

The Secret To Earning Earn \$100,000+ per Year While Working Only Two Hours Per Day

By: James Sides

Important Note:

**You do NOT have permission to share this PDF but
you are welcome to apply to promote it as an affiliate.
I pay 100% commission on this ebook.**

LIMITATION OF LIABILITY/DISCLAIMER OF WARRANTY

The author of this ebook has done his best to ensure that the information contained herein is valid, truthful information. However, he makes no claim or warranty, expressed or implied, as to the accuracy, applicability, fitness, or completeness of said information.

By reading this ebook, you acknowledge that you will not hold the author liable for any loss or other damages that may occur as a result of any of the information in this ebook.

This document is protected under international copyright laws, and any attempt to illegally reprint or distribute it is prohibited and will be prosecuted to the fullest extent of the law.

EARNINGS DISCLAIMER

The earnings that you actually experience will depend on the amount of labor that you put into your efforts as well as your ability, experience, education, market trends, search engine algorithms, and many more personal and external factors. We do not guarantee or otherwise promise that you will earn any particular amount of money.

Who Am I?



My Kiddo and Me Enjoying the St Louis Zoo (Something living the internet lifestyle allows me to do any time the mood strikes)

So just a quick heads up, this is not a typical ebook that you'd purchase from me. There's no fancy formatting, no structured chapters, no real flow, etc.

This is simply a down and dirty off the cuff view into one of the ways I earn money online.

But before I dive in let me give you the brief 30,000 feet overview of who I am.

I started online marketing in 2005 and SUCKED at it big time.

I lost my butt month after month year after year until I finally had a breakthrough.

In 2010 I had my first five figure year.

Then, In 2011, I had the real epiphany of my career - I decided to stop spending all my energy promoting other people's products and started building my own.

In less than six months I broke six figures net and from that point forward never looked back.

Since then I've had a hand in millions of dollars in transactions online, yes millions.

Am I a millionaire?

Not yet I'm not, but I've managed to do pretty well for myself and realistically there's no reason you can't do the same.

Here's the inside scoop though - as fancy and impressive as all the above may sound, I made SERIOUS mistakes getting from point A to B.

I nearly destroyed my marriage chasing this "impossible" dream.

I missed out on time with my daughter that I'll never get back.

And I sacrificed my health all to make this happen.

But NONE of it was necessary.

(These days my marriage is amazing, my daughter and I spend loads of quality time together, and I hired a personal trainer who keeps my butt active at the gym six days a week)

The majority of the time I invested into my business was a total waste that achieved no measurable results.

And that's why in this pdf I'm going to show you how to avoid becoming another marketer who sits at his/her desk 60 hours week while only getting 5 or 10 hours of real work done.

I don't want you to go through the hell I went through so please, pay special attention!

I promise you I'll lay out exactly how anyone can earn six figures per year while only working 2 hours per day.

(And yes, you get to take weekends off!)

In other words I want to teach you how to build a "Lifestyle business" aka a business that allows you plenty of time to enjoy yourself while still covering your financial needs.

Let's Dive In

So what business model am I going to show you today?

Well I'm glad you asked...

The model I'm referring to is product development but more specifically software products.

By the way, I don't actually do any programming myself. I couldn't program my way out of a wet paper bag.

Don't worry if it sounds scary or expensive, its not near as bad as you have been led to believe.

(There's a reason most successful software vendors try and spread the myth that it cost tens of thousands to start etc - the less people making software the less competition they have to deal with)

Software isn't the only way I earn online but its by far my favorite because of the extremely low time requirement and massive ROI it provides on my money invested.

But...

Before I begin let me address the big pink elephant in the room.

If this model works so well and I'm making money with it why would I tell other people about it...and if I was going to tell other people why in the heck wouldn't I charge a lot more than I am?!?

And the answer is...I dunno.

Yea, I know I know... I should have some sort of brilliant plan to monetize this info but the reality is I was sitting in the waiting room of a Dr office bored out of my mind and randomly decided that I wanted to write this pdf if enough people were interested so I threw up a post on my Facebook wall stating I'd create this product if at least 50 people hit "like."

