



Sailing through a sea of code

General⁵

Deployment¹

Web Development⁹

Algorithms¹

GNU/Linux²

Patterns¹

Tools¹



Subscribe in a reader



Subscribe via email

Blog by **Ayman Farhat**. Built with **Django**
and hosted on **Digital Ocean**.

2 years ago

The Hackathon Toolbox - Essential Tools and Practices

Fork me on GitHub

In a couple of days I will be mentoring a team of students at the **NYUAD 2015 hackathon - Building Apps for Social Good in the Arab World**. Since I tend to participate often in development competitions and hackathons, I thought I'd compile a comprehensive list of essential tools needed for any Hackathon as a reference to get back to and share with team mates and friends.

Being effective at a hackathon is all about efficiency. In this article I focus on different tools, tips and practices that will help your team be as efficient as possible in churning out a gorgeous product MVP in 3 or 4 days.

Since a hackathon is not just about writing code, tools related to ideation, design, time management, resource sharing, and presentation are also discussed in detail.

Wireframes

Drawing quick wireframes for describing your UI and UX, is essential during the ideation phase. Prototyping your UI with such a tool will give you more clarity on the product and how it can look like.

Furthermore, a good wireframe can guide your developers through the hackathon in implementing the desired UX while focusing on implementation. Here are some popular tools that can help:

- **Moqups**
- **Basalmiq**
- **Invision**
- **Axure**
- **OmniGraffle**

File sharing

You will definitely run into the need to share files between your team members. Pick one good file sharing platform and make sure to have it set up prior to the hacking period. Ideally, do it at the end of the ideation day before the hacking starts.

- **Dropbox**
- **Google Drive**

Project management

Clarity is power, the more you know exactly what you want to do, the easier it is to execute.

Organizing your team's MVP goals into a simple and clear project plan can make a lot of difference.

You can use a virtual **Scrum board** for organizing, prioritizing and assigning tasks to the team. Here are some good tools to help in that:

- **Trello**
- **Plan.io**
- **Kanbanize**
- **Jira**

If you prefer a simple todo list approach, you can check **Asana** or **Wunderlist**

Image editing

You will probably need to edit images much more than you think. Tweaking that small icon or enhancing the logo at the last minute? Make sure to have a good image editor setup before the hacking starts.

- **Photoshop**
- **Gimp**
- **Illustrator**
- **Inkscape**
- **Sketch**
- **Pixelmator**

Design

In the context of a hackathon, you can't afford spending time on polishing a static photoshop design and then build the UI. Design has to be part of the UI development process, so make sure your team and designer is comfortable with that. Teams with designers who know how to code, have a big advantage here.

Here are some resources that can help in coming up with a decent design implementation:

- **9 Quick Wins for a Halfway Decent Design**
- **Bootstrap Zero** - Free Bootstrap templates themes
- **HTML5 UP**
- **UX Archive patterns** - UX and UI inspiration from top apps.

Fonts

- **Google fonts**
- **Fontface.me** - Collection of gorgeous Arabic fonts available for free
- **Font Awesome Icons** - Collection of easy to use SVG icons available as CSS classes
- **Font Pair** - Helps you in pairing google fonts together

Backgrounds

- **Geopattern** - Generate beautiful SVG patterns for backgrounds
- **Trianglify** - Javascript library for generating colorful triangle meshes that can be used as SVG images and CSS backgrounds
- **Subtle Patterns**

Colors

- **Colors** - A nicer color palette for the web.
- **Adobe Color CC** - Another palette generator

Cool CSS effects

- **Creative Link Effects**
- **Medium scrolling effect**
- **Creative button styles**
- **Icon hover effects**

More UI design inspiration can be found on **Codrops**

Version control

Git is an indispensable CVS that your programming team should be aware of. You can host your project for free on both **Github** and **Bitbucket**, then use whatever service your team is most comfortable with.

Git presents lots of benefits including:

- Merging updates from different team members can be much faster and organized
- Deploying your product online is easier with git especially if you use a PaaS service such as Heroku (more on that below)
- Rolling back and forth between versions, this could be a lifesaver in some cases. Make sure to commit often.

If you don't know how to use Git yet, learn about it right now with **this getting started guide**. In order for your team to use Git effectively, they need to be already familiar with it and have used it in at least one project before the hackathon.

Web application hosting

Hosting your application, if required, is going to take considerable time. Don't even consider architecture and server setup best practices. Try to use services that abstract as much as possible from your deployment process. Such service include:

- **Heroku**
- **Google app engine**
- **Open Shift**

If you choose any of the above mentioned solutions, familiarity with them is important. Pick one and discover it as much as possible before the hackathon. In my opinion, the easiest one to get started with is Heroku, especially if you have some experience with Git.

If you'd still want to go with a vanilla VPS solution, such as **Linode**, **Amazon EC2** or **Digital Ocean**, make sure to have your server setup before the hacking starts. That includes installing all the required software and testing that deployments are working.

Documentation

As we all know, docs are vital for any developer to go on implementing anything. That is why it is very important to gather all your technical docs and organize them under bookmarks that you can access fast.

