THE COMPANION TO

THE COMPLETE WEB DEVELOPER COURSE

How to make **$10,000** while learning to code

Second Edition

By Rob Percival

Preface To Second Edition

This book was initially written in the summer of 2014 as an accompaniment to The Complete Web Developer Course, which in turn I hoped would help a small number of people learn web development and start new careers.

Since then, both the course and book have gone far beyond my initial expectations, with the course selling over 200,000 copies, and the book itself becoming one of the highest-rated career advice books on Amazon.

Both the course and the book inevitably have become somewhat outdated, and this second edition arrives with The Complete Web Developer Course 2.0 as a revised and updated version of what came before.

The general career advice sections have remained largely unchanged (with the exception of a few broken links), as what I wrote remains solid advice today. Finding a niche, faking it until you becoming it, and building an online reputation are as powerful techniques in 2016 as they were in 2014.

Other aspects of the book haven't aged quite so well, with the landscape of freelance website changing dramatically in the last couple of years.

This edition serves to revise and update the first edition, with some new material, new web links and an expanded version of My Story, taking the reader up to the present day.

I hope you enjoy it, and, as always, happy coding!

Rob Percival

Cambridge, February 2016

Foreword

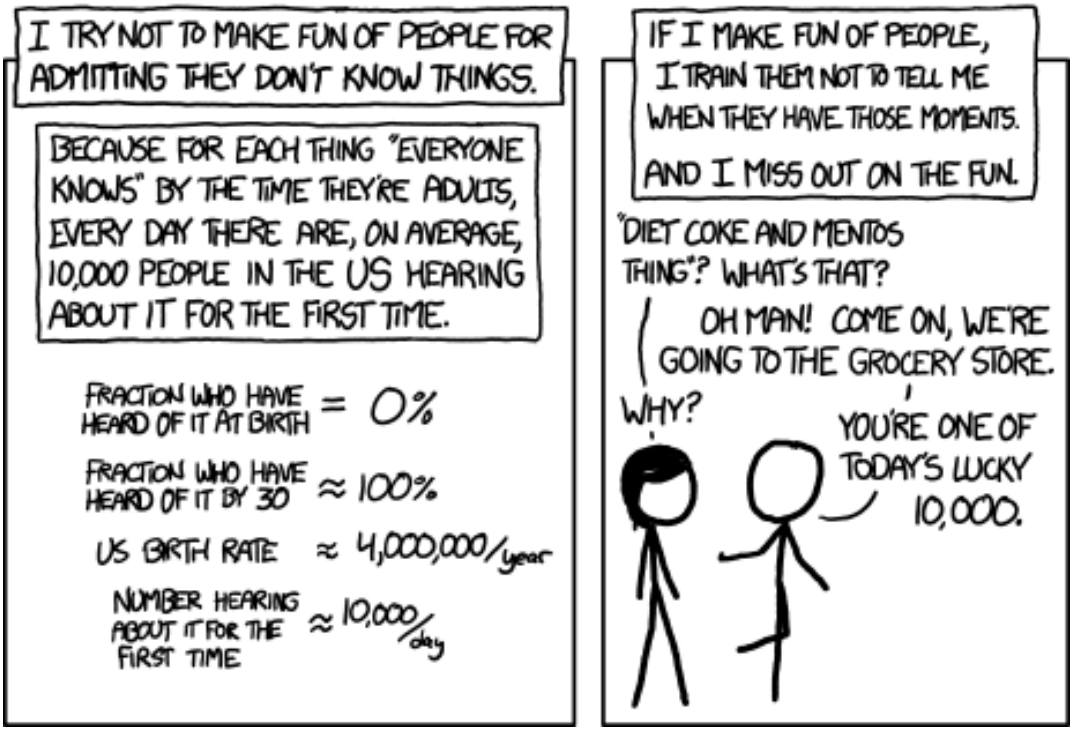
Learning to be a developer was one of the best decisions I've ever made. It's given me the freedom to quit my job, start my own businesses and pursue my hobbies. I can work anywhere in the world, and in the last few year's I've travelled to Morocco, Namibia, China and spent 3 months backpacking through Kenya, all paid for by revenue from my online systems.

The beauty of learning to code is not just that you can learn very cheaply, but that you can start making money almost immediately, if you know the tricks of the trade. This book contains everything I learned during the early years of my web development career, and if you follow the step by step guides you'll be earning money much quicker than I ever did.

This book is designed as a companion to The Complete Web Developer Course (www.completewebdevelopercourse.com), but you can use any method of learning to code you like. The course is the best way I know to learn the basics of web development, and I guarantee that you'll earn the course cost ($199) back within 6 weeks, or I'll give you a full refund. You'll also get free web hosting, and I'll be there to help you out in the forums.

You won't need any pre-existing skills or experience, and you don't need to have a a large following or mailing list (although I'll show you how to make the most of these if you do have them).

The book contains over 20 'challenges', some of which will directly make you money, and others will just move you in the right direction. Each are clearly explained, and I'd recommend trying out all the challenges before you move on to the next section.

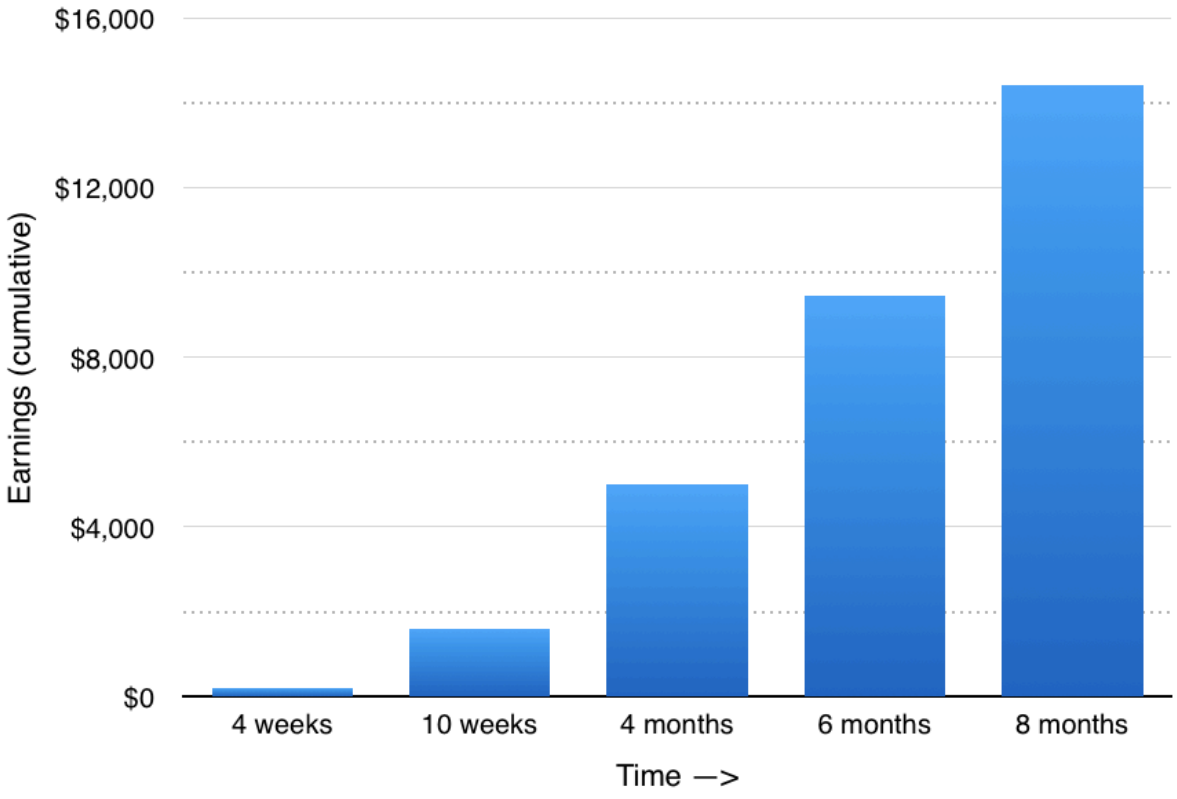


Learning is fun! xkcd.com/1053/

It's worth adding that although following the advice in this book is *simple*, that

doesn't mean it will be *easy.* You will earn money from day 1, but your 'hourly rate' might be devastatingly low. Just remember that what you earn in your first month will be nothing compared to what you will earn in future months, and the growth will continue year-on-year. In your first few months your main goal is to *learn* the principles and practices of being a web developer, and the fact that you can earn a decent amount of cash on the side is fantastic, but secondary.

Here is a graph of your projected earnings:



A quick word about what this book is *not*: I won't teach you to code in this book. For that, I'd obviously recommend The Complete Web Developer Course, but you can learn using any method you like.

So, without further ado let's get started - congratulations on starting this exciting journey, and I look forward to catapulting you into becoming a professional web developer.

— Rob Percival, Cambridge, UK.