And because 50+ people liked that post I have my proof of interest.

So I figure its like this; I'll write this ebook and hook you up for next to nothing.

Then when you take action with it and have some success I'd like to think that you'll reach out and offer a testimonial for me that I can use for my high ticket mentoring program.

Yes, one of the ways I earn money online is through high ticket (\$500-\$10,000) programs.

Want to guess where the leads for those programs come from?

DING DING DING, my software products!

Or heck, maybe you'll decide by the end of the report that I actually know a little about what I'm talking about and want to hire me yourself - that would count as monetization right?

I have to tell you - for a long time I have wanted to share this information with people but I really felt like it would be a waste of time because the minute people heard the words "software" they'd run for the hills.

I hope like heck that my fears are unfounded and you prove me wrong by taking this info and changing your life.

Over the last three to four years I've developed and sold dozens of different software products.

The least expensive of these was a tool called TubeSpy that cost me \$650 to build (I also gave my developer a \$200 bonus on top of that).

**I've actually developed software that cost less than that but I used them as lead bait or gave them away as bonuses so technically I did not "sell" them.

The most expensive was an affiliate network called ProductPay.com (I sold my ½ of the company in December 2012).

And even though over \$4.5 million in sales have happened on ProductPay, it was actually TubeSpy that changed the course of my career - and my life.

You see, when I launched TubeSpy in the summer of 2011, my biggest month ever online was just shy of \$8,000 and on average I was making less than ½ of that... yet within 72 hours of making the software available for sale I had over \$52,000 in sales.

The #'s That Changed My Life

Product Sales Stats

TUBESPY 					
Views	Visitors	Sales	Refunds	Sales Revenue	Paid to Affiliates
14147	10379	2484	58	\$53080.02	\$26048.84

To put that in perspective, in 2005 when I started online marketing I had just been fired from my job where I was earning \$31,500 PER YEAR as a district manager for a local payday advance company.

And, TubeSpy was my FIRST ever software product. I didn't have a clue what I was doing - I was simply "winging it."

Looking back, knowing what I know now, I could have very easily made 3-4x more money on that product but hindsight is always 20/20.

(For example, one mistake I made was only making it available for sale for one week before closing the offer, forever.)

The bottom line is having my first ever \$10,000+ day really changed my entire mindset.

I knew right then that I wanted to develop software for a living and I've been doing it ever since.

Hopefully you're starting to see the potential here because now I'm going to show you how to actually do it yourself.

Step One - Market Research

In nearly every marketing course I've ever taken I was told that I needed to find a problem and then provide a solution to that problem.

That method works, and works well but I have to tell you I rarely go seeking problems when it comes to software.

Instead, I simply look at what is already selling and model it.

Notice, I said model, not copy.

Copying a product means you basically copy it exactly as it is, sorta like what Google did (in my opinion) with Google Docs vs Microsoft Excel.

Copying a product is, in my opinion, very unethical and leads to a very bad reputation in most cases.

In fact, just a few weeks back I saw a very large vendor venting on Facebook about how someone had copied his software almost to the T.

I can guarantee you that person who copied it will never get a promotion from this person!

Google gets away with it because they're Google but I wouldn't recommend trying it as a "small time" developer.

Instead what I like to do is find a product that is selling well and look for ways that I can improve it.

Perhaps its confusing to use and I can simply provide a more user friendly UI (user interface).

Maybe its missing some options that would make it a lot better tool.

Maybe it has too many unnecessary options and I can improve it by “dumbing” it down.

The list goes on and on but the reality is, you need to not only improve a product but also change the look/feel.

If you want to learn more about modeling I recommend you check out a book called “Steal like an artist” which is a New York Times Best Seller.

I’m not affiliated with the book in any way - just read it a few times in the last year or so and really liked it.

You can grab it on Amazon for around eight bucks.

Step Two - Wireframing

This is the part that most people seriously screw up.

Every time someone comes to me and says..James I hired a developer but the software he made isn’t anything like what I actually wanted I ask to see their wire frame.