An even better solution is to download and use them offline. Hackathons tend to be crowded events with everyone using the internet connection, you don't have to be completely dependent on the connection speed to access your docs.

There are some interesting desktop apps that enable you to cache, read and search through your favorite docs. Two great apps are **Dash** (Mac OS only) and **Zeal** (Cross-platform).

Code libraries and frameworks

Here is a list of the most popular libraries that you might want to include in your project to speed things up. Most of the recommendations are inclined towards web-based technologies, they can also be used within the context of a cross-platform HTML5 app running on top of something like Cordova.

Front-end web UI and Responsive Grid systems

- **Bootstrap 3**
- **Foundation**
- **Materialize** - Provides a full implementation of the Google material design, including a responsive grid.
- **Chopstick**

Less fully-featured frameworks include:

- **Skeleton**
- **Neat**

Core Javascript libraries

- **jQuery**
- **jQuery UI**
- **KnockoutJS**

Javascript frameworks

If you're not already familiar with a Javascript framework, it might make a lot of sense to get into one. If you're experienced with any of the below mentioned frameworks, you'll be able to prototype your client-side functionality much faster than doing it in vanilla JS or utilizing simple libraries like jQuery.

- **AngularJS**
- **EmberJS**
- **Meteor**

Don't plunge into a full-fledged framework unless you have around a month or two to get to experience it. If your hackathon is next week and you don't know any frameworks, you're better off exercising on libraries you already know and explore small libraries and plugins that can help for different use cases.

UI Control Libraries

- **Selectize**
- **Drop.js** - The fast and capable dropdown library
- **Scroll Reveal** - Easily reveal elements as they enter the viewport
- **NProgress.js** - Slim progress bars like on YouTube and Medium.
- **Hover** - Awesome CSS3 animations on mouse hover
- **jQuery Raty** - Star Rating Plugin
- **xEditable** - In-place editing with Twitter Bootstrap, jQuery UI or pure jQuery
- **Alertify** - Sweet looking alerts and browser dialogs
- **UI Box** - Curated HTML, CSS, JS UI Component Library
- **Masonry** - Cascading grid layout library

Charting libraries

- **Google Charts**
- **High Charts**
- **CanvasJS**
- **Chartist.js**
- **FlotJS** Lightweight and comes as a jQuery plugin so integrates really well if you're using jQuery
- **AMCharts**
- **D3JS** - This is a low level (relatively) library with a significant learning curve, make sure you invest considerable time on it before using it.
- **Angular-chart** Mainly useful if you're already using Angular. What's really cool about it is that it allows you to build reactive charts.
- **Ember Charts** If you're using Ember already as a framework of choice

Mapping libraries

- **Google maps API**
- **Mapbox**
- **Open Street Map**
- **LeafletJS**
- **Open Layers**
- **Data Maps**
- **Bing Maps API**
- **Kartograph**

Backend as a service

I haven't included any backend frameworks or technologies above on purpose. You probably won't need to implement a custom backend within the scope of a hackathon.

Don't allocate precious time building a perfect backend system nor architecting the perfect database. Since your prototype will be simple, it's more important to focus on the business logic, front-end UI and UX in order to make things happen in a short period of time. You can sacrifice building a perfect custom backend and API by consuming easier BaaS (Backend as a service) solutions.

A good BaaS, will enable your team to:

- Have a CRUD enabled database, abstracted via a nicely documented API in no time
- Have a User management and authentication system baked into the backend
- Enable social media auth integration
- Get a push notifications system up and running
- Sync and store application and user data instantly across multiple platforms
- Save time on provisioning or managing server infrastructure

All you have to do is consume the API from the application in order to do the common data persistence tasks.

Here is list of popular solutions that meet the above criteria and more:

- **Firebase**
- **Parse**

- **Backendless**

Tools and setup

Make sure to setup your tools, code repository and frameworks before the official hacking starts. That way you'll be able to focus on implementation as soon as the hacking starts. Don't waste precious time for setting up environments or frameworks.

The Presentation

It would be sad to hustle for three days building something amazing and end screwing up the demo presentation. It happens a lot, a well polished and rehearsed presentation is essential to spend time on. Here are some tools and resources that can help:

- **Google Slides**
- **Prezi**
- **Keynote**
- **SlideRocket**
- **RevealJS** - A framework for easily creating beautiful presentations using HTML.
- **How to present a successful hackathon demo**
- **Top 5 tips for presenting at hackathons**
- **Top 10 ways to improve hackathon presentations**

Have fun!

First and foremost, a hackathon about experimenting, meeting interesting people, learning something new and getting the opportunity to push your limits. Yes its a competition, but you don't have to stress yourself too much about winning. Make sure you have a good time, put your best effort and learn from your experience.

That was my list for important tools to keep an eye on for your next hackathon. Do you think I have missed anything? Please let me know in the comments!

[mvp](#) [tricks](#) [practices](#) [tools](#) [hackathon](#) [tips](#)

↑ **Back to top**