Chapter One: Rebrand Yourself

Earnings Summary

Before This Chapter	After This Chapter
TOTAL EARNED: N/A TOTAL TIME SPENT: N/A	TOTAL EARNED: $0 TOTAL TIME SPENT: 1 week

It sounds cheesy, but the first step toward becoming a web developer is believing that you *are* a web developer. Like with any new skill or profession, you're likely to feel something of a fraud until you've been doing it for several years. There's a great TED talk by Amy Cuddy entitled 'fake it till you become it' (www.tinyurl.com/fakeitbecomeit) which describes this feeling perfectly.

I was a teacher before I became a developer, and it took several years before I could confidently tell someone that I could develop websites. This was well after I had

been paid to make several sites, and had a number of income-generating sites of my own.

I'm not saying you should lie about your capabilities, and I definitely wouldn't recommend taking on jobs that you don't have the skills or time for, but it's absolutely critical that from day one you start believing that you are a coder, programmer, software developer, or whatever you want to call yourself. Put the work in, and it really won't be long before your bank statements start to back up your belief.

I'd recommend focussing on the activities in this chapter whilst completing the first few sections of The Complete Web Developer Course. Aim to spend around a week doing this, and then you'll have the beginnings of an online rep as well as basic web development skills. Not a bad start!

Challenge: Change Careers In A Day

The great thing about the challenges in this chapter (unlike all the others in this book) is that you don't need to know how to code at all to complete them. For this challenge I want you meet someone new. Talk to the person next to you on the train, or in a bar, or at a football match. And when the inevitable question 'what do you do?' is asked, respond 'I am a web developer'. Obviously you'll need to be a little bit careful

that the person you're speaking to isn't best friends with your boss, but the odds of that are fairly low. The goal of this challenge is primarily to make you feel like a web developer, and to start having the sorts of conversations web developers do. You'll almost certainly know much more about the web than the person you're speaking to, and if you get rumbled you can always say that you still have a 'day job' .

Once you've done this a few times, share your experiences on my course forum - I'd love to hear from you!

Choosing A Niche

As I've mentioned, I was a teacher before I became a web developer. This means I've got a lot of insight into how schools work, how teachers think and what makes students tick. I've also got a lot of contacts in local schools and universities. This gives me a big advantage over other web developers when pitching for jobs at educational institutions, and several of my highest-paying gigs have come from these areas.

If you have experience or expertise in a particular area, I'd recommend thinking about how you can brand yourself to match these skills with your web development work. Increasingly, developers are required to do much more than just build a website to a specification. You need to be able to suggest how a website or app will benefit an organisation, how it should work and even train people on how to use the site. If you

have knowledge or experience of working in a particular area, this will be extremely valuable.

Of course, this isn't a necessary precursor to becoming a web developer, but it can be much easier to find work, particularly in the early days, if you focus on a particular niche.

Build An Online Reputation

As a web developer you're going to need an online rep. Some of you may already have 5,000 Twitter followers and a blog with a million hits a month, but for the rest of us, now is a great time to get started.

Very simply, at this stage you need to do two things - buy a domain name for yourself and join Twitter. There are obviously a lot of other things you can do to build your reputation, and we'll examine some of them later on in the book, but this will do for now.

Buying A Domain Name

Your first domain name will be your own online space. Unlike your Twitter account, Facebook or even your wordpress.com blog, you will completely own your content, you can post what you like and will be able to export it at any time. This can be critical down the line - other providers may go bust, start charging high fees or ban

you for some reason, but your domain will always be yours to do with as you like. Whenever someone writes about you, or you build a website for them, ask them to link to your main site. This will likely bring in a trickle of traffic, but will definitely boost your results in Google and the other search engines.

I won't go into choosing a domain name in great detail (there are a few links in the bibliography at the end of this chapter for this). Suffice to say, try to go for a .com if possible, and a domain name that features your name prominently. If you have a particular username that you use widely on the web, using that instead can work well, making that your personal 'brand'. Alternatively, you can build your name into a pun, such as automattic.com by Mat Mullenweg, the founder of Wordpress. Don't spend ages over this decision - the content of your site is much more important than your domain name.

Challenge: Get Your First Website Live

1. Purchase Your Domain Name

Once you've decided on a domain name, get out there and purchase it. You can do this on my site, www.ecowebhosting.co.uk (£6.99+VAT per year for a .com domain) if you like, or any one of many domain name providers.

If you don't buy it with Eco Web Hosting, you can link it to your free web hosting

by putting in a support request to your domain provider asking for the nameservers to be changed to

ns1.ecowebhosting.co.uk

ns2.ecowebhosting.co.uk

This will link the domain to our servers and allow you to run your website and email through us.

2. Create Hosting And Set Up Email

Once you've purchased the domain name, create a hosting package for it at ecowebhosting.co.uk/adddomains. Then, set up an email address in the control panel (there are video guides for this in the first section of the Complete Web Developer Course).

3. Edit The Home Page

Finally, go back to cpanel.ecowebhosting.co.uk, click File Manager to view your website files and then double-click on public_html to view the HTML files on your site. Right click on index.html (the home page for your site), select Edit and add a couple of paragraphs of content.

This content is primarily for Google, so that when they index your site they won't find it empty. The content could be a brief bio, perhaps explaining why you're planning to be a web developer. If you know some HTML, you could add in some links to your Twitter account and anythings else about you online. If not, just put a single paragraph of text about yourself.

Congratulations! Your first site is live! Drop a link in the forum and I'll take a look!

Joining Twitter

Twitter is an increasingly useful place to build a following and connect with people online. It's free and takes no time at all to set up. If you have an unusual name, you can probably use that as your Twitter 'handle', but if not you'll have to get creative. You might want to use the name of your particular niche to immediately alert potential followers to what you're about (my Twitter handle is @techedrob). If you're having problems picking a username, check out some of the links at the end of this chapter.

Once you're online, you need to start building followers. I've linked to a number of guides at the end of the chapter, but here's a few tips to get you started:

1. **Connect with people you know.** Twitter already does a good job of this by importing your email contacts, but search for anyone else you know, follow them

and send them a quick personal tweet (something like '@techedrob Hi! I loved your book!').

2. **Start tweeting.** You'll want to get a few tweets on your timeline before you can expect people who don't know you to follow you. Try to stay 'on message' - tweet about interesting articles you've read about web development, or useful resources that you've found. Aim to tweet at least once a day - buffer.com is an easy way to keep up the flow of tweets when you're busy.
3. **Follow others.** Search for people similar to yourself, ideally from the same geographical area, who post regularly and have less than 500 followers. Follow them and favourite a couple of their tweets. Reply to one of their tweets if you have something useful to add. You should find that over half follow you back.
4. **Join the conversation.** Keep an eye on your Twitter feed and spend 5 minutes a day favouriting and replying to tweets. Be constructive, helpful and positive, offering advice and encouragement.

Challenge: Get 50 Twitter Followers

This challenge is simple. Using the tips above, and the ones linked to at the end of this chapter, try and get 50 Twitter followers in a week. Post about your progress in the course forum (you might even get a few more followers there!).

Other Branding Ideas

Here are a couple of other things you can do to alert people to your newfound skills.

- **Update your LinkedIn profile.** This will depend on your current professional status, but updating (or creating) your LinkedIn profile with links to your website and Twitter feed, as well as a (regularly updated) list of any websites you have worked on or created, can be a useful exercise.
- **Create an email signature.** A simple statement such as “Need a website? I can build one for you” can bring in all sorts of interest.

Yourself. Rebranded.

By now you should have completed the HTML, CSS and Javascript chapters of The Complete Web Developer Course, as well as got your own website up and running, and built a small Twitter following. Not bad for a week’s work!

You should continue to engage in Twitter throughout the course, and keep adding to and improving your own website as your skills increase. These are ongoing tasks that won’t earn you money directly, but will give potential clients and partners something to look at when they search for you, and an opportunity for you to showcase your (increasingly sophisticated!) work.

In the next chapter we'll dive straight into earning some cash through freelance websites.