And guess what...

Every time they respond with..what’s a wireframe..or I didn’t do one.

So what the heck is a wireframe?

Well, I like to think of it as the “blueprint” for my software.

Could you imagine the result if I hired a contractor to build a house for me but rather than building off a blueprint they just built on the fly based off of suggestions I gave them verbally?

The end result would be a disaster and that’s exactly what happens with most people who try to develop software for the first time.

You absolutely have to look at your developer as a “robot.”

That robot is going to do exactly what your wireframe tells him to do - his job is not to design the software - his job is to program it to your specifications.

There are several ways to create a wireframe.

Here are some ways I like to do my wireframes:

Draw it on a piece of paper

- I can then snap a picture of it with my phone and send it to my developer

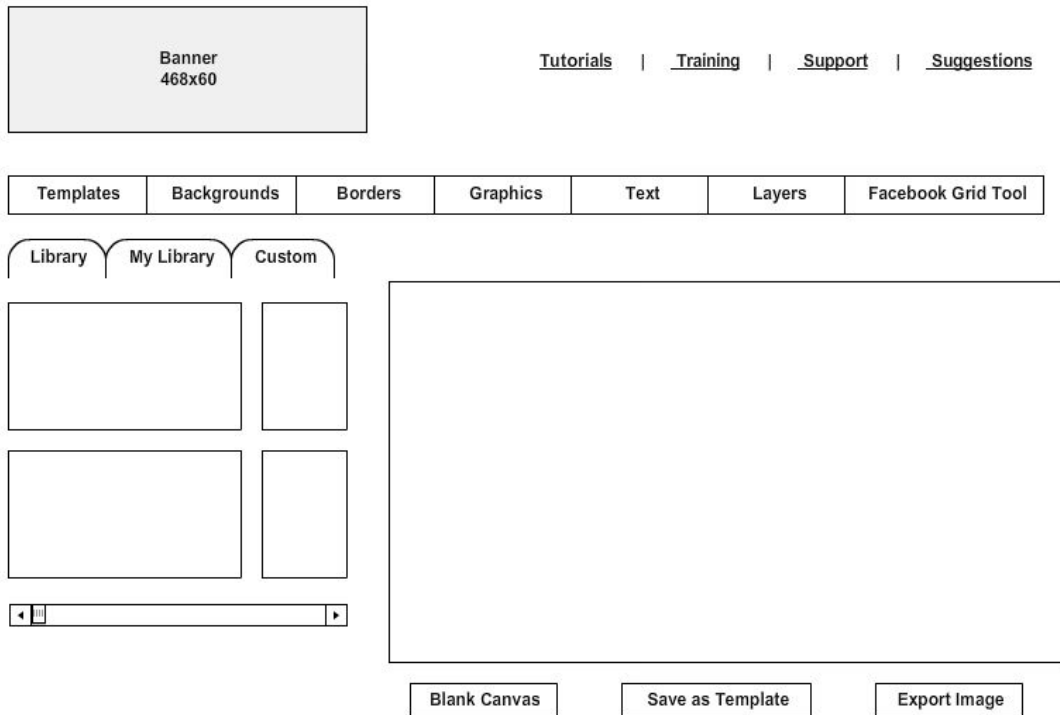
Use Powerpoint or Keynote

- I used to do this when I started out but most often I go with option 3 now

Use wireframing software

- There are loads of affordable options online for wireframe tools, many of the free versions will serve your purpose just fine

