hackNEHS Hacker Info | Start Guide

Hello, and welcome to hackNEHS!

We've put together this list of resources to get you started, but there's also a lot more out there on the interwebz. So feel free to explore other options beyond these, and remember that Google and StackOverflow are going to be your best friends today!

Without further ado, let's get started.

1. Brainstorming a project

So it's Saturday morning and time to hack. Perhaps your team quickly agrees on an idea -- if so, fantastic! But more often, coming up with (and agreeing on) a good idea can be tough. Here are a few things to keep in mind as you choose your project.

1. **Whatever you present tonight is probably going to be pretty different from what you have in mind right now.** There are always going to be difficulties and new ideas that spring up along the way, causing you you change direction. That isn't a bad thing at all! Just remember to go with the flow. So if you can't come up with the greatest idea right now, that's fine -- you have to start somewhere.

2. **Make sure that the scope of your project is manageable.** Don't try to create Google in a day. Remember that you will be presenting a prototype of your idea tonight, not a finished product. Also, a general rule of thumb is that you can usually pick up one or maybe two new tools or frameworks in a day, but three or more is probably too much.

3. **Choose a fun project that you all want to work on!** This should go without stating but we think it's worth stressing. It sucks to realize halfway through the day that your project isn't very interesting to you, and you will probably lose motivation after that. The goal is for everyone to finish a project that they are proud of, and have fun in the process!

As always, if you are stuck or want some advice, you should flag down a hackNEHS mentor and ask for help.

2. Figure out what tools you will need

Note: this is not an exhaustive list Also, while we have done our best to summarize them, the differences between various options can be very dependent on your application, so we strongly recommend that you do a little bit of research of your own before choosing.

Also, if you already know what you're going to use, then you can probably skip this section.

Version control

First thing's first. GitHub is a must when working on a group project. In a nutshell, it is a version control system that is the gold standard for developers across the board. If you're new and need help setting up Git, see their Bootcamp for help. There are also various useful resources for learning Git and GitHub available. Set this up first.

Text editors

Everybody has their own personal favorite for this one, but we still have recommendations. Atom and Sublime are hands-down fantastic. There is also Notepad++, which is a bit less powerful but still pretty versatile. You *can* use the native text editors that come with your OS (Notepad, TextEdit, etc.) if you really feel like it, but believe us when we say that they suck for writing code.

Web development

The basics

So you want to build a website? If you've never worked with HTML or CSS before, then you should check out Matt Macarty's workshop. (The files from the workshop can be found here.) For JavaScript, check out the LIberty Mutual workshop.

W3schools is a great reference for HTML/CSS/JS If you want pretty CSS with minimal work, check out the packages Bootstrap and Skeleton. There are also some fancy JavaScript tools that make building a responsive frontend much easier, such as jQuery, ReactJS, and AngularJS. (W3schools also has good resources for Bootstrap, jQuery, and AngularJS.)

Get Sh*t Done (by one of our sponsors, Creative Tim) is a collection of beautiful HTML/CSS/JS elements, which you get free access to today! Here is the description from them:
The 'Get Sh*t Done' kit is built on top of Bootstrap 3, so you can safely use it on your existing or new Bootstrap project. No code from Bootstrap 3 will be changed, so you don't have to worry about undesired effects to your work. We provide all the necessary CSS resources. To get going, just include 'css/get-shit-done.css' in your HTML template. Your project will get the new look. The easiest way to start is to use our start-up template where all the files are already included and ready to use.

For the Get Sh*t Done Pro use this link and the coupon code: hacknehs-voucher.
For Material Kit Pro use this link: with the coupon code: hacknehs-voucher-mk.

Frameworks

There are various frameworks that you can use if you need to go beyond plain HTML/CSS/JS, each with their own specific strengths and weaknesses. However, which one you go with is going to depend most on what you are already familiar with.

If you are familiar with Python, then Flask, Webapp2, and Django are some good options. Flask is probably the most lightweight and the quickest to pick up if you've never worked with these kinds of frameworks before. Webapp2 is also pretty good because it is supported by Google App Engine (see the Hosting section) for free. Django is definitely more heavy-duty -- if you are familiar with it, then it's a fantastic option for quick development, but we don't recommend it if you haven't worked with it before.

Other common options include NodeJS, Meteor, and Ruby. You might find Node confusing and hard to understand if you aren't somewhat familiar with it already. Meteor is pretty powerful yet also not too hard to pick up. Ruby is a really good option if you're familiar with it, but may be tougher than some of the others if you haven't worked with it before.

In general, don't reinvent the wheel -- there are lots of boilerplates out there (such as these) that do some grunt work for you. Feel free to use code that is open-sourced -- that's the whole point of it.

Deployment

We're not going to try to beat this.

We will add, however, that Google App Engine is a good option for certain frameworks (e.g. webapp2, which it supports natively). However, it isn't always free, so make sure you check it out completely first.

App development

Android

MIT App Inventor is a good place to start if you're *completely* new. If you've done some Java before then check out the Android Developer Getting Started Guide. If you want an IDE, Android Studio and Eclipse are pretty good.

iOS

If you aren't already aware of this, you can't easily do iOS development without a Mac. (Bummer, we know, but we can't change that.) Anyhow, Swift is a good option. You can also use Xcode. Here's a neat little start guide for that.

Game development

There are lots of options out there. If you're new, Processing is fantastic, and there are lots of good tutorials on the Internet. GameMaker is another easy-to-learn option if you have little experience. If you know some coding, Unity is very good because it is far more powerful.

3. General tips

1. **Don't reinvent the wheel!** Use stuff that's already out there, whether that means tutorials, boilerplate code, or anything open-sourced that might be helpful. Just be careful to not infringe on any copyrights.

2. **Have fun!** Not to be cliché, but this is actually important.
3. **Manage your time carefully.** You only have 9 hours in total to hack, and even less if you go to workshops, so don't get hung up making decisions, creating graphics, etc. Your project is called a hack for a reason, so it's perfectly fine if it isn't all professional-looking.
4. **Google things!** Use the internet! It's there to help you.
5. **Make full use of the mentors.** They are a great resource and they're volunteering their time to assist you, so make sure to flag them down and ask for help when you need it.