Further Reading

http://technori.com/2012/05/1693-how-to-make-money-as-a-new-developer/

Great getting started tips

http://www.webdesignerdepot.com/2013/01/how-to-find-your-niche/

Advice on finding your niche

http://business.tutsplus.com/articles/skyrocket-your-freelance-business-by-going-niche--fsw-30751

More niche-related suggestions

http://fourhourworkweek.com/2009/02/27/how-to-buy-domain-names-like-a-pro-10-tips-from-the-founder-of-phonetagcom/

Advice on buying domain names

http://www.dummies.com/how-to/content/how-to-choose-a-good-twitter-username.html

http://socialmarketingwriting.com/6-tips-to-choosing-the-perfect-twitter-name/

How to choose a Twitter username

http://computer.howstuffworks.com/internet/social-networking/information/10-ways-to-get-more-followers-on-twitter.htm

http://thenextweb.com/twitter/2014/01/06/9-ways-grow-twitter-following-ethically/

http://twiends.com/get-twitter-followers

http://socialtriggers.com/twitter-tips/

Tips on building your Twitter following

http://www.amazon.co.uk/HTML-CSS-Design-Build-Sites/dp/1118008189/ref=sr_1_1?ie=UTF8&qid=1398100778&sr=8-1&keywords=web+development

A great HTML and CSS primer

http://www.amazon.co.uk/Dont-Make-Think-Revisited-Usability-ebook/dp/B00HJUBRPG/ref=sr_1_4?ie=UTF8&qid=1398100778&sr=8-4&keywords=web+development

A solid introduction to constructing great websites

Chapter Two: Using Freelance Websites

Earnings Summary

Before This Chapter	After This Chapter
TOTAL EARNED: $0 TOTAL TIME SPENT: 1 week	TOTAL EARNED: $200 TIME SPENT: 4 weeks

Getting web development work in the early stages is tough. That is why I suggest using freelance websites such as freelancer.com, peopleperhour.com, elance.com or one of the others in the bibliography at the end of this chapter. The competition is strong, and it may take a few attempts before you get your first paid gigs, but remember that you have a few crucial advantages over the more experienced developers on those sites:

- **You're primarily there to learn.** Your first job may take you 3 hours and earn you $10, but that's *fine* because you will have learned a great deal about communicating with clients, fixing website code and bidding for a project. Not only that, but you will have earned your first 5 star review (a proud moment, I can tell you!)
- **You can take your time.** Most developers on those sites post generic bids on a large number of projects. You're still learning, so you can take your time and post a thoughtful, relevant bid that shows that you've actually read the details of the post. Believe me, bids like that are few and far between.
- **You can use geography to your advantage.** If you live in the US or Europe, make the most of this by offering to speak to the client on the phone, and using polished English when bidding and replying to messages. Doing this, you'll stand out by a mile.
- **You can go the extra mile.** As you're there to learn, you can do more than what the client asked for without worrying about the extra time spent. If you're setting up Wordpress, install an SEO plugin for them. If you're making a form, use some custom CSS to make it beautiful. Reply quickly and thoroughly to all their questions, and earn their gratitude.

I'll say it again - you will earn money here, but that is your secondary goal. Primarily, you're here to learn how to do freelance web development, and build up your online portfolio and positive reviews.

Pick A Freelance Site, And Stick With It

The hardest part of getting your first gig will be overcoming your lack of positive reviews. For that reason, I'd advise picking one freelance site and sticking with it, at least for now. You can join another later, but once you've got three 5 star reviews on freelancer.com, you'll find it much easier to find work there than you will with an empty profile on elance.com.

I'm not going to go into the pros and cons of each of the freelance sites - I'd simply advise that you check out a few of them and pick whichever site you like the look of. Check that you can receive funds in your country and that you are happy with their payment terms, and sign up - don't waste a lot of time going through all the sites. I've had the most experience with freelancer.com, so I'm going to focus on that site, but the others all work in a similar way.

Here's my list of sites you should check out:

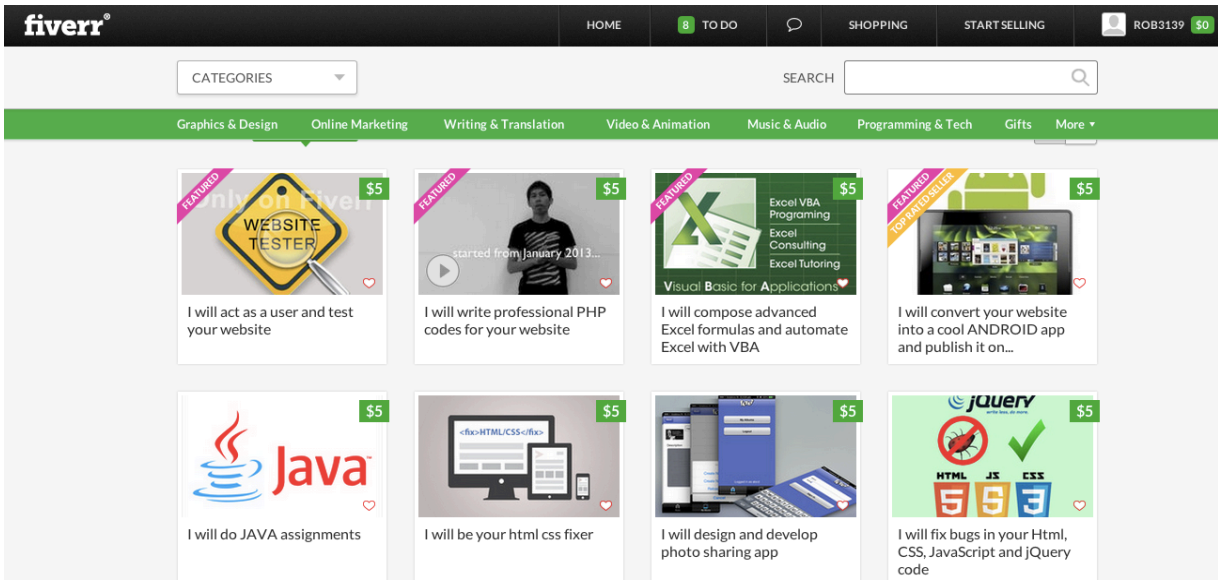
- Elance
- Guru
- Freelance
- Freelancer
- GetACoder

- UpWork
- PeoplePerHour

A really useful comparison of these and other sites is available online at http://www.entrepreneur.com/article/245953, and there are a mighty 71 options for freelance work at https://www.freshbooks.com/blog/freelance-jobs/.

Should You Use Fiverr?

Fiverr.com deserves a particular mention, as it can be an easy way to find a lot of quick and easy jobs. On the whole, I would only recommend using fiverr.com if you are having difficulties getting work on one of the other freelance sites. Unlike those sites, there is little potential for getting larger jobs on Fiverr, which is ultimately what you want. Fiverr can be useful for practicing your skills and earning a little cash (and getting some good reviews to post on your website or Twitter feed), but my advice would be to prioritise the other freelance websites if you can.



Challenge: Create Your Profile

Once you've picked which site you want to work with, you need to sign up and create your profile. I've linked to a couple of guides in the bibliography of this chapter, but here's a few basic tips.

- **Use your real identity.** You'll want all the parts of your online presence to tie together, so use your real name, upload a photo and talk about yourself
- **Be honest.** Don't claim to have skills you don't have. At this stage 'Proficient in HTML, CSS and Javascript' would suffice, and you can then add further skills as needed.
- **Link to your Twitter feed.** If the freelance site allows, put in a link to your Twitter feed - this will add authority to your profile and reassure prospective clients that you are a genuine developer.

- **Complete the exams.** Most freelance sites have ‘exams’ that you can take both in language (English being the most useful) and various languages. They usually cost around $50, but are worth it to get you off the ground when you don’t have any reviews.

When you’re done, link to your profile in the forums, and get feedback from me and other students.

Bidding For Gigs

Initially, look for small, relatively straightforward gigs, with a maximum of $50. Updating websites, fixing broken layouts and adding small features are all common requests. Bid on as many projects as you can, bearing the following in mind:

- **Keep your bid low.** Remember you’re here to learn and build your reputation. Keep your bid low, especially when you have zero reviews. This will get you gigs more quickly and you can increase your price as you go.
- **Explain why your bid is low.** You don’t need to tell the client that you are learning, but you might want to say that you are bidding low in order to get your first reviews on this site. They will see that you have no reviews, and referring to it yourself will show that you understand their concern and have made a low bid as a result.

- **Don't take on big jobs.** You're still learning, so avoid big or technically advanced jobs. Feel free to take on jobs slightly above your current skill level, as long as you're confident you can learn what will be required, but the last thing you want is a bad review and a disgruntled client.
- **Clarify the job.** It's essential that you're clear on what is required, and that it has been objectively stated on the freelance site messaging system. That way, if there is any disagreement, you can refer back to what the job was originally set out to be. Ambiguous language or general aims (such as 'build me a site') are a recipe for disaster.
- **Agree on payment structure.** Even with small projects, it's important to make it clear when payment will be due. I would advise not to start work until a milestone is created (ie. the buyer has made a downpayment, which is held by the freelancer site until the job is finished). That way, if there are any disagreements, it is up to the freelancer site to establish whether the work has been done and release the payment.
- **Be wary of buyers with no reviews.** Buyers have reviews too, and if a buyer has no reviews, be careful. They may well be reliable, but they may not - in this case it is particularly important to make sure the requirements of the job are clear, and that a milestone is paid before you start work.

Challenge: Get Your First Gig

This is the challenge you've been waiting for! Get out there and get your first paid gig as a web developer. Follow the steps above and don't be disheartened if it takes a lot of bids before you're chosen. It will happen eventually, and every bid is a learning opportunity.

Then post your success on the forum so I can congratulate you!

Wash. Rinse. Repeat As Required.

After you've got that all important first gig, you'll find the others come much easier. Keep bidding, keep going the extra mile and earning great reviews and within a month you should have been able to complete 10 jobs, with an average cost of at least $20 per gig. That's $200 (incidentally, the cost of The Complete Web Developer Course!).

