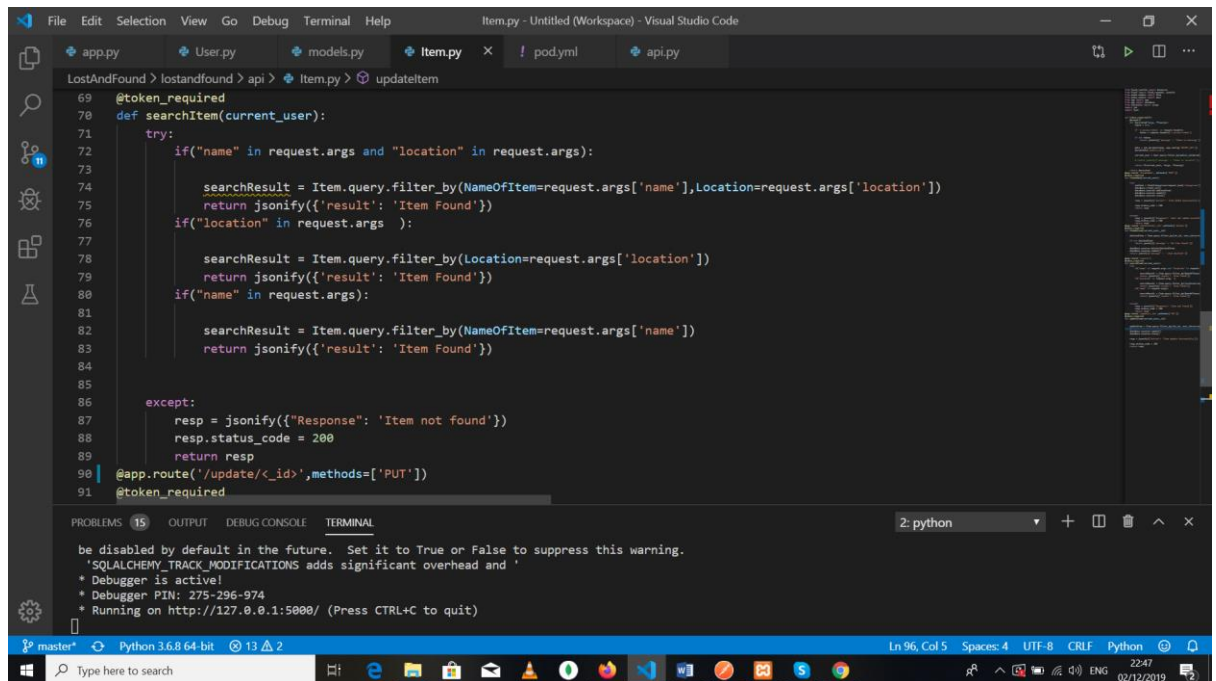


- **Update**

End user can update



As well as search the item



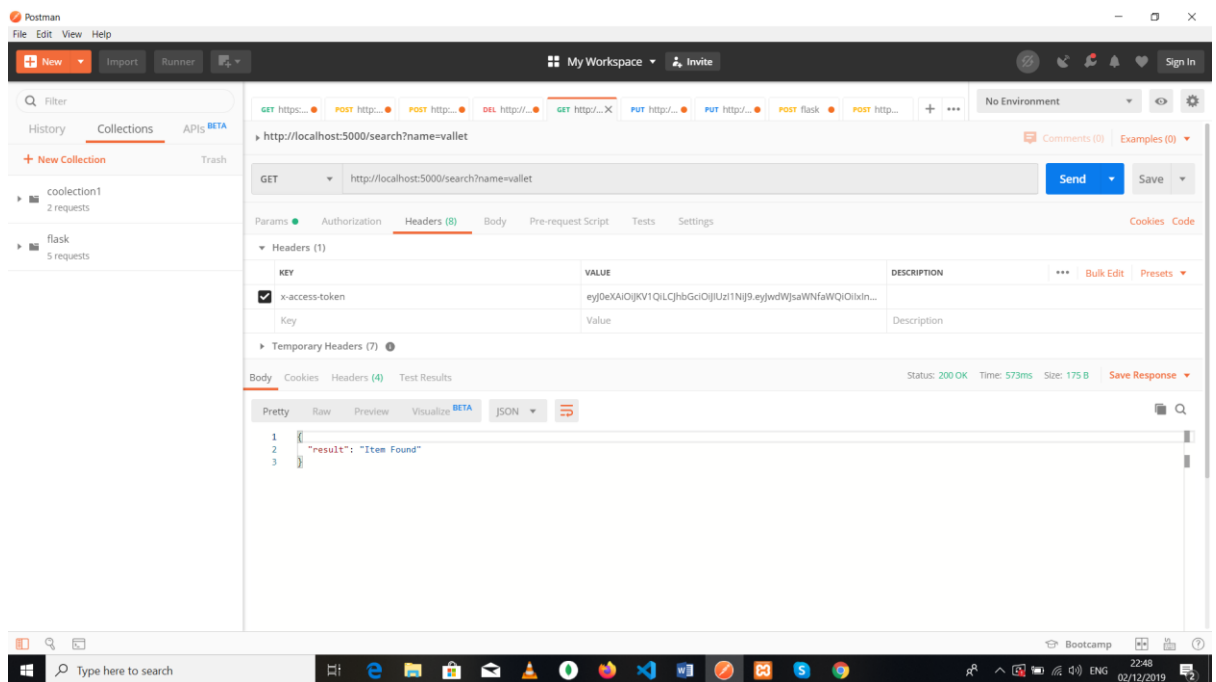
The screenshot shows the Visual Studio Code editor with a Python file named `Item.py`. The code defines a `searchItem` function that takes `current_user` as an argument. It uses `request.args` to get 'name' and 'location' parameters. The function filters a database query based on these parameters and returns a JSON response. If both parameters are provided, it filters by both. If only 'location' is provided, it filters by location. If only 'name' is provided, it filters by name. If neither is provided, it returns 'Item not found'.

```
69 @token_required
70 def searchItem(current_user):
71     try:
72         if("name" in request.args and "location" in request.args):
73