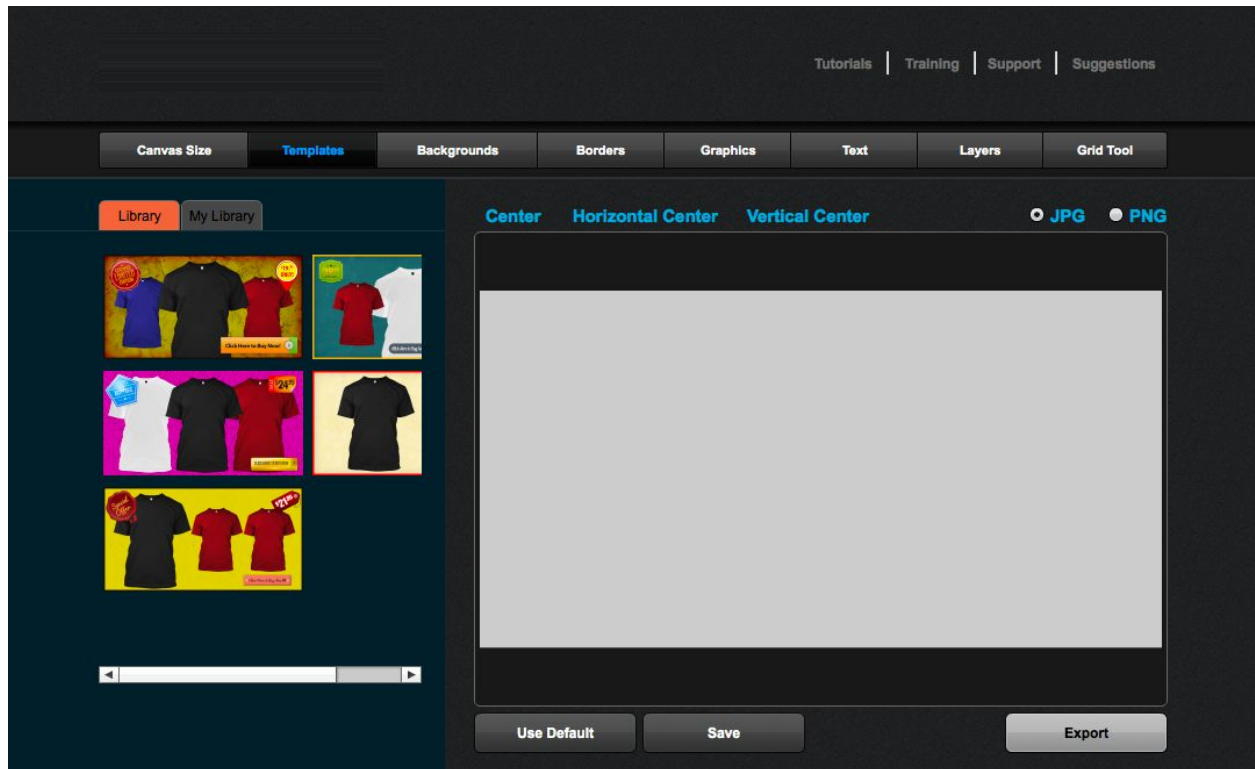
How about I show you an example of an actual wireframe for one of my software products?



Pretty cool right?

I used a free software to make this image and then handed it to my developer. He knew exactly what buttons should be there, what they should do, the lay out, etc.

And guess what, here's a screenshot of the actual tool when it was finished.



Do you see how amazingly close to my wireframe that is?

Its all because I gave him a blueprint he could follow to build me the exact tool I was imagining in my mind.

If you do a great job with your wireframe the chances of you getting what you want are dramatically improved.

Oh, and it also reduces the cost of development because it saves your developer time!

Step 3 - Hiring

In all seriousness, I could write an entire book on outsourcing/hiring without breaking a sweat. In fact, I've taken courses that cost as much as \$2,000 that cover nothing but this one topic so there's no way I can get as detailed here as I would like to but I'm going to give you some basic tips that will get

you rolling. (and trust me, these tips were more than I had when I was starting out)

Unless you're a programmer yourself then you're going to need to hire people to build your software for you.

Either that or you're going to have to learn how to program.

And even if you do know how to program, I never recommend programming your own products unless of course money is super tight.

Its extremely time consuming and time is the most valuable resource we have. Its finite and can never be replaced.

So how then do you go about finding qualified programmers to build your software for you?

Easy peasy.

I use Odesk.com and Elance.com (Both of these are in the process of shutting down for good, they've merged into a new site called Upwork.com) for all my hiring nowadays.