While you're waiting for those gigs to come in, you should have been working through the next few chapters of the course, so should now have experience of Wordpress, Bootstrap, and PHP/MySQL. Add these skills to your profile and take gigs using the new skills whenever you can.

Congratulations! You've started earning real money as a web developer. You should also have continued to build your Twitter following to around 200. In the

Wordpress chapter of The Complete Web Developer Course) I show you how to build an awesome portfolio site and blog, so your online presence should now be looking pretty professional. Nice one!

Progress Update

So far, we have total earnings of $200:

Activities	Total Income ($)
Month 1 10 small freelance jobs at $20 each	200
TOTAL	**$200**

Further Reading

http://www.freshbooks.com/blog/2013/01/16/freelance-jobs/

17 freelance websites - a great comparison

https://www.youtube.com/watch?v=0-gMy2IoMfQ

Introductory guide to freelance sites

http://www.freshbooks.com/blog/2013/11/12/6-steps-to-creating-a-freelance-profile-that-wins-business/

http://www.shoutmeloud.com/creprofessional-freelancing-profile-more-leads.html

https://ebyline.zendesk.com/entries/22311088-How-to-create-a-great-Freelancer-Profile

https://www.upwork.com/legal/contractor-guidelines/

Creating a great profile

https://www.workhoppers.com/blog/bidding-on-freelance-work-good-bad-ugly/

Advice on bidding for freelance work

Chapter Three: Building Beautiful Websites With Wordpress

Earnings Summary

Before This Chapter	After This Chapter
TOTAL EARNED: $200 TIME SPENT: 4 weeks	TOTAL EARNED: $1400 TOTAL TIME SPENT: 10 weeks

So far you've learned some basic web development skills, and earned some cash doing small web development work on freelance websites. That's great! But if you want to work on larger (ie. higher paid) jobs, you're going to need to be able to build websites from scratch. This is where this chapter comes in.

At this point, it's worth clarifying the difference between web *development* and web *design*. Generally speaking, web development is putting together a website that *works*, and web design is putting together a website that *looks great*. If you're one of those lucky people that can do both, and has an knack for making great looking sites, then you may not need this chapter.

As developers, however, design is not likely to be our strong suit. We need a way to make our site look good that doesn't require us to put together a beautiful design in Photoshop. Fortunately there is one. Read on, MacDuff…

Premium Wordpress Themes

Most clients don't make the distinction between development and design, and will expect you to be able to make something that both works and looks good. Fortunately, in recent years a number of gorgeous and hugely flexible themes have sprung up, which can be customised as required to make a website which is clean, professional and looks great.

Go on your freelance website of choice and search for 'build a website'. You'll see a number of projects around the $500 mark, and most likely they'll want the following:

- A clean, professional, custom design

- 5-10 pages of content, including an ‘about us’ page and a contact form
- Possibly a blog, portfolio or shop
- The ability to update the website themselves

If you can cater to this market, there will be no shortage of work for you.

Now, take a look at themeforest.net, specifically the Wordpress Themes section. Spend a few minutes clicking around the demos of the Avada and X themes, and any others that take your fancy.

These themes generally look great, are hugely customisable and contain a huge variety of page layouts and designs, allowing you to display your client’s content in an engaging and attractive way. They also have blog and shop designs built in. Finally, they generally come with thorough documentation and active forum-based support.

You need to become an expert on creating sites with these themes, so that you can develop attractive, functional sites fast. Follow through the Wordpress chapter on The Complete Web Developer course for a full guide on how to use these themes.

It is very difficult to get these $500 web design jobs without a portfolio. In fact, I wouldn’t even try. So how do you get your first gig? Here are a few challenges to get you going.

Challenge 1: Your Portfolio

Now is the time to revamp your own website and make it look awesome. There is a step by step guide to doing this at the end of the Wordpress section of The Complete Web Developer Course, but you can also just install a theme and get cracking. Your site needn't (ie. shouldn't) be flashy, but it should be clean, clear and showcase your work.

As always, post a link in the forum to get feedback from me and other students!

Challenge 2: Create A Website For A Friend

Find a friend who needs a website for themselves and offer to make one for them for nothing. Or better, charge them $100 (including the cost of the template), and run it through your freelancer site. This will make them value the site more, and get you a great review. Do an excellent job and add it to your portfolio. Find some more friends and repeat!

Challenge 3: 'Build A Website for $100'

If you run out of friends, go public. You'll need a couple of sites in your portfolio

first, but when you've got them, write a post on your blog offering to design a site for anyone (or perhaps the first 10 people) for $100. This is a great deal, and if the sites you've developed already look good, you'll be inundated with requests. Post your offer on reddit and other forums, and make it clear that you're doing this to build up your portfolio. Run the gigs through your freelancer site, and do an excellent job.

Challenge 4: Design Your Own

If you really can't get anyone to pay you, then spend your time creating great sites anyway. If you have an idea for an app or product, create a landing page marketing the project. Redesign a famous site (paypal.com is a popular one for this!) and post the result on dribble.com. Or build a theme to sell on themeforest.net - simple but effective landing pages are always popular.

Or a final suggestion, find a poorly designed site for a local small business and do an awesome redesign. Send the design to them and offer it to them for $200 (or more). Very few business owners would turn that down.

Bidding For Gigs

Now you've got a small number (4 is a minimum) of sites in your portfolio, start bidding on website design jobs on your chosen freelance site. Remember that you are still there primarily to learn, so keep your bids low (perhaps around $300 to start with), give a detailed bid, pointing out similar sites you've made if possible, and be

responsive to messages.

Once you have your first website design gig, put your heart and soul into it, send them a great first draft, make any necessary changes that they request and then send a second draft.

I always make it clear that after the draft is agreed no substantive changes to the design (eg. colours, layouts) can be made. I would also recommend having a two stage payment release structure, so that half the fee is released when the design is agreed, and the rest on completion of the website. As before, make sure that the milestone is paid in advance, and that you are clear what needs to be done.

It's particularly important to clarify what you will provide and what the buyer will provide. Are you expected to create a logo? What about images? Any content?

Challenge 5: Get Your First Website Design Gig

So you've got your portfolio, you've identified some jobs to bid on, you've decided on your price, so get bidding! Again, it may take a while before you get accepted, and that's fine - keep going and keep positive. You'll get there eventually.

When you secure your first web design gig, post the result in the forums so we can give you feedback and congratulate you.

You're A Web Designer!

Nice work! Hopefully by now you will have completed four $100 web design jobs and two $300 jobs. I'm assuming you're still doing some small jobs (although we'll start to phase these out soon), and that you've earned $200 from those. That means a grand total of $1,400 in 10 weeks, not bad for a beginner!

At the same time, you should have more or less finished The Complete Web Developer Course, so you'll know about API's, making HTML5 apps for iOS and Android, and have built your Twitter clone. That's really impressive, and gives some opportunities to start developing some revenue-generating websites, which is where the real fun begins.

For now, grab a latte and congratulate yourself on a fantastic couple of months - the best is yet to come!

Progress Update

So far, you should have earned around $1,600:

Activities	Total Income ($)
Month 1 10 small freelance jobs at $20 each	200
Months 2/3 10 small freelance jobs at $30 each 4 website jobs at $100 each 2 full website jobs at $300 each	1200
TOTAL	**$1,600**

Further Reading

http://wordpress.org/

The home of Wordpress

http://wordpress.org/themes/

Free Wordpress themes

http://themeforest.net/

The biggest collection of premium themes

http://theme.co/x/

The X theme

http://theme-fusion.com/avada/

The Avada theme

http://www.elegantthemes.com/

http://themify.me/

http://www.templatemonster.com/wordpress-themes.php

http://themeroulette.com/

Some other sources of Wordpress themes

http://codex.wordpress.org/Using_Themes

The Wordpress guide to installing themes

Chapter Four: My Story

A Little Bit About Me

This might be a good point to tell you a little bit more about me. If you're not interested, feel free jump straight to the next chapter.

I coded a little as a youngster, messing around with BBC Micros trying (and failing) to recreate Zelda. When Windows came along and I hit my teens coding took a back seat to more pressing concerns, and was largely forgotten. I did a Mathematics degree at Cambridge University and went into teaching. After I got over my initial fear of my students I loved the job, but after a few years I started getting itchy feet, and started playing around with computers again.

Web development was the obvious path, as it required no special software, and the fruits of your labours could be shared instantly with the world. My brain buzzed with 'great' business ideas and I threw myself into building websites. I knew nothing about testing my ideas before building them, or customer development (if you don't know what I'm talking about pay attention in the next chapter!) but loved the technical challenge of building a website to do what I wanted.

I built a home exchange website called HomesExchange.org (other than appearing on a list of humorous domain names as HomeSexChange.org that one didn't go anywhere). I partnered with a friend of mine to make green-england.co.uk, an eco-friendly listings site. That one is still there, and was a minor success, but we both moved on to other things and it hasn't been updated for a few years now.

