Site Planning Outline - Max Moulds

Site Plan

Purpose

- Main goals of site are to gain familiarity with common website building blocks.
- Development of new skills in web development
- Make a modern website

Audience

- I will use this site for my own personal website and for future templates for website so the audience would be anyone interested in this type of website.

User acquisition

- Gaining new users is not an immediate concern of mine, actually keeping traffic manageable is more my goal right now. Many of the future designs of my site contain elements I am not certain of my ability to support.

Content

- Initially the site will contain write-ups of my own issues around the site and deployment as a testament to technical support and documentation associated with my work. Then I will transition into a custom weighted feed aggregator.

Look and feel

- The site will mainly be for full screen users but have a mobile version with as few limitations as possible. Scalable and minimal designs along with a logical navigation will be the dominating points of design.

Layout

- I will try and use a collection of modules all unified under a single design umbrella like Google Material Design. I will also try and limit the implementations to JS/CSS3/HTML5 and a handful of core wide use libraries/frameworks and eventually morph into a custom library.

Schedule

- Main page, splash, login, article, contact, legal, and faq are all reasonable simple and should be completed by the end of this class. Media and js element pages are fairly complex. The mobile only elements are also a ways off.
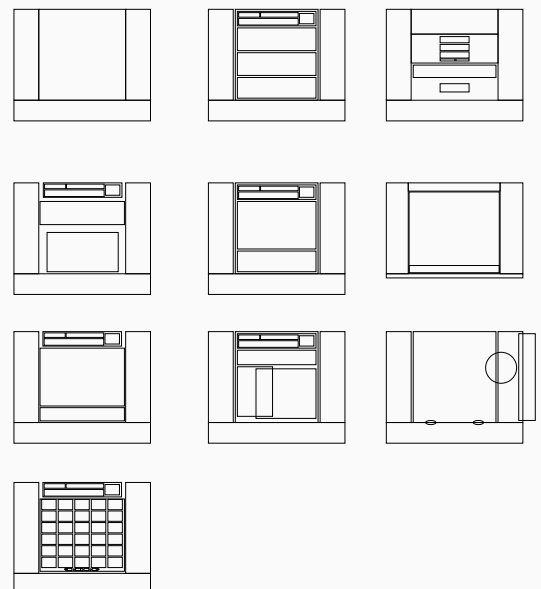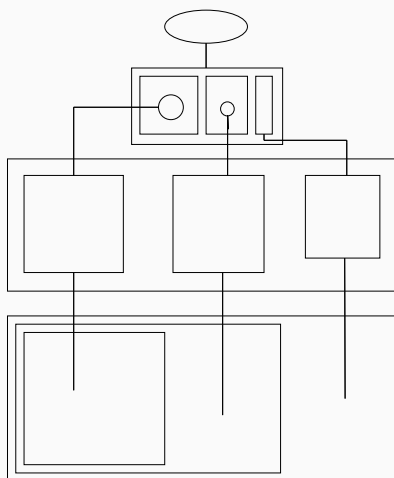
Flowchart

[see precursor/markup]

Brainstory (Thumbnail Sketches)

https://precursorapp.com/document/all-3x3-17592203019508 or are we looking for more detail (yeah)!

Rough Draft

https://precursorapp.com/document/all-3x3-17592203019508

Get color page IMG from old-img from web

Translating the design to the web

SEO

> *Search terms listed as JSON.*

```json
    ["C", 60],
    ["Python", 30],
    ["C++", 30],
    ["Java", 45],
    ["Assebmly", 25],
    ["Systems Administration", 35],
    ["JavaScript", 20],
    ["C#", 10],
    ["Bash", 10],
    ["PHP", 10],
    ["Network Security", 30],
// Roles
    ["Leadership", 30],
    ["Flexible", 35],
    ["Software Engineer", 20],
    // Soft Skills
    ["Innovative", 40],
    ["Initiator", 40],
    ["Goal Oriented", 40],
    ["Team Player", 35],
    ["Dedicated", 25],
    ["Coaching", 40],
    ["Mentoring", 40],
    ["Team Building", 30],
    ["Conflict Management", 25],
    ["Change Management", 10],
    ["Distributed Teams", 30],
    // Architectures
    ["Mobile", 40],
    ["Highly Scalable", 25],
    ["High Availability", 25],
    ["Web", 30],
    ["Enterprise", 30],
```

```
["Embedded", 25],
["Architecture", 34],
["Systems Programming", 40],
// Agile
// Tech Practises
["OOP", 30],
["TDD", 30],
// Tech
["UML", 25],
["HTTP", 10],
["REST", 10],
["SOAP", 10],
["TCP/IP", 10],
["JSON", 10],
["GSM", 10],
["SMS", 10],
["JMS", 15],
["JMX", 10],
["XML", 10],
["XSLT", 10],
["CSS", 10],
["HTML5", 10],
["Bootstrap", 5],
["jQuery", 10],
["JSON", 5],
["Prototyping", 10],
["Node.js", 10],
["Angular", 10],
["Linux", 40],
["AngularJS", 10],
["NodeJS", 10],
["Eclipse", 10],
["Perl", 10],
["Cisco", 30],
["Apache", 10],
```

```
    ["nginx", 10],
    ["Tomcat", 10],
    ["BSD", 10],
    ["UNIX", 10],
    ["Nagios", 10],
    ["MS Exchange/IIS", 10],
    ["Debian", 10],
    ["RedHat", 10],
    ["SuSe", 10],
    ["CentOS", 20],
    ["SVN", 10],
    ["CVS", 10],
    ["CMS", 10], ];
```

Version

0.0.2

End of Site Planning

TEXT THAT FOLLOWS IS FOR EXAMPLE...

> The overriding design goal for Markdown's formatting syntax is to make it as readable as possible. The idea is that a Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions.

Tech

Dillinger uses a number of open source projects to work properly:

- AngularJS - HTML enhanced for web apps!
- Ace Editor - awesome web-based text editor

- [Marked](#) - a super fast port of Markdown to JavaScript
- [Twitter Bootstrap](#) - great UI boilerplate for modern web apps
- [node.js](#) - evented I/O for the backend
- [Express](#) - fast node.js network app framework [@tjholowaychuk](#)
- [Gulp](#) - the streaming build system
- [keymaster.js](#) - awesome keyboard handler lib by [@thomasfuchs](#)
- [jQuery](#) - duh

And of course Dillinger itself is open source with a [public repository](#) on GitHub.

Installation

You need Gulp installed globally:

```
$ npm i -g gulp
```

```
$ git clone [git-repo-url] dillinger
$ cd dillinger
$ npm i -d
$ gulp build --prod
$ NODE_ENV=production node app
```

Plugins

Dillinger is currently extended with the following plugins

- Dropbox
- Github
- Google Drive
- OneDrive

Readmes, how to use them in your own application can be found here:

- [plugins/dropbox/README.md](#)
- [plugins/github/README.md](#)

- plugins/googledrive/README.md
- plugins/onedrive/README.md

Development

Want to contribute? Great!

Dillinger uses Gulp + Webpack for fast developing. Make a change in your file and instantanously see your updates!

Open your favorite Terminal and run these commands.

First Tab:

```
$ node app
```

Second Tab:

```
$ gulp watch
```

(optional) Third:

```
$ karma start
```

Docker, N|Solid and NGINX

More details coming soon.

docker-compose.yml

Change the path for the nginx conf mounting path to your full path, not mine!

Todos

- Write Tests

- Rethink Github Save
- Add Code Comments
- Add Night Mode

License

MIT

Free Software, Hell Yeah!