I've also used Freelancer.com, Rentacoder.com (no longer around), OnlineJobs.ph, manilla craigslist, etc.

There are loads of freelance sites out there and while they all have pros and cons, they're pretty much the same at the most basic level.

Simple toss up a job with a description of what you want build and then sit back and relax while applications come in.

Be sure to keep your job description somewhat brief as you don't want to give away your idea to any competitors that also use the same freelance sites.

Once you have 10-20 applicants you'll want to start the interview process.

This is when I give specific details on the job on an individual basis and assess their skills.

Assessing skills is a bit of an acquired talent but there are some things you can watch for even when first starting out.

#1 Ignore their previous work.

I'd say that probably every outsourcing course I've ever bought told me to base my decision off a applicant's previous work.

And for the life of me I can't figure out why!

Its just flat out terrible advice.

The fact is they can easily fake a resume.

So rather than trusting a bunch of links they give you and screenshots you have to go with your gut.

#2 Pay close attention to response times.

The last thing you want is to hire a programmer that takes hours or days to respond when you ask them a question or need to talk with them.

If I add a potential candidate to skype and in the middle of our conversation he/she seems to vanish for several minutes at a time I simply move on.

#3 Check their profile to see how many open jobs they're working on.

One of the biggest mistakes I made when starting out was not doing this..and it cost me big. One specific project that nearly cost me over \$3,000 blew up in my face when the programmer quoted 6 weeks for the job but more than six months later still had not completed it..and in fact he never completed it. I was fortunate enough to only lose my initial payment of \$300 (10%).

What I found out after the fact was this developer was working on more than a dozen projects at once for a dozen different clients, there was no way he could handle it all.

You really have to keep a close eye on this because a lot of times what you'll run into is a "middle man" who is not a programmer at all but he's out there bidding on jobs and then hiring developers on his own to complete them.

Then if a programmer doesn't come through for him he can't deliver on the job and you're the one that suffers.

#4 Test them

Now, this is a bit of an advanced strategy but I thought I'd throw it in here for good measure.

If you have any friends you know who are programmers who you know 100% you can trust, have them whip up a tiny program of some sort - something that would takes them only two to three hours.

Then in your interview process, give the applicants the details of the test job and ask how long it would take them to do this job.

Any that answer 2-3 hours hire and tell them to build it and give them a time limit of three hours.

Anyone that answers much higher, end the interview process - they're either trying to overcharge you or they're not good enough.

Now, anyone that actually gets the software done in the 3 hour window is a legit programmer and you know it.

The downside to this method is you'll need to pay them for the time they work on the software but it is very very effective for weeding out the "wannabes."

#5 Keep your cards close to your chest

Don't ever let a developer know you're clueless when it comes to programming.

Consider yourself the owner of your company, and make them consider you one as well. They don't need to know you're a solo person operation.

If you let them know you're clueless you're very likely to regret it.

When I first started doing mobile games I once paid a guy \$300 to change my ad IDs in my code. I had no idea what the job was worth and it sounded decent enough.

Later I learned how to change the id's myself and discovered it was literally a FIVE MINUTE job!!!

So don't think for one minute that a programmer won't try and take you for every dime you have if you let them know you know nothing about programming.

Step 4 - Development

The development stage is actually one of the least time consuming parts for the entire process for you because the reality is you're not doing much.

The key to being in business is outsourcing the stuff you're not good at or the tasks that are not worth your time - most people already know that.

The thing is, most people never get there because they're afraid to let go of control.

One of the most beautiful parts of a software business is you HAVE to let go - unless you are a programmer yourself of course.

With all that being said, there is a sure fire way to almost guarantee poor results with your software.

And that is to ignore the development process altogether.

If you hand your wireframe to a programmer and then fail to speak to them until the project is done its a risky move on your part.

You should be getting daily updates of where the project is at AND MOST IMPORTANTLY daily builds of the source code.

I can't tell you how many people I know who don't do this and its unbelievably scary to me.

If you're not getting regular updates of the source code and things go south with your developer you're basically just out of luck. You have nothing to show for your money.