Whilst developing Green England, I looked for an eco-friendly web host. The options were few and far between. Those that were available were expensive and offered nowhere near the features that the big providers did. I decided I could do better, and ecowebhosting.co.uk was born.

I built the website myself with the help of a designer friend, and did a bit of basic SEO (Search Engine Optimisation - we'll look at that in the next chapter). Customers started signing up straight away, and growth has been steady (although never spectacular) from day one. The site now provides around half my total income, and requires about an hour's work a day.

When I'm not working on the site, I'm usually building other websites - some for myself, and some for others. I also enjoy the odd game of tennis and spending time with my wife and son (soon to be sons).

There are two things I'd like you to take away from my story. First, if you're

looking to build a money-generating website, your first idea is unlikely to be successful. Be prepared to keep trying out new ideas, improving your skills until you hit oil. Second, when coming up with ideas, try to 'scratch your own itch'. If you find yourself looking for something that doesn't exist, it's likely others are looking for that thing too.

Update For Second Edition

Since the first edition was written, a few things have happened. Most importantly, my second son, Ralphie, was born and is a very cheerful little chap. More unexpectedly, my courses have become the most successful ever on udemy.com, and have sold nearly half a million copies.

I am obviously thrilled with this development, but the thing I want to emphasise here is that success doesn't change your life as much as you think it will. My main sources of happiness before - my family and friends - are still my main sources of happiness. I worry less about money, but I still spend the bulk of my working day at a computer doing much the same sort of things as before.

One thing I am enjoying is only working on *one thing*. As a freelancer, you inevitably have several things on the go at once - a few freelance jobs, maybe a couple of small income-generating websites, your portfolio, maybe a part-time job. While I thoroughly encourage doing exactly that, it can be draining, and I remember at multiple

points thinking that I was doing several things pretty well, but nothing brilliantly.

I'm now lucky enough to have a small bunch of fantastic employees that keep the wheels turning, meaning I can focus on making courses and running my business - two things I thoroughly enjoy.

If there is one take-away I'd like to leave you with though, it's that the scatter-gun approach *works* (or at least it worked for me). Keep going at enough things, and be prepared to stop devoting time to things when they stop working, and you'll eventually find the one thing you can do better than 99.9% of other people. That's when you'll make stuff people love, and when things really start to come together.

Keep planning, keep improving and keep mercilessly culling activities that don't work, and you'll get to where you want to be. Just don't forget to enjoy the journey!

Chapter Five: Building Income-Generating Websites

Earnings Summary

Before This Chapter	After This Chapter
TOTAL EARNED: $1400 TOTAL TIME SPENT: 10 weeks	TOTAL EARNED: $5,000 TOTAL TIME SPENT: 4 months

I'm hoping that by now you're making at least $1,000 (that's two websites) a month from freelancer websites. My guess is if you've built a few sites for friends and colleagues you may have earned a fair bit more than that. Do share your success (or lack thereof) in the forums.

In terms of your skills, you should have finished the course and have had a fair

amount of practice building basic (and not-so-basic) websites. You should be comfortable with Wordpress and Bootstrap and have built one or two PHP/MySQL based websites. This is more than enough to start building sites of your own.

This Chapter Is Not For Everyone

I'll say it again, this chapter is not for everyone. For me, the potential of web development to provide a recurring income is fantastic. It's the ultimate freedom, and with two or three successes you could potentially never work again.

Having said that, it's not easy. It requires a lot of work up front, a willingness to spend time on marketing and customer support, and more than a little luck.

If you're interested in trying to build your own income generating websites, read on. The great thing is it requires very little financial investment, so the only thing you have to lose is your time. And you'll learn a huge amount along the way.

Still with me? Then let's talk ideas.

Generating Ideas

Essentially we're looking build an online business, with all that that entails. Every business starts with an idea, but how can you be sure it's a good one?

Generally, we want to keep to a particular niche. You're unlikely to build a competitor to Amazon or Google. But you might be able to build an 'Amazon for Education' if you have experience of selling products to schools, or a 'Google for Twitchers' if you know what questions birdwatchers ask that Google currently doesn't answer well.

The web has seen the concept of the Long Tail (http://en.wikipedia.org/wiki/Long_tail) develop, as the ease of reaching potential customers means that viable business can be made out of very specific products or services (eco-friendly web hosting for example). The 'long tail' refers to the the fact that while a large proportional of people's interests may be served by the mainstream, there are still millions of individuals with niche hobbies and specific business requirements, who are unlikely to have their needs met by large companies. These are precisely the type of people you should be targeting.

As I mentioned, you want to scratch your own itch. If you come across a problem that you can't solve, then it's likely others have had the same problem, and might pay for a solution. To paraphrase the saying, an idea without a problem is like a fish without a bicycle.

Challenge: Identify Your Itches

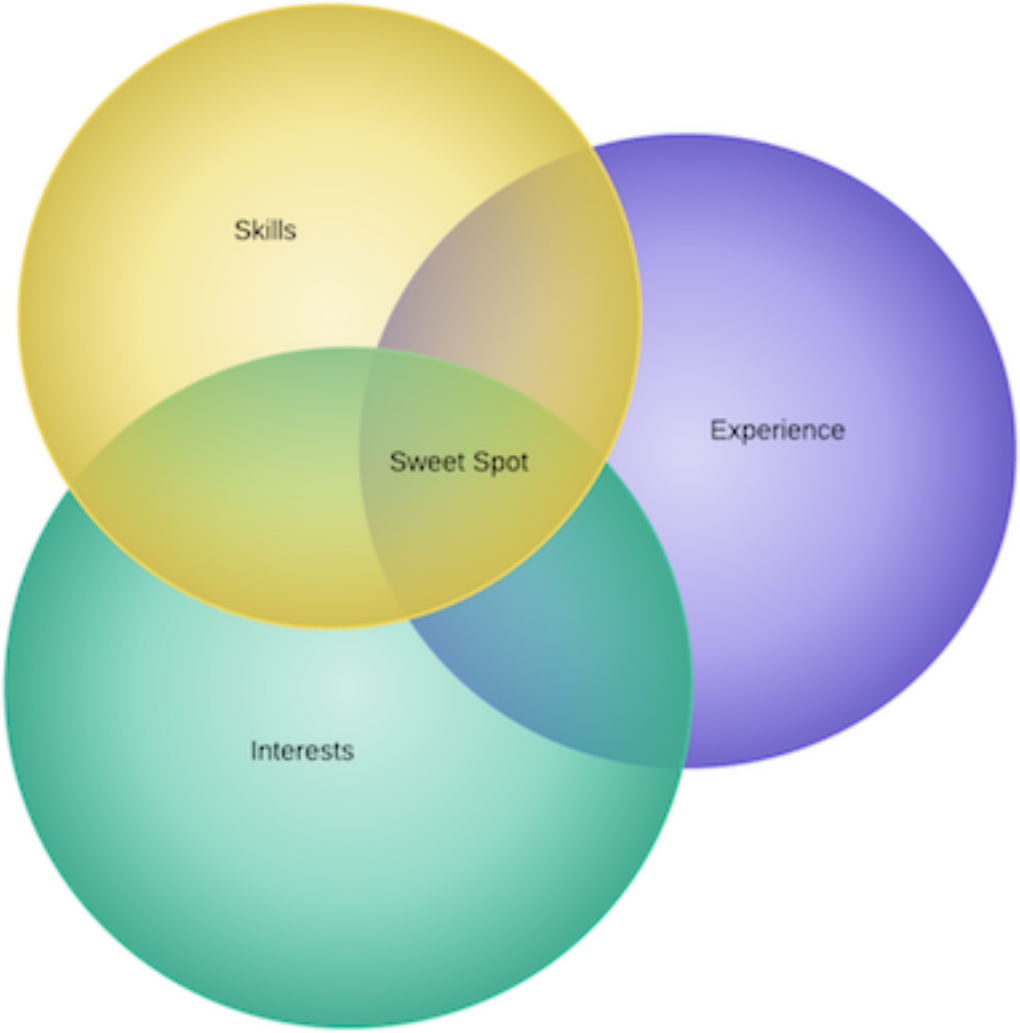
Spend 10 minutes thinking back over your work in the last two weeks. What was

frustrating? What could have been done more easily if you'd had the right tool? Make a list of these frustrations and share it with others in your line of work. Do they share your frustrations? Keep your list handy - you'll need it soon!

Find Your Sweet Spot

You're much more likely to come up with a good website idea if you have an interest in that particular field.

In fact, your best chances of success are if an idea falls in the overlap of your skills, interests and experience.



Challenge: Finding Your Sweet Spot

Draw two vertical lines on a piece of paper, dividing it into three equal columns. Title them Skills, Experience and Interests. As quickly as you can, fill the columns with your own skills, experience and interests. Ask your friends and family to add anything

they feel you've missed (it's easy to underestimate how any skills and interests you have). Once done, try to think of areas where the three might overlap - these will be fertile areas for you to consider building an online business.

