starting out

It's good to start building sites for money. You learn to deal with customers, manage expectations and prioritize work.

The site in question is http://www.tranna.co.za/

Business is all about exchange of value. Will Tranna benefit from your work? More bluntly, will they make more money from having a nicer website? Are they currently losing customers because of their current website? Whatever they make from the exchange, that's what you can charge minus one Rand. If you can deliver the same value at a lower cost (less time/work) than a competitor, you can out-compete. That is, if you don't hate your life making retail websites.

I published some thoughts on retail site design: http://petrustheron.com/posts/retail-landing-page-design.html

Their current site is actually pretty good, just outdated. It's clearly branded, fast and contact details are on the front page. The worst aspects for me are:

- No analytics (how many visitors are they getting?)
- Key info is below the fold
- bad quality of food photos
- Photo menus instead of regular tables, which loads longer and is hard to work with, esp. on mobile.
- Not mobile friendly

Be wary of making their website worse by modernizing it. As designers, we are tempted to design "for the sake of design", when good design is as little design as possible. Almost any change you are likely to add will make it load longer, so you have to bring something to the table.

some general things on freelancing:

1. Never work for free. There are many reasons, of which the main one is that you won't get honest feedback on your work, which can lead you astray. Always charge.
2. Take 50% up-front.
3. Get something in writing.
4. Only start work when the initial 50% is paid.
5. Retain copyright until full payment is made and only publish work on a domain name you control until such time.
6. You are going to get screwed unless you are diligent, either on purpose or by proxy.
7. Software development time is notoriously hard to estimate, so charge for time instead of a project-fee if you can. "6-8 hours" is code for "3-7 weeks". "3-7 weeks" mean 2 years. So break down any task over 30 minutes into smaller tasks. You will find that it balloons pretty quickly.

Approach them with *their best interests* at heart, i.e. "I can help you make more money. You are losing customers because X. I can fix that with Y and it will only cost Z." So the first thing I would do is add Google Analytics for them and watch their traffic for a week to guide your design.

more on the above

Writing code is a great way to fund other interests, esp. when you're young :)

1. Their markup is that verbose because their site was made in MS Publisher - a Wysiwyg tool. XML sucks - write Lisp instead!
2. Don't work for free or do any work up-front. Rather work on something interesting or learn something new than trying to "tentatively" add value for them. Unsolicited redesigns usually don't go down well, so solicit first.
3. Ask if they are interested in a redesign. Focus on their benefits: clearer brand communication, increased usability.
4. If they are interested, they will ask about cost. Tell them your fee and terms. You will learn more this way.
5. If you learn to use Git and publish some code on GitHub, or publish your own website, I will hire you in a second.
6. Learn Clojure because it will be the highest-paying skill in 3 years. Reason: it is the most productive, yet undervalued tool on the market. Also, understanding Lisp (and macros) will make you a better engineer for all your days.

Hiring a designer without a personal website is like getting a haircut from a bald barber :).

“I have never let my schooling interfere with my education.”
— Mark Twain

advice regarding education

High school tries to avoid producing dropouts instead of producing exceptional people. If you are faster than the teacher, they are likely to say, "Slow down. I need you to be over here...with everyone else. Today we are learning about *varrrriables,*" instead of saying, "Here, go implement quicksort."

So you have to start taking responsibility for your own education.
Read There Is No Speed Limit by Derek Sivers.

Re: git. Dropbox was the right tool for you at the time, but you need to learn a version control tool. There are two major ones: Mercurial (hg) and Git (git). IMO, Mercurial has a cleaner interface, but Git has won in the marketplace because GitHub is so good. I pay them money every month.

Both git and hg have vastly different branching architectures and philosophies about forking, but don't waste time learning the differences or getting into flame wars about the merits of each. Just use Git - it already won. Git probably has the worst CLI (command line interface) ever invented (except for the tar command), but the storage model is simple. Git just stores snapshots and then diffs between them.

There is a big difference between Dropbox and a source control tool: merging and pull requests.

Go through this excellent Hg tutorial: hginit.com

Re: Tranna. If they are nearby, just go over there. Or phone them.

How does one register a domain name?

It's so complicated that I built a domain registrar for .co.za domains called MyNames.