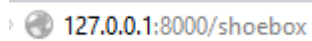


Piotr Bogun | Chris Vick

ShoeBox Application

CSC308

The application is very simple and easy to use. There are more features that I would like to add to it when I have more time. Each view contains an explanation of the application similar to what this readme does.



- **Run the Web2Py application after you put the shoebox directory into your web2py/applications folder. Run localhost/shoebox as shown**
-

ShoeBox v1.0 Complete!

Author: Peter Bogun

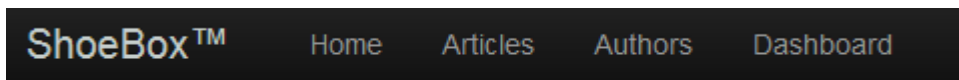
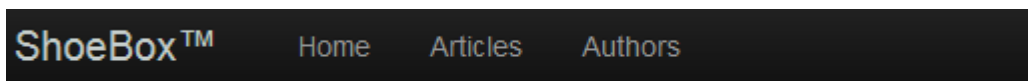


The basic ShoeBox milestones have been completed:

- 2 Tables
- 4+ Controllers
- Views for Each Action (and a text explanation)
- Permissions

While there are a few ways to optimise my work and many cool features that I want to implement to learn, I have other projects to do! I have learned many things from this class and by working with the Web2Py framework that this application does not showcase. I have in another project added ways to link live streams from justin.tv and twitch.tv along with live chat into the views. I will either continue adding features to this project since it is fun building these projects or I might make new ones. Either way I am glad I was introduced to Web2Py and I really want to improve using these tools.

- **The front page posts are handled automatically through the dashboard view.**



- The dashboard view is only available to administrators
- Login with: peter.bogun@example.com | password = "test"

Featuredposts

[+Add Record](#) [Search](#) [Clear](#) 3 records found

Id	Title	Content	Postauthor	Postdate	
6	ShoeBox Featured ...	Any ShoeBox Admin...	Peter Bogun (1)	2014-11-20	View Edit Delete
7	ShoeBox v1.0 Comp...	The basic ShoeBox...	Peter Bogun (1)	2014-11-21	View Edit Delete
8	Old Test	Old Test Stuff	Dan Thom (54)	2014-11-19	View Edit Delete

Export: [CSV](#) [CSV \(hidden cols\)](#) [HTML](#) [JSON](#) [TSV \(Spreadsheets\)](#) [TSV \(Spreadsheets, hidden cols\)](#) [XML](#)

- The dashboard view grid shows you all the front page articles. The two newest articles are automatically shown on the Home/Index page.
-

- The article view allows anyone to search and read all submitted articles.

Keyword:

Submit

Title: Programming Comments

Date: 2014-11-20

Author: Ryan Hewitt

Title: Database Indexes

Date: 2014-11-18

Author: Ryan Hewitt

Title: Java 9 Features Revealed

Date: 2014-11-17

Author: Peter Bogun

Title: Salted Password Hashing

Date: 2014-11-13

Author: Dan Thom

Title: PubSub Design Pattern

Date: 2014-11-12

Author: Peter Bogun

Title: My Experience With Django

Date: 2014-11-10

Author: Dan Thom

- **After searching for articles you are able to click on them to take you to a dynamically generated page for the selected article to read it.**

Programming Comments

Written By: Ryan Hewitt

Code comments that I find helpful

I'm a huge believer in code quality. When I write code, I put in a lot of effort to make it be as easy to read and understand as possible. Because code is read much more often than it's written, by writing your code in a way that's easy to understand you're saving lots of time for the developers who are going to read your code in the future (one of them being future you) and you're making it easy for them to build on your code and extend it.

As we all know, a big part of writing good code is comments that explain crucial points about the code. This is what this blog post is about.

