Thats all!

2010-2012, Rails Girls.

Contact us at team@railsgirls.com

railsgirls.com

Rails Girls

Your cookbook to the world of web development.

railsgirls.com

My first investor deck

The Team
Describe your team. What makes you special? Why are you going to succeed?

Problem
Describe your problem, how it exists today? How come it's a problem?

Solution
Describe your solution. Make it tangible, real and descriptive.

Revenue and Competition
Who is paying for what? What are they paying for? Who are your competitors?

The Magic
What does the future look for you? What is your vision? What is your mission?

Make it real,
Make it happen

Coding is only part of the fun. As soon as you have an application up and running, you should start thinking about your users and how to keep your app running.

Use this space to sketch out the first e-mail newsletter template to your users.

This book belongs to

Get excited and make things

Software changes cultures. We believe that in a world where everything involves the Internet, we need to get the girls involved too.

Building web applications is modern craftsmanship. Focus on building tomorrows products. Learn to solve problems. Be curious and learn how things work.

RailsGirls can provide you the ingredients, but come up with recipes and cook.

Start with simple mockups of your idea

Balsamiq http://balsamiq.com

Mockingbird http://gomockingbird.com

Prototypesapp http://prototypesapp.com/

Design

Dribbble http://dribbble.com/

CSSgrid http://cssgrid.net

Colour Lovers http://www.colourlovers.com

Marketing Magic

Shopify http://www.shopify.com

Moo http://www.moo.com

Stickermule http://www.stickermule.com

Customer service & feedback

UserVoice http://www.uservoice.com/

Wufoo http://wufoo.com

Google Docs (Forms) http://docs.google.com

Developer communities

Github http://www.github.com

Stackoverflow http://stackoverflow.com

Forrst http://www.forrst.com

Social

Facebook Social Plugins http://goo.gl/wSkVL

Twitter Resources http://goo.gl/UOMhs

Get going with something small

Once you've gotten the first bite of coding web apps, try building on top of it.

The idea is not to start building a big app like Facebook. Try creating something simple, yet interestin,which you can complete in just a couple of days.

Focus on one problem you want to solve at a time - this is what building applications is about.

Knowing how

"I've long argued that UI design, programming, and product strategy should be learned apprentice-style with your hands and through experience, not through school and pedagogy. When I go to conferences about design I see a lot of declarative knowledge. Knowledge of. The latest CSS rules. The new JavaScript syntax. Ten ways to make users happy (supposedly) or whatever else.

What I don't see are procedures—somebody standing up there with a pen or a text editor and making things happen and showing how it's done. That's what I want to see and that's what I think our industry needs more of."

- Ryan Singer, 37 Signals

Futher Reading:

The first step is to start
http://goo.gl/02u5

Code as Craft by Etsy
http://goo.gl/5iX7L

Why Software is Eating the World
http://goo.gl/ry3Sc

Product Managers in Modern Development teams
http://goo.gl/TjRB6

Have we all become hackers
http://goo.gl/PA8e2

A Tour of the Design Process at 37signals
http://goo.gl/Fc49

5by5 Podcasts
http://goo.gl/QJqU

Don’t daydream, Just design!

Draw. Sketch. Wireframe. Mock up.

Plans and ideas are nothing compared to something you can see, touch, hear or smell.

Most applications are just screens with boxes, anyone can draw them. If you are having trouble, try talking to people about it and flesh it out first.

Drawing captures your idea into something concrete and real.

Getting futher

Try Ruby
http://www.tryruby.org
Programming Ruby in your browser.

Codeschool
http://www.codeschool.com
Try out the Rails for Zombies module: a free, browser-based Rails workshop. Other classes interesting too!

Codeacademy
http://www.codeacademy.com
A new Y-Combinator startup aiming to teach the world to code.

Don’t fear the Internet
http://www.dontfeartheinternet.com/
Frontend stuff (CSS & HTML) explained through short videos.

A Quick (and Hopefully Painless) Ride Through Ruby (with Cartoon Foxes)
http://mislav.uniqpath.com/poignant-guide/book/chapter-3.html
The legendary Why’s Poignant Guide to Ruby: part comic book, part coding exercise.

RailsGuides
http://guides.rubyonrails.org/
Tutorials and guides created by the Rails community

Cheatsheet for your app

Your Ruby on Rails app can be found at:

```
projects/railsgirls
```

Start your app by opening Terminal and:

```
cd projects/railsgirls
rails s
```

Access your app:
Go to `http://localhost:3000` in your web browser

Most of the time you edit the files at app/ directory

- `app/views/` - the user interfaces of your app
- `app/models/` - models define app information structure
- `app/controllers/` - are the glue between models and views
- `app/<other>/` - some code will not fit in the 3 buckets above; feel free to create other directories and files

Tools we use

- **Twitter Bootstrap** (to style the app) http://goo.gl/YC53s
- **Sublime Text** (to edit the code files) http://www.sublimetext.com/
- **Heroku** (to host the application) http://heroku.com

Checklist for concepts

Define the core idea for the project/product.