74             searchResult = Item.query.filter_by(NameOfItem=request.args['name'],Location=request.args['location'])
75             return jsonify({'result': 'Item Found'})
76         if("location" in request.args ):
77
78             searchResult = Item.query.filter_by(Location=request.args['location'])
79             return jsonify({'result': 'Item Found'})
80         if("name" in request.args):
81
82             searchResult = Item.query.filter_by(NameOfItem=request.args['name'])
83             return jsonify({'result': 'Item Found'})
84
85     except:
86         resp = jsonify({"Response": 'Item not found'})
87         resp.status_code = 200
88         return resp
89
90 @app.route('/update/<id>',methods=['PUT'])
91 @token_required
```

The terminal at the bottom shows the following output:

```
be disabled by default in the future. Set it to True or False to suppress this warning.
'SQLALCHEMY_TRACK_MODIFICATIONS' adds significant overhead and
* Debugger is active!
* Debugger PIN: 275-296-974
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Search result :

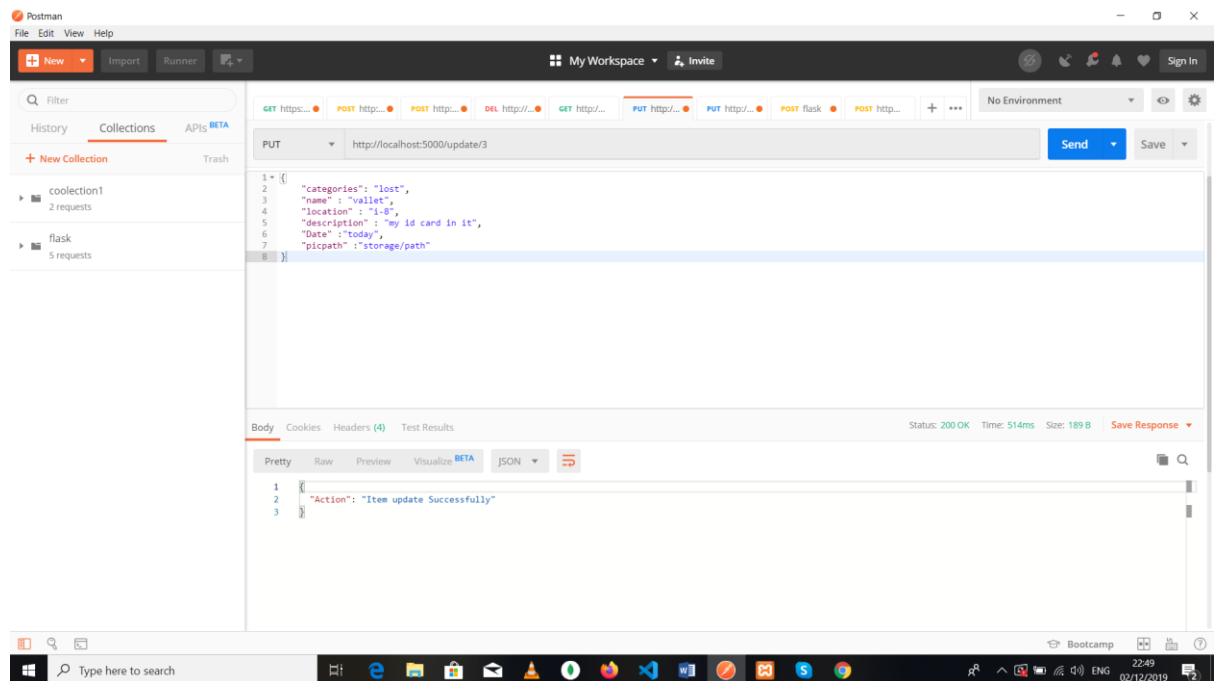


The screenshot shows the Postman application with a GET request to `http://localhost:5000/search?name=vallet`. The request headers include an `x-access-token`. The response status is 200 OK, and the body is a JSON object: `{ "result": "Item Found" }`.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> x-access-token	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50b3R5IiwiaWF0Ij0iMTUzOTI1MjE5LjE5In0=	
Key	Value	Description

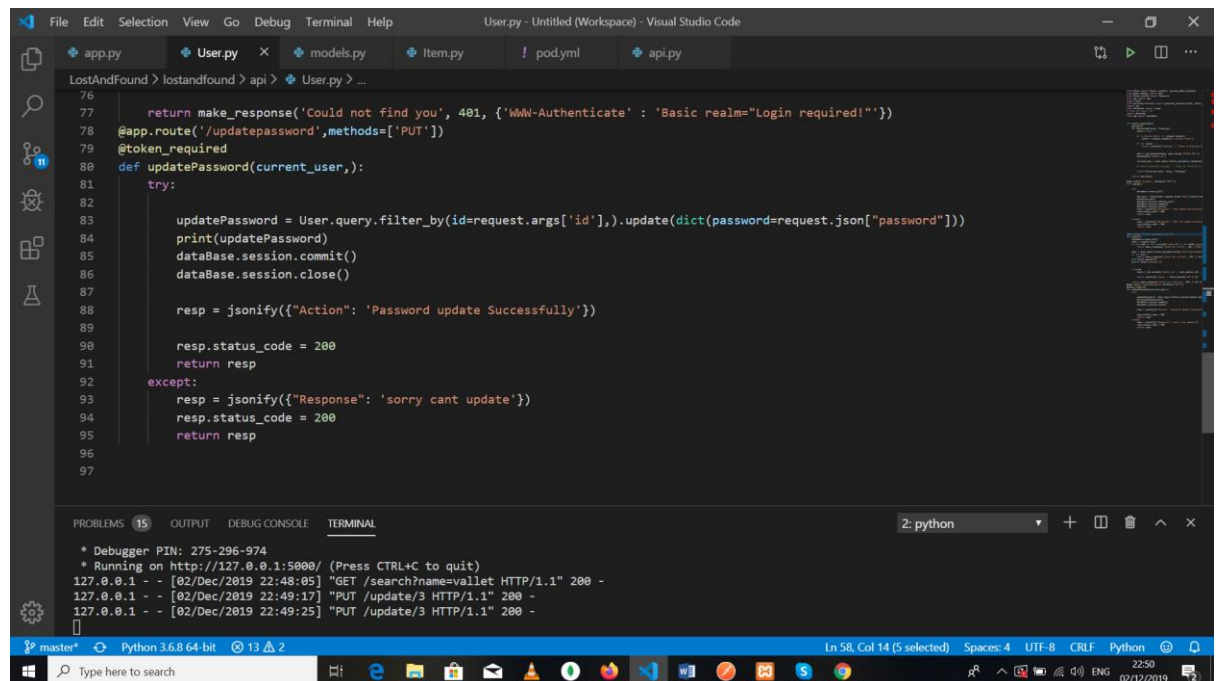
The response body is shown in the 'Body' tab, displaying the JSON result: `{ "result": "Item Found" }`.