Sources Of ideas

Still not got anything? Here's a few places you can look online for ideas:

https://news.ycombinator.com/item?id=7616910

Hacker News Idea Sunday

http://www.businessnewsdaily.com/1999-great-business-ideas-2012.html

19 Best New Business Ideas for 2016

https://news.ycombinator.com/item?id=7452630

Free Startup Ideas

http://www.springwise.com/top-10-business-ideas-opportunities-for-2016/

Top 10 business ideas & opportunities for 2016

http://www.telegraph.co.uk/finance/businessclub/sales/11051628/7-hot-start-up-ideas-that-could-make-you-a-millionaire.html

7 hot start-up ideas that could make you a millionaire

https://www.theselfemployed.com/start_ups/50-self-employed-business-ideas-can-start-100/

50 Self-Employed Business Ideas You Can Start For Under $100

As you can see, there is no shortage of ideas. The true value of a business is usually in the execution, not the idea.

Testing Your Ideas

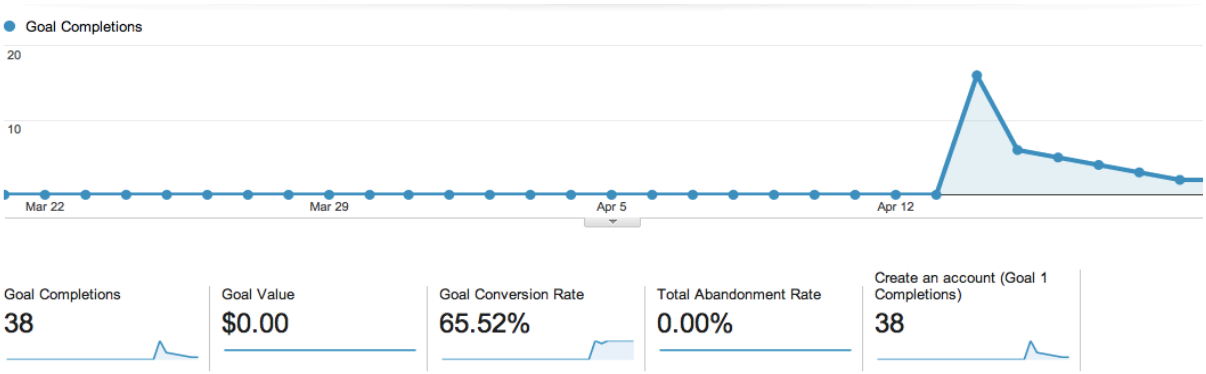
When I started out, I spent zero time testing out my ideas. I spent about an hour thinking about it, and if I couldn't think of a good reason not to build it, I sat down and started coding. In the early days, there's nothing wrong with this - you'll learn so much from building your own apps, and you'll have something else to add to your portfolio.

However, if you want to save time and effort, it's well worth testing them out first. Fortunately, there's a quick and easy way to do this.

First, put together a quick website explaining your idea. You can use a service like unbounce.com for this, or just build it yourself with Bootstrap or a Wordpress theme. There should be a clear description of what problem your app will solve, and how it will do it. There is no need to talk about price at this stage.

Then, add a Mailchimp.com form to the site, and set up google analytics to test conversion rates (that is, what percentage of visitors sign up to your list). I describe how to do this in the last chapter of The Complete Web Developer Course. Drive traffic to this website using your Twitter feed and Google Ads or Facebook Ads. Spending £100 on Facebook ads will be well worth it if it saves you 10 days' of development time! You can also post your idea on sites like reddit, Hacker News or ratemystartup.com.

I did this when I built completewebdevelopercourse.com, and got the results below:



The numbers weren't huge, but 65% of people who visited the website signed up to hear more about the course. That was enough to convince me that there were enough people who would want this course to make it worth my while building it.

Challenge: Test An Idea

Why not give it a try? Pick the best idea you've come up with, build a quick showcase website for it. Use the links at the end of this chapter to see some examples of great landing pages, and also find sources of free stock photos you can use to jazz up your page. Keep it simple, and post your results on the forum. At the very least, you'll learn a huge amount and have another site for your design portfolio!

What now?

So how did it go? If your conversion rate was lower than you'd like, maybe the idea needs tweaking, or perhaps you should try something else.

If you got a decent response, then congratulations - it's time to get to work! Start putting your site together, focussing on the core features that the product needs. This is known as the Minimal Viable Product: what is the minimum you can make that people will pay for?

Once you've got something that you think people will pay you for, ask a small selection of your mailing list to try it out and give you feedback. Make changes based on their feedback (known as 'iterating'), and keep going until you have a handful of paying customers. Then email your list telling them the product is ready, and continue the feedback—> iteration cycle.

There is obviously a lot more to building a great product than that, but this isn't the place for a detailed examination of building and marketing products. If you're planning to build your own products, I'd recommend Rob Walling's *Start Small Stay Small* - it's full of hard-earned advice on launching your own business. You can purchase it at http://www.amazon.co.uk/Start-Small-Stay-Developers-Launching-ebook/dp/B003YH9MMI. Rob also co-hosts the excellent *Startups For The Rest Of Us* podcast.

Progress Update

We're 5 months in, and by my calculations we're on $5000:

Activities	Total Income ($)
Month 1 10 small freelance jobs at $20 each	200
Months 2/3 10 small freelance jobs at $30 each 4 website jobs at $100 each 2 full website jobs at $300 each	1200

Months 3/4 5 small freelance jobs at $40 each 4 full website jobs at $500 each 2 users on your web-based startup at $20 each per month	2280
TOTAL	**$5,000**

Further Reading

http://www.theguardian.com/small-business-network/2013/feb/19/how-to-find-your-business-idea

How to find your business idea

http://www.smarta.com/advice/starting-up/business-ideas/

Business idea guides

http://jmarbach.com/solve-problems-dont-build-ideas

Solve problems - don't build ideas

https://github.com/mmccaff/PlacesToPostYourStartup

An exhaustive list of places to post your website idea

http://designrope.com/design/find-stock-photos-dont-suck/

A great source of free stock photos

http://unbounce.com/landing-page-examples/built-using-unbounce/beautiful-landing-page-design-examples/

http://www.formstack.com/the-anatomy-of-a-perfect-landing-page

http://blog.hubspot.com/marketing/landing-page-examples-list

http://www.wordstream.com/blog/ws/2014/03/05/landing-page-examples#.

Examples of great landing pages

http://www.amazon.co.uk/Start-Small-Stay-Developers-Launching-ebook/dp/B003YH9MMI

Start Small, Stay Small by Rob Walling

http://webappsuccess.com/

A Practical Guide To Web App Success by Dan Zambonini

Chapter Six: Finding Profitable App Niches

Earnings Summary

Before This Chapter	After Chapters 6 & 7
TOTAL EARNED: $5,000 TIME SPENT: 4 months	TOTAL EARNED: $9,460 TIME SPENT: 6 months

Building HTML5-Based Apps

The great thing about building HTML5-based apps is that you can (in theory) build once for all platforms: web, Android, iOS, and all screen sizes. In practice, the

more your app uses platform-specific features, such as cameras or gyroscopes, the harder it is to build a single code-base for each platform.

I'd recommend focussing on simple apps that are text-based, without too much animation or reliance on device-specific features. Avoid resource-intensive apps like games and video players, and instead consider information apps, integration with web services, and online sync. Make the most of the fact that your app is cross platform by providing something that users will want access to from all their devices, such as a shopping list app or reminder apps.

Use a mobile framework like jQuery Mobile, Sencha Touch, PhoneJS, IonicFramework - check out a full comparison at http://mobile-frameworks-comparison-chart.com/. The frameworks make your life much easier when developing for many platforms, and usually provide all the standard pages and widgets you might need.

You definitely need to focus on a niche: don't try building the next camera or email app. But do consider building an app marketed at fishermen, or ballet dancers, especially if you have some expertise in the area.

For full details on actually building mobile apps with HTML5, check out The Complete Web Developer Course. In this chapter I'll focus on finding profitable niches

for which to build income-generating apps.

Generating Ideas

Much like the web as a whole, the app stores are not the free-for-all they once were. There are a large number of developers, so you need to work harder to find gaps in the market and lucrative app opportunities.

To develop your app ideas, follow the same path as with the previous chapter. Are there any apps that you'd love to have, or your friends or colleagues have suggested? Once you have some ideas, follow the steps in the next section to see if they are worth continuing with. If not, bin them and move on.

If your idea fountain runs dry, try some of the following ideas.

Search the app stores for apps which have large numbers of downloads but poor reviews. It's likely that if you build a better app that provides the same features, their users will migrate to your app.

Find popular apps which are only in English, and consider producing a similar app in a different language. Increasingly, non-English speakers are looking for apps in their own languages, especially if the content is tweaked to be more relevant to them at the same time.

Similarly, look for opportunities to create location-specific apps, such as City Guides, discount card apps or local messaging-based apps. How about an app for people to find tennis partners near them, or a Tokyo-only lonely hearts app?