What's the problem you're trying to solve?

Why does the problem need solving?

Who is the intended customer?

What are the goals of the product?

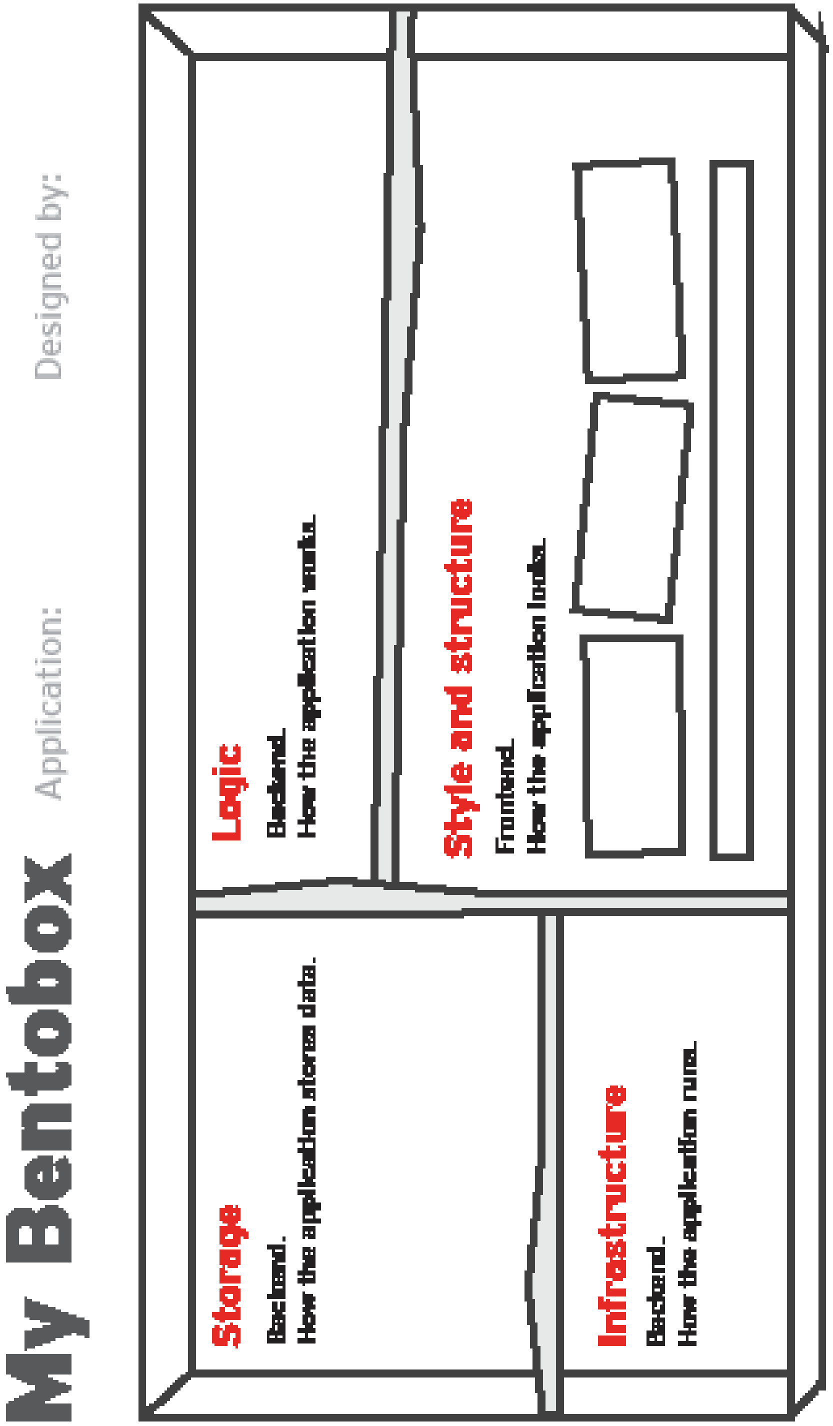
When are you going to ship the first version?

What should be in the Minimum Viable Product?

Lunch box your web app

The Bentobox is our conceptual model for understanding web applications and their components.

The Japanese lunchbox model aims to have a balanced mix of things. Much like web apps, the “Ekiben“, looks for content which is arranged in the most efficient, graceful manner. The bento is presented in a simple, beautiful, balanced way. Nothing lacking. Nothing superfluous. Not decorated, but wonderfully designed.


My Bentobox
Application:
Designed by:
Logic
Backend.
How the application works.
Style and structure
Frontend.
How the application looks.
Storage
Backend.
How the application stores data.
Infrastructure
Backend.
How the application runs.