April 28, 2015

A01952442

Adam Tew

# Django Kittens

### Abstract

This report is on the implementation of a Django website. It describes some functionality of a Django project dealing with eCommerce. It includes sessions, authentication, the use of a database, and the nature of MVC.

### Introduction and Motivation

My name is Adam Tew. I'm a student at Utah State University in the department of Computer Science. I have been developing websites professionally, personally, and academically. I have used ColdFusion, ASP.net, and Django. CF and ASP have strong backend capabilities; however, there are certain things that Django can do with ease that I have not been able to implement in the other two.

Django is a powerful framework built on a versatile language with an ever-growing market full of clients and developers. It would be in anyone's best interest to take a closer look at all it has to offer.

### Background

This report is a conclusion to the final proposal submitted earlier, where, in the description, it was stated that

"This application will be an eCommerce website that sells kittens using Django. There will be an SQL database that dynamically feeds data to each page. Although there will also be a lot of HTML, CSS, and JavaScript to help with the front-end functionality, Django will handle all of the backend and dynamics. There will also be session scope components that have to be taken care of, namely login, keeping track of items selected to purchase and checking out."

The architecture of this particular implementation is a Python MVC (Model View Controller) framework called Django. It is a library built on the python programming language for web development.

* Model: Models are Python objects relationally mapped to a table within a database.
* View: Views are actually implemented via files called templates. Within the template one formats all of the data he or she wants presented to a user.
* Controller: Controllers are implemented through a file named views.py. According to the Django developers "a view normally delegates to a template, which describes how the data is presented."
* Another important item to note is the urls.py file. Retained within this file are the "url confs." These are bits of regular expression that help resolve the URL typed into an address bar to a particular view, which then presents a given template.

### Detailed problem description

The fundamental issue to be resolved by this endeavor is that of presenting coherent information about products to the public at large. The vehicle of implementation in our example is that of feline in various varieties. Though the content on this website may not pertain to many industries, the framework could be transferred to most anyone with an interest in selling items online.

By harnessing Django's libraries utilized in this application, we can solve the problem of dynamic content presented on each page, preserve user purchasing decisions throughout their time at the website, and use authentication so that many different individuals may browse and checkout simultaneously.

### Describe the architecture of your solution

The architecture of this solution is patterned after a traditional eCommerce website. The framework utilizes libraries within Django to fetch information from a database and dynamically present it on a series of pages. These pages include index, about, home, contact us, item categories, item details, login, cart, and checkout. Of these pages, the index, about, home, and contact are static; categories, details, login, cart, and checkout are all dynamic.

### Describe the implementation of that architecture.

This implementation utilized sessions, authentication and user management, and models using the following Django libraries:

1. django.contrib.sessions.middleware.SessionMiddleware
2. django.middleware.csrf.CsrfViewMiddleware
3. django.contrib.auth.middleware.AuthenticationMiddleware
4. django.contrib.auth.middleware.SessionAuthenticationMiddleware

The page breakdown is as indicated:

* The home page presents some of the featured cats.
* The index page resolves to the home page.
* The about page describes what the website is about -- selling cats.
* The contact page shows contact information.
* Items may be selected for purchase on the categories or details pages.
* Items selected may be viewed on the cart and checkout pages.

### A thoughtful conclusion

Recently, I have been developing websites professionally, personally, and academically. I have used ColdFusion, ASP.net, and Django. CF and ASP have strong backend capabilities; however, there are certain things that Django can do with ease that I have not been able to implement in the other two. The most prominent example is the pure MVC nature of Django. Many times have I seen spaghetti code full of logic and presentation code, however, this is not allowed in Django, which means those lines will never accidentally be mixed.

As stated earlier, Django is a powerful framework built on a versatile language with an ever-growing market full of clients and developers. It would be in anyone's best interest to take a closer look at all it has to offer.

### References

Tew, Adam Final Proposal, March 18 2015

<https://docs.djangoproject.com/en/dev/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-view-the-template-how-come-you-don-t-use-the-standard-names>