If on the flip side you're always getting daily updates of the source code this does two things.

- 1) Guarantees you have at least something for your money if your developer goes MIA on you
- 2) Keeps a fire under your developer's butt because they know they have to turn in that new code every day

Step 5 - Bug Testing

As you reach the end of the development process you're going to take a more active role in testing the software.

If you simply trust the developer that everything works properly then you're likely to get a rude awakening when you start selling your product.

I can't really tell you what it is about freelance developers that makes them a bit lazy in this area but in general I find that every product I develop has bugs when the developer first hands it to me and says its ready.

Trust me when I tell you, do not ever take your developer's word for it when it comes to your software being market ready.

Instead its a good idea for you to spend a few hours yourself trying every button, every option, etc that the software has - making sure that it all works as its supposed to.

Then I like to give it to a handful of customers to beta test for me.

If you don't have any customers to beta test don't sweat it, just pick a few friends that you trust and have them check it out for you.

****Hint:** If you're struggling to find people to beta test I've found that Facebook is an easy way to get people in the door.

Step 6 - Software Launch

Your launch is where the rubber really meets the road.

This is the funnest part by far because its where you get to see money actually coming into your account.

But in all fairness I have to warn you, launches are a very complex subject - so complex in fact that you'll have no trouble finding \$2,000-\$3,000 courses that teach nothing but launch strategies.

I am not telling you that to freak you out - I'm just letting you know that what I show you in this ebook is the tip of the iceberg.

It will be more than enough to get you going - and in fact in the right hands is more than enough to earn low to mid six figures per year.

The other good news is, the strategy I'm about to teach you requires very little time, is low stress, and can result in some very strong Jv partnerships in the future.

There is no technical term for the strategy and I've never seen it taught anywhere but I'm sure there are other folks that have figured it out; of that I have no doubt.

I simply refer to it as a private launch.

Unlike a public launch a private launch does not require an affiliate contest, it does not require endless recruitment, it doesn't require giving away huge commissions, etc.

Instead private launches consist of me hitting up a few people that I know have laser targeted lists for my product.

I offer them a 50% commission and set up a special landing page just for their list.

Then I allow them to promote the offer for 72 hours - many times with a special discount to help encourage sales.

And when 72 hours is up the product is either no longer available to that list or is no longer available at the discounted rate.

This allows me to add a countdown timer to the offer to help get people off the fence who aren't sure about purchasing.

The reason this method is so powerful is its a true win/win.

The affiliates who promote this way end up making more sales aka more money and I am able to make solid money myself because I don't have to pay out any crazy affiliate prizes, high commission rates, etc.

Now, will I make as many sales as if I had done a public launch?

No, the chances are very slim of that happening but a real launch is a huge amount of work whereas this is very low stress and can be done around a full time job if need be.

Remember, the whole point of this ebook is to show you how to make an extra \$100k year while only working 1-2 hours per day.

You're always welcome to work more and in turn you have a better chance of making more.

Now, you may be saying to yourself...that's easy for James to do because he's got tons of connections and while that may be true for some projects its totally false with others.

For example, when I launched Springly via this method in September of 2014 I had to go out and try to get big tshirt "gurus" to promote for me and many of them I had never met or even spoke to before.

At the end of the day people are in business to make money and if you have a quality product that they feel they can make money on they don't really care if they know you or not!

****Tip:**

Sometimes the fastest way to get your foot in the door is by finding someone who already has their foot in the door and partnering with them. When it comes to my students I am always willing to consider a partnership to help them reach more affiliates/customers with their offer so don't be shy about contacting me when you have a FINISHED PRODUCT.

Please do not contact me with "ideas" as I get way too many people who do that and never actually create a product.

Step 7 - Updates & Support

A lot of folks get it in their head that their job is over once they've sold a product but the reality is the people who make it long term are the ones that understand how important updates and support are for your products.

Even the best products are going to run into problems from time to time.

For example, one of my recent software products called Springly which pulls designs back from the popular web site Teespring.com was working

great but suddenly one day last week I got a flood of tickets at my help desk telling me it was not functioning.

