Strategy to Make Money Web Based Businesses

Fenton Travers

Sept 20, 2010

Contents

List of Figures

1 Overview

1.1 Overview

This document lays out the strategy for developing a successful online tailor clothing business. This business will take advantage of the cheap costs of tailor labour and fabric in Shenzen, combined with the marketing potential of the internet, to create a successful business model.

1.2 Background

In Shenzen, the cost to make a Jacket and two pairs of pants totals 180 ($HK). This is including labour and material. This translates to approximately 23 ($CAD). To have a custom fitted suit made can cost approximately 1000 ($CAD), leaving 4300% markup margin potential. Of course there will be costs such as shipping that may double the overall cost, however there is still plenty of margin to make this a viable business.

1.3 Keys for Success

To build a successful online business, the skill set required is technical, and for the success of this business, it will be important that the key business owners possess these skills. These skills will be outlined in the following section.

1.4 Web Business Skills

To have a successful web business you need to know how to build a web application. A web application typically has some interactive functionality that lets you order products. The typical structure for this is an architecture like the following image:

1.5 Application Server

The application server is where all the code for your website lives. This is where 90% of your development effort will be spent. You will write your web application in some *Language* such as Java, PHP or Ruby, to name a few. This application will store data, such as new orders, product catalog, etc..., in the Database.

You will want to learn the language you will use for your website. You will also need to learn/understand web technologies such as: Cascading Style Sheets (CSS), Javascript, and HTML.

You will also need to learn to use the tools to write, manage and test your code. These tools are an Integrated Development Environment (IDE), Version Control System (VCS) and bug tracker.

1.6 Database

The database contains your data. Typically, we'll use either an Oracle database or MySQL. The skills required for this are learning some basic SQL commands.

1.7 Getting Started

1.7.1 HTML

The first thing you need to do to become a Web Business entrepreneur is learn some basic HTML. For this you will need a text editor that can write plain ascii files...i.e. NOT MS WORD. Notepad will do, but it sucks ass. Just find a simple text editor initially, but also consider mastering a more powerful editor. My personal recommendation would be either Eclipse or Emacs. Eclipse might be easier to learn because it is very visual. If you want to get very hard core like me...go for Emacs. Regardless, your first steps will be to learn some basic HTML. I'm not going to reproduce HTML tutorials here, as they abound on the web, so go to google and learn some. However, I'll say some topics that you should understand.

- Try to understand how to create a table in HTML.
- Know how to include an image, a hyperlink, even a video.
- Learn how to do basic formatting, like font size, bold, etc...

That should get you started for HTML. Now lets move onto CSS.

1.7.2 CSS

You need to learn the basics of CSS. Here are the bullet point topics to understand:

- Learn how to include an external CSS file in your HTML file you created above.
- Learn how to control font size and color in your CSS.

Again, I won't go into how to do that here, as tutorials abound on the web.

1.7.3 Javascript

Once you start learning Javascript, you are getting quite advanced. Here is a short list of items/objectives you can aim to become able to do.

- Check that the value of a text input box is a number
- Mousing over an image updates the value of some text somewhere in the DOM (Document Object Model)

I recommend you use firebug, a firefox plugin, that allows you to debug javascript.

1.8 Conclusion

If you can learn the above items, you have progressed quite far in your education of Web technologies. All of the above is what we call 'client-side' technologies. There are 'server-side' technologies. Common types of these are PHP, Perl-CGI, Ruby on Rails, JSP's, ASP's, etc...

1.9 Server Side Programming

When building a web site, significant portions of the site will be 'server-side' code. This is because, normally, you have a database on the server side where you store your orders, catalogue information, etc. So the normal flow is like this:

1. A customer goes to your website, putting a URL into their browser.
2. The request goes from their web browser, over the internet to your web server.
3. The web server then examines the URL that was requested and passes that request to your back-end system for processing
4. The backend system then does any processing that is required, including reading from and too the database
5. Finally the backend system constructs an HTML page to send back to the customers browser.

The above process repeats for every request from the customer. So to be able to do the above there are several skills that you need to acquire, and they are listed below.

1.9.1 A backend (server-side) programming language

One of the first things you need to do is to determine which back-end language you will use for constructing your web-site. You have options to choose from. Some examples are PHP, Ruby on Rails, JSP's/Servlets, etc. The language that I will be writing about is the latest Java technologies. The drawback with using Java is that it is slightly more complex to setup initially. The advantage is that it is easier to debug Java programs than other types of languages. This starts to pay off as your site gets bigger and bigger.

1.9.2 Environment Setup

Since setup is sometimes a pain with Java, you want to find an option that is as simple as possible. For this I recommend using Eclipse or NetBeans. As a first option, I'll probably suggest using NetBeans, as it is the most up

to date (today) for the latests and greatest Java web technologies. You'll need to download the latest version of the Java SDK, and the latest version of NetBeans for your platform. Go ahead and get those two things installed and we'll move on to the next steps.

1.9.3 Next steps

The first thing you'll want to do is some basic templating, and server side functionality. Basically what we are doing is moving from 'static' html pages to 'dynamic' html pages. That means the content of the web page gets constructed on the fly.

One simple example of this would be to create a web page that displays the current time. Everytime this web page is viewed, it would display a different (current) time. So you see this is very different than a person writing an HTML page by hand and putting the time inside that page...as you can see, that is practically impossible to do. However, with a computer, this can be achieved trivially. That is the power of computers!

So your objective here should be to create a single page website that just dynamically displays the current time to the user.