Update result:

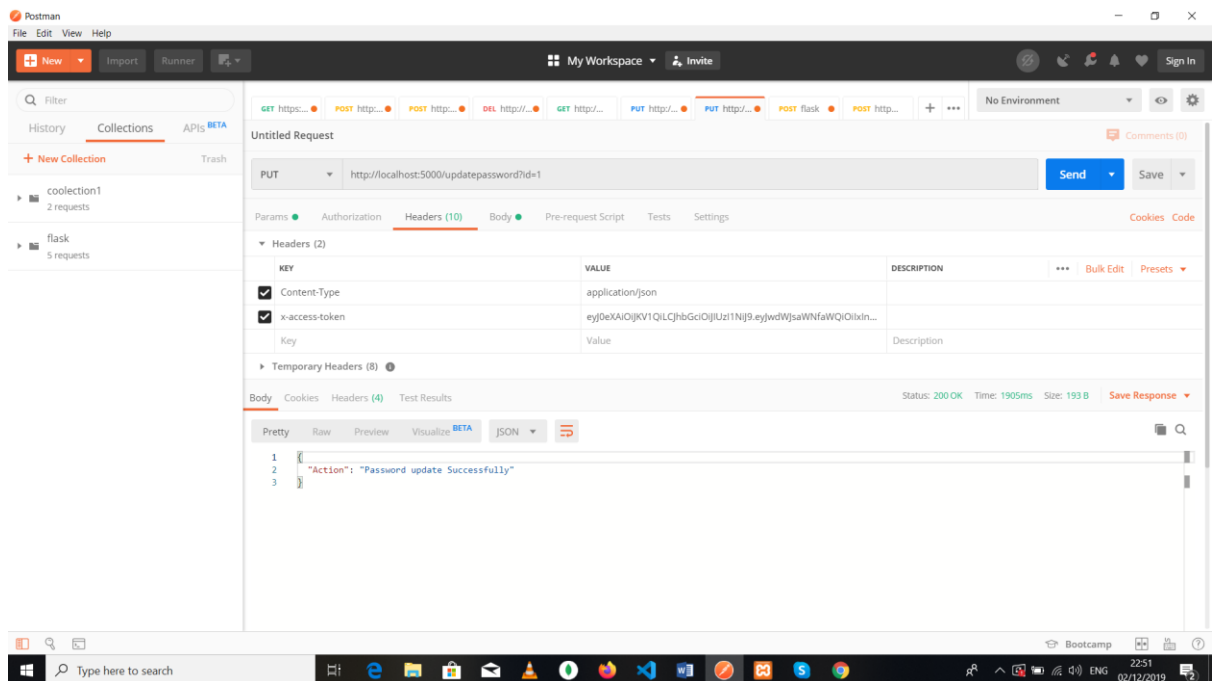


- **Change Password**

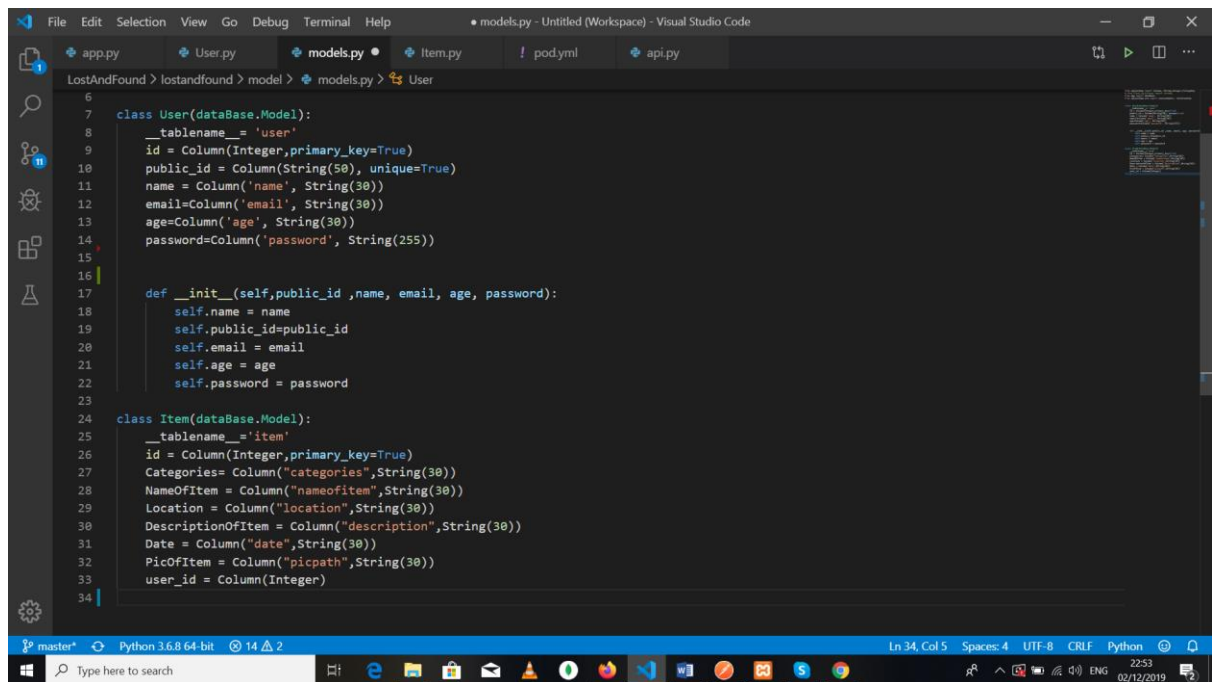
User can change his password



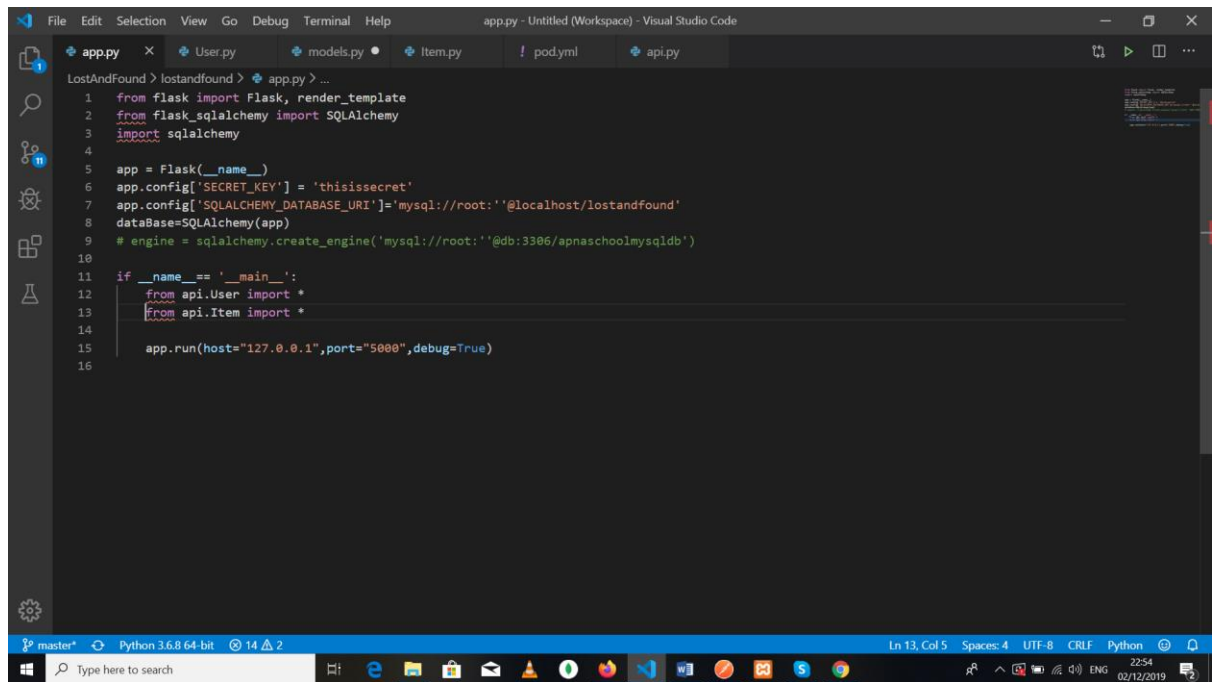
The output is :



- The database model



- The main file



```
File Edit Selection View Go Debug Terminal Help app.py - Untitled (Workspace) - Visual Studio Code
app.py x User.py models.py Item.py ! pod.yml api.py
LostAndFound > lostandfound > app.py > ...
1 from flask import Flask, render_template
2 from flask_sqlalchemy import SQLAlchemy
3 import sqlalchemy
4
5 app = Flask(__name__)
6 app.config['SECRET_KEY'] = 'thisissecret'
7 app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:' + '@localhost/lostandfound'
8 database=SQLAlchemy(app)
9 # engine = sqlalchemy.create_engine('mysql://root:' + '@db:3306/apnaschoolmysqladb')
10
11 if __name__ == '__main__':
12     from api.User import *
13     from api.Item import *
14
15     app.run(host="127.0.0.1", port="5000", debug=True)
16
```

master Python 3.6.8 64-bit 14 2 Ln 13, Col 5 Spaces: 4 UTF-8 CRLF Python 22:54 02/12/2019