When I researched the problem I found that apparently TeeSpring had updated their own site which meant that I had to update my software to work with the new site layout.

This was a chance for me to either a) let my customers down and lose them for life or b) rise to the occasion and prove to my customers I was one of the “good guys” that would take care of them.

Obviously I chose to update the product and within 3-4 days I was able to send out an update not just fixing the problem but also adding several new features.

By adding these new features I went above and beyond what my customers expected and that really goes a long way in building loyalty.

And make no mistake, a small raving fan base will outperform a huge lukewarm fan base any day.

If you're planning on building more than one product/software then you're going to want your existing customers to buy from you again and again..and to tell their friends about you.

The way to do that is to go beyond the call of duty every chance you get.

Look at it like this, a great update may only cost you a few hundred bucks but the relationship you build with your customers is priceless.

Putting It All Together

As you read through this ebook you may be feeling like I left quite a bit out and the truth is I absolutely did.

I didn't focus on any of the advanced strategies for building a multiple six figure per year business.

I didn't focus on how to recruit affiliates for a huge six figure launch or how to run paid traffic or any of the traffic generation techniques that can take you from six figures to seven.

And I didn't go into large amounts of detail on any one section.

I did that on purpose - in order to give you the end all be all I would have had to charge you a lot of money for this product.

I wanted to be able to give it to you at a price that did not break the bank.

But all that being said, I want to also remind you that the information in this guide is MORE than enough to build a full time income through software development.

How do I know?

Because I would have LOVED to have this when I was starting out.

I had to figure it out all on my own, and even then I was able to make six figures within just six months implementing this model.

Remember, the goal of this guide is to get you to just an extra \$100,000 per year!

You can do this.

Don't let fear stand in your way.

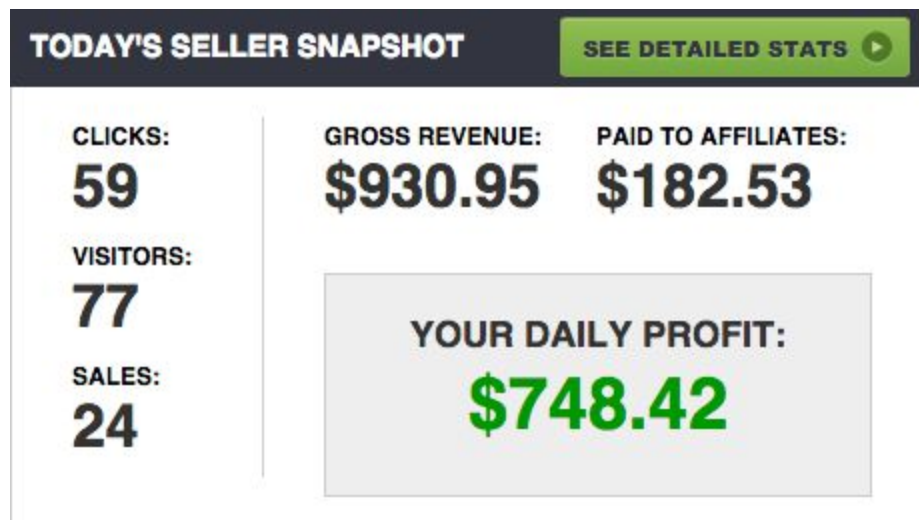
If you're doubting your ability to do this thing long and hard about my story.

It was less than five years ago that I was struggling to make \$100 day while working my butt off.

This week I've been sick, like really feeling bad and laying in bed sick.

I have not sent out a single email to my list, have not ran any ads, and have not asked any affiliates to promote my products.

Yet here are the stats from just my most recent software from just TODAY:



Notice, very little of that came from affiliates.

When you create awesome software people will tell their friends about it FREE.

Do you think you'd be ok taking a sick day and still earning \$748.42 profit?
I know I sure am.

So get out there and make it happen!

Yours for success,

James Sides

Serial Entrepreneur & Software Fanatic