Making good code comments is not a trivial thing. We've all seen comments like this:

- The author page is very similar to the article page. Search for authors in the search bar:

Keyword:

[Hewitt, Ryan](#)

[Thom, Dan](#)

- The search bar only searches for shoebox "Authors". There are many shoebox users in the db but only 3 authors at the moment (Peter Bogun, Ryan Hewitt, Dan Thom).
- You are able to click on an author to get more information about them and see all of their submitted articles.

Author: Ryan Hewitt

Published Articles

[Title: Database Indexes](#)

[Date: 2014-11-18](#)

[Title: Programming Comments](#)

[Date: 2014-11-20](#)

- You are able to click on their articles in this view to again read their articles.

Database Indexes

Written By: Ryan Hewitt

Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed.

-
- All controller actions take place in controllers/default

Name	Date modified	Type	Size
default	11/22/2014 12:35 ...	Python File	4 KB
appadmin	9/15/2014 6:36 PM	Python File	24 KB

```
Python 2.7.8: default.py - E:\web2py\applications\shoebox\controllers\default.py
File Edit Format Run Options Windows Help
# -*- coding: utf-8 -*-

def index():
    """ Displays most recent featured post on the index page """
    query = db(db.featuredPost).select(orderby=~db.featuredPost.postDate, limitb
    return dict(query = query)

def articles():
    """ Search all submitted articles """
    form = SQLFORM.factory(Field('keyword', requires=IS_NOT_EMPTY())).process()
    if form.accepted:
        query = db.articles.title.contains(form.vars.keyword)
        rows = db(query).select(orderby=~db.articles.articleDate)
    elif form.errors:
        response.flash = "Incomplete Search!"
        rows = ''
    else:
        rows = ''
    return dict(form=form, rows=rows)

def article_CLICK():
    """ Loads individual page for each article submitted to ShoeBox """
    article_id = request.args(0, cast=int)
    article = db.articles(article_id) or redirect(URL('articles'))
    return dict(article = article)

def authors():
    """ Search all shoebox authors """
    form = SQLFORM.factory(Field('keyword', requires=IS_NOT_EMPTY())).process()
    if form.accepted:
        query = ((db.auth_user.first_name.contains(form.vars.keyword)) |
                 (db.auth_user.last_name.contains(form.vars.keyword))) &
                 (db.auth_user.is_author==True)
        rows = db(query).select(orderby=db.auth_user.last_name)
    elif form.errors:
```

- All of the tables are in models\db_content.py

	Name	Date modified	Type	Size
	 db	11/19/2014 5:18 PM	Python File	4 KB
ads	 db_content	11/22/2014 12:56 ...	Python File	2 KB
places	 menu	11/21/2014 5:32 PM	Python File	2 KB

```
Python 2.7.8: db_content.py - E:\web2py\applications\shoebox\models\db_conte...
File Edit Format Run Options Windows Help

db.define_table(
    'articles',
    Field('title', requires = IS_NOT_EMPTY()),
    Field('articleContent', 'text', requires = IS_NOT_EMPTY()),
    Field('articleOwner', 'reference auth_user'),
    Field('articleDate', 'date', requires = IS_NOT_EMPTY()),
    format = '%(title)s: %(articleContent)s: %(owner_id)s')

db.define_table(
    'featuredPost',
    Field('postTitle', requires = IS_NOT_EMPTY(), label="Title"),
    Field('postContent', 'text', requires = IS_NOT_EMPTY(), label="Content"),
    Field('postAuthor', 'reference auth_user'),
    Field('postDate', 'date', requires = IS_NOT_EMPTY()),
    format = '%(postTitle)s: %(postContent)s: %(articleOwner)s')

#####

""" Populates tables with sample data """

if db(db.auth_user).isempty():
    import datetime
    from gluon.contrib.populate import populate
    pb_id = db.auth_user.insert(first_name = "Peter", last_name = 'Bogun',
                                email = 'peter.bogun@example.com',
                                password = CRYPT('test')[0])

    populate(db.auth_user, 50)
    db(db.auth_user.id <= 25).update(is_author = True)
```

- All of the views for each page are in views/default/<view>.html