Search for apps that are successful, but only available in a particular geographical area. Lots of apps start out in San Francisco only as a result of the Silicon Valley effect. Replicate the functionality and put a local twist on it to make it work even better in your city.

Look for companies that have a website with a customer login area but no app. Build an app which provides a better experience than the website when on mobile. Not only will you have easy search visibility, but there is a chance the company will buy your app from you!

Challenge: Generate 5 Ideas

As in the last chapter, try and come up


THE COMPANION TO
THE COMPLETE WEB DEVELOPER COURSE
How to make $10,000
while learning to code
Second Edition
By Rob Percival

least 5 solid ideas for a great app. The functionality should be minimal, and the benefits clear. Then ask your friends and family which they think is best (it's always preferable to offer a friend a choice between several ideas rather than asking them if they like a particular idea).

Testing Your Ideas

You should go through a similar process to that described in the previous chapter to test your ideas: create a simple landing page with a signup form, drive traffic to it using Google or Facebook ads, and analyse the conversions. If they look good, go for it!

Having said that, in the early days of your development career, you will learn a huge amount just going through the process of developing, managing and marketing an app. So you don't necessarily need to fully validate every idea. Make the most of the learning experience, and just build something. If it fails, no-one will judge you and your second idea will be the better for the failure.

Simplify your app down to the bare bones - what is the killer feature, the one thing that it will do better than any other app out there? Make it awesome at that task, and don't spend too long on the bells and whistles.

Challenge: Build An App

Once you've sifted through your ideas, commit to one idea and make it happen. Create a minimal viable product and submit it to the app stores. Your first app store presence is always an exciting moment - share it with us on the forums (and get a few early downloads!).

Look in the Further Reading for tips on marketing your app and creating a great user experience.

Further Reading

Finding app ideas:

http://freelancedoodle.com/app-dev-2-finding-a-killer-app-idea-and-defining-guidelines/

http://www.theguardian.com/theguardian/shortcuts/2013/mar/26/how-to-become-an-app-millionaire

Marketing your app:

http://www.apptentive.com/blog/10-inbound-marketing-tips-for-mobile-apps/

http://info.localytics.com/blog/mobile-app-marketing-best-practices-slideshare

http://mobiledevices.about.com/od/marketingapps/tp/Top-10-Tips-To-Market-Mobile-Application.htm

http://www.businessinsider.com/top-app-store-marketing-tips-2013-10

http://blog.kissmetrics.com/master-mobile-marketing/

https://blog.kissmetrics.com/mistakes-in-app-marketing/

http://www.entrepreneur.com/article/228328

Designing a great app:

http://mashable.com/2012/04/11/mobile-app-design-tips/

http://www.creativebloq.com/tag/App-design

https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/

http://www.smashingmagazine.com/2009/08/11/how-to-create-your-first-

iphone-application/

Chapter Seven: Building Affiliate Websites

What Is An Affiliate Website?

Affiliate schemes have been a core part of the internet almost since its inception. Essentially, it works like this: you build a website that attracts traffic, such as a blog or a free online tool. You then join an affiliate scheme for an e-commerce site such as Amazon. You can then get special links to products on Amazon, which you put on your site. If someone clicks on one of those links and buys *anything* from Amazon on that computer within a certain time period, you get a cut of the sale.

Affiliate websites are unlikely to make a huge amount of money. But if you have a knack for it revenues of $20-50 per month are fairly common, and if you build ten or twenty of them…well, you can do the sums. It's not quit-your-job cash, but it's a nice supplementary income and the sites usually need very little maintenance once built.

If you're serious about building affiliate sites, I'd recommend looking at www.affilorama.com. They have a lot of free information for affiliates, as well as tools and guides that go well beyond the scope of this book.

Which Affiliate Scheme Should You Join?

I won't go into the relative merits of all the affiliate schemes out there, mostly because it will depend on your niche. You'll want to find a site that is relevant to the users you are targeting, and converts well. If in doubt, amazon.com is always a good choice - their affiliate scheme is reasonably generous, and they have great name recognition amongst internet shoppers.

Again, you can get much more detailed advice on the schemes available at www.affilorama.com.

Ideas For Affiliate Sites

As with the previous chapter, finding a good niche is critical to creating a successful affiliate site. The niche should be:

- something that people are searching for on Google
- uncompetitive (ie. there are not many other sites ranking well for that niche)
- easily matched to specific products that people might buy

If you have particular experience in that niche, all the better, as you are more likely to understand the people searching for it.

You can use the Google Keyword Planner (http://adwords.google.co.uk/KeywordPlanner - requires a free Google Adwords account) to find keywords that have large numbers of searches but low competition. Do Google searches for those keywords and see what the top sites are like. Are they relevant to what the searcher is looking for? Are they high quality sites or are they spammy and unmaintained?

As a quick example, a year ago I was looking to buy a home video projector. I soon realised that the 'throw distance' of a projector is particularly important if your projector needs to be far away from your screen. Calculating the 'throw distance' required for your room is straightforward, but it took me a while to find a site that explained it. I searched for 'throw distance calculator', but the top sites were not quite what I needed. So I purchased throwdistancecalculator.com and built a simple site to calculate a user's throw distance. Once the user has used the calculator, a link to purchase projectors on Amazon appears.

This niche was certainly uncompetitive, and easily matched to a purchase (almost everyone calculating their throw distance will be looking for a projector). The number of people searching for it, however, is very low, so even though the site is the #2 result on Google for 'throw distance calculator' and it converts fairly well, it only

brings in around $15 per month.

Challenge: Find 5 Niches

Brainstorm areas that you are familiar with that might work well as niches. Use the Google Keyword Planner to examine the competitiveness and popularity of keywords related to that niche. Finally, google the keywords to see how relevant the results are. Try to come up with 5 sets of keywords that might work well.

Building Your Affiliate Site

You need to offer something of value to your potential users. In the case of throwdistancecalculator.com, this was a simple tool to help them calculate their throw distance. You might consider building a site to help people with their taxes (I can never find the right tax information when I need it). Or a site that tells people the time of high and low tide in their area. Keep it very simple and specific.

An alternative to a tool is to provide information. Perhaps a list of the best places to visit in your home town. You could build a web scraper to collect information from a site such as imdb.com, and make film suggestions for specific genres. Whatever you choose, your site should have a very simple, single purpose, with an obvious set of products to purchase once the user is finished.

Ideally, choose a domain name containing your top search term, such as

throwdistancecalculator.com. Make it a .com domain if possible, or country-specific if your site is related to a particular country.

Keep your design simple and focused, with a very obvious ‘call to action’ - a button to click, or a text field to fill in. If your information or tool is particularly valuable, you might want to ask the user for their email address so you can keep them updated about your site, but that is not the primary goal here.

Once your tool works well, or your information is clearly set out, integrate your affiliate links in an unobtrusive, natural way, ideally so that the user clicks the link immediately after they have got the information they came for.

Challenge: Build An Affiliate Site

Take your most promising niche from the previous challenge, decide on the keywords you will target and purchase a domain name. Then build your tool or upload your content, and post a link to it on the course forums for feedback.

Marketing Your Affiliate Site

Website marketing (or SEO - Search Engine Optimisation) is a hugely complex business, but I’ll cover the basics here. To rank well in the search engines, you need to do two things: have relevant website content, and build links to your website.

Hopefully, you should already have relevant website content if your site is useful to your users. If you are building a tool that doesn't require much explanation, put a few paragraphs of text describing what your tool does in more detail underneath the tool itself. Make sure you include your keywords and related words there. Ensure the website title contains your keywords, and that your website description is clear and relevant.

Building inbound links is a bigger challenge. If you have built relevant websites, put links on those to your new site saying 'We support abcd.com' or 'Check out abcd.com'. Post on relevant forums, explaining why your tool is useful and posting a link asking for feedback. You could offer to write a 'guest post' on a relevant blog, giving their users some useful information in exchange for a link back to your site. There are myriad other ways to build inbound links, and I've included a number of articles with a range of suggestions in the bibliography for this chapter.

Monitoring Your Affiliate Site

Make sure you sign up to Google Webmaster Tools and Google Analytics for your site, and keep an eye on your search rankings and traffic levels. Beyond that, there's not much to do other than sit back and wait for the dollars (or possibly cents!) to roll in.

If you have some success, build another site and slowly you'll build up a

potentially significant income that requires no maintenance or upkeep.

Progress Update

We're getting close! I'm assuming conservatively that you've only done 4 website jobs in the last two months, that you've had 1 moderately successful affiliate website, and that you've managed 1,000 downloads of your app across the App Store, Google Play and the Windows Store at $1.99 each.

Activities	Total Income ($)
Month 1 10 small freelance jobs at $20 each	200
Months 2/3 10 small freelance jobs at $30 each 4 website jobs at $100 each 2 full website jobs at $300 each	1200

Months 3/4 5 small freelance jobs at $40 each 4 full website jobs at $500 each 2 users on your web-based startup at $20 each per month	2280
Months 5/6 5 small freelance jobs at $50 each 4 full website jobs at $500 each 5 users on your web-based startup at $20 each per month 1,000 downloads of your app at $1.99 each Affiliate income from 1 website at $20	4460
TOTAL	**$9,460**

Further Reading

Introductory guides to affiliate marketing

http://uk.cj.com/what-is-affiliate-marketing

http://www.problogger.net/archives/2009/07/07/what-is-affiliate-marketing/

https://www.youtube.com/watch?v=KRiYsuJJuHc

http://www.seanogle.com/entrepreneurship/how-to-build-a-niche-site

http://www.2createawebsite.com/money/affiliate.html

Finding ideas for affiliate sites

http://nichehacks.com/profitable-niches-for-affiliate-marketing/

http://www.smartpassiveincome.com/niche-selection-tips/

http://www.affilorama.com/blog/cooking-affiliate-programs

http://sugarrae.com/affiliate-marketing/finding-a-niche-in-affiliate-marketing/

https://www.ventureharbour.com/10-steps-to-a-succesful-affiliate-marketing-strategy/

Affiliate company comparison

http://affiliate-marketing-services-review.toptenreviews.com/

Marketing your affiliate site

http://marketingland.com/7-big-mistakes-new-affiliate-marketers-make-19195

http://www.theguardian.com/money/2011/may/21/affiliate-marketing-lucrative

Chapter Eight: Bonus Material

Earnings Summary

Before This Chapter	After This Chapter
TOTAL EARNED: $9,460 TIME SPENT: 6 months	TOTAL EARNED: $15,515 TIME SPENT: 9 months

In this chapter we'll look at four other ways you can earn money online.

I'm not including any of the earning potential of these ideas in the above total: instead I recommend you build a second app, and continue to develop your online business and build more websites.

That's because the suggestions in this chapter can be a little more tricky to make substantial returns on, and you don't need to do them to be successful. Having

said that, the returns can be great - I've earned $60,000 from the first idea alone, and I'd definitely recommend trying them out.

Find An Organisation That Needs You

One problem that people have when hiring web developers is that they don't know what they want. So find an organisation that you think you can help, and approach them with an offer. This might be a business with a crummy website that you can improve upon, or an organisation with some great content that is poorly presented. There could be a website that would work better as an app, or a collection of physical materials that could be digitised and sold.

Ideally this will be an organisation in a niche you're familiar with. For me it might be a school or university, or some other educational institution. It should be local to you (you will likely need to meet with them in person) and preferably you will have had contact with the organisation in the past.

Put together an offering to take to the organisation to show what you can do for them. This doesn't need to be a written plan, but it should be easily explainable and provide a tangible benefit to the organisation, such as gaining them revenue or exposure. You should have a rough idea of the overall costs and timescales.

Then, simply approach someone at the organisation, preferably someone you

have had contact with in the past, and offer to buy them coffee or come to their office for a brief chat. If you've thought through your plan well, it's likely they will agree with you, and they will be impressed that you've had the initiative to contact them about it.

Have your portfolio to hand, as they will want to see the sort of work you do, and be gracious if they turn out not to be interested. They may well think of you in the future, if and when they decide to go forward with a new website or app.

Challenge: Find 5 Organisations

Make a list of the organisations that you have worked with in the past, and consider what you might be able to do for them. Apps are particularly popular (and lucrative), but a fresh website design or new features can go down very well.

Post your successes (and failures!) in the forums - I'd love to hear from you.

Sell Your Scripts Online

Selling scripts is another idea that is unlikely to make you large amounts of money, but if you're persistent and produce good quality, useful scripts could well provide a nice supplementary income. In less than two years, Wim Mostmans — better known as Sitebase on CodeCanyon — created and listed 55 scripts, and has amassed over 10,000 sales.

Quite simply, if you find yourself writing a bit of code for a website that achieves some useful function, or some attractive CSS buttons or forms, consider packaging it up and selling it to others. You can also do this on a larger scale with themes - if you design a great site from scratch, give it some generic content and offer it to others for a small fee.

Take the CodeCanyon user jigowatt (http://codecanyon.net/user/jigowatt). He's sold over 22,000 scripts, at an average of $5 each - that's over $100,000. Not everyone will be that successful, but it's certainly worth giving it a shot.

Challenge: Sell A Script

Have a look through codecanyon.com and themeforest.net at the sort of scripts that sell well, and consider if you've written anything that could be repurposed as a standalone script. Make sure the code is well structured and commented and upload it. Post it on the forums for a couple of quick sales!

Sell Addon Products

As a web developer, you have access to a wealth of knowledge and understanding that your clients don't. Make the most of this by offering them extra products when you fix their website or build them an app.

Web hosting is a common add-on for developers to offer (you can buy a reseller

hosting package from Eco Web Hosting which allows you to sell as much hosting as you like for £19.99 per month). You get a small amount of recurring revenue at little or no cost to yourself.

Marketing and SEO are also very useful extras to clients, as is ongoing maintenance and support. Be creative, and give them a reason to take all their services from you. Offering a one-stop-shop is a huge time saver for your customers.

Challenge: Sell Your First Addon

Try it out with your next sale - offer your customer some marketing, or web hosting, or anything else that you can provide. They may well turn it down, but at least they will be aware that you offer that service, and may well come back to you in the future.

Post your success (or otherwise!) on the course forum.

Buying Websites

This is a risky one, which is why I've put it last. It's the only suggestion I'll make which requires any sort of upfront investment beyond your time, and should not be done without considerable planning and investigation.

Essentially, you purchase either the code or a full active site, and improve and

market it yourself for a profit. If you choose carefully, you can do very well, and save yourself a lot of time over building and marketing a business from scratch. Make sure that you investigate any businesses very carefully, however, as there's rarely any chance of a refund.

The most popular website for buying small online businesses is www.flippa.com. Check it out, and see if anything interests you.

Flippa have put together a thorough guide for anyone interested in buying websites, which you can access at www.flippa.com/pro-guide-to-buying-websites.pdf. I won't reinvent the wheel here - suffice to say make sure you know what you're buying and how you are going to improve and market it.

Progress Update

In just over half a year, you've learned a huge amount: HTML, CSS and Javascript on the front-end site, and PHP/MySQL on the back-end. You've built some great sites with Wordpress and Bootstrap, and learned how to integrate web services into your sites with API's. You've built HTML5-based apps for the Android and iOS app stores.

You've also earned a lot - by my calculations you should have made revenues of

around $15,000. Subtracting from that the various fees from freelancer sites, app store fees etc, I make that $10,000 and change. I haven't factored in any of the work that you've found outside of this, using the methods in this chapter or your own contacts, so my hope is you've actually earned a lot more.

Here's my breakdown:

Activities	Total Income ($)
Month 1 10 small freelance jobs at $20 each	200
Months 2/3 10 small freelance jobs at $30 each 4 website jobs at $100 each 2 full website jobs at $300 each	1200
Months 3/4 5 small freelance jobs at $40 each 4 full website jobs at $500 each 2 users on your web-based startup at $20 each per month	2280

Months 5/6 5 small freelance jobs at $50 each 4 full website jobs at $500 each 5 users on your web-based startup at $20 each per month 1,000 downloads of your app at $1.99 each Affiliate income from 1 website at $20	4460
Months 7/8 4 full website jobs at $500 each 20 users on your web-based startup at $20 each per month 1,500 total downloads of your two apps at $1.99 each Affiliate income from 3 websites totalling $100	5875
TOTAL	**$14,415**

Further Reading

Marketplaces to sell your scripts

http://codecanyon.com

http://phpmarket.com

http://hotscripts.com

http://sourcecodeshop.com

http://spikesolutions.com

http://binpress.com

http://flippa.com

Popular marketplace for buying and selling websites, domains and business.

www.flippa.com/pro-guide-to-buying-websites.pdf

Definitive guide to buying websites

Chapter Nine: Epilogue

Where Do You Go From Here?

I hope you've enjoyed this book, and have managed to follow the steps to make $10,000 while learning to code. You'll notice that some of the income (your apps, your scripts, your online business and your affiliate sites) is ongoing - you don't need to do anything to keep it coming in (although a little marketing here and there never hurts!).

A key question to consider is whether to focus on the day to day work of bidding on freelance jobs and building your portfolio, or building apps and websites which are more risky but offer potentially higher (and ongoing) returns. What you choose will depend on your temperament and situation, but I like to keep a mixture of both in my work portfolio.

You now have the skills and experience to call yourself a Web Developer without any of those "I'm a fraud" worries, so congratulations! Spend some time thinking about how far you've come and where you want to take your developer career. You may decide you want to keep the day job and keep your web development as a hobby or side income, or you might want to take it on full time.

Whatever you do, I wish you the best of luck, and hope you'll keep in touch on the course forums. And if you have a moment to write a quick review of this book on Amazon, I'd be eternally grateful.

— Rob Percival