

A woman in a white shirt and dark pants stands at the head of a long wooden conference table, pointing at a whiteboard. Three people are seated around the table, looking towards her. The room has large windows and modern lighting.

**MIT GSL**

**2018**

week 4 | Friday

App Development Checklist

# Agenda

- Questions on anything we've covered before we start programming?
- Designs
- App Specification
- Checklist before programming
- Hackathon
  - Designs/Mockups
  - Programming and setting up application
  - Hardware

# App Development Checklist

A close-up, shallow depth-of-field photograph of a person's hands typing on a silver laptop keyboard. The person has dark skin and is wearing dark nail polish. The background is heavily blurred, showing what appears to be a desk with a laptop, a glass of orange juice, and some papers. The overall lighting is warm and soft.

# Checklist

- Understanding of fundamental problem
- List of Features, prioritized
- Design of Application (not web page mockups)
  - Read: <https://guide.meteor.com/structure.html>
- ***meteor create app-name*** or clone boilerplate code
- Add react packages and version control (Git)
- Add Heroku to app
- Follow style guidelines for javascript(ESLint) and meteor
- Lookup [guide.meteor.com](https://guide.meteor.com) for specific help with different aspects of meteor
- Save through git often

# Setting up Meteor Application - Version control

- <http://carlofontanos.com/deploying-meteor-app-to-heroku/>
- Good set of instructions on how to deploy your apps on heroku

# Boilerplate

- Your choice in starting point
- Read the readme section thoroughly before cloning
- Make sure you understand the pieces you do not need and those that you do
- <https://github.com/AdamBrodzinski/react-ive-meteor>
- <https://github.com/AdamBrodzinski/meteor-react-boilerplate>

# Code Style

- Read this before starting your applications : <https://guide.meteor.com/code-style.html>
- Install ESLint to help with error checking and consistency of your code

# Application Structure

- Read this before to understand the structure of Meteor applications :  
<https://guide.meteor.com/structure.html>



# Flow routing

- <https://guide.meteor.com/routing.html>
- `meteor add kadora:flow-router`
- <https://dev.to/damcosset/meteor-react-and-flowrouter-quick-setup-6g5>

# Flow routing

```
import React from 'react'  
import { FlowRouter } from 'meteor/kadira:flow-router'  
import { mount } from 'react-mounter'
```

```
import App from '/imports/ui/App'  
import HomePage from '/imports/ui/HomePage'  
import AboutPage from '/imports/ui/AboutPage'
```

```
FlowRouter.route('/', {  
  name: 'Home',  
  action(){  
    mount( App, {  
      content: <HomePage />  
    })})})})
```

```
.....  
FlowRouter.route('/about', {  
  name: 'About',  
  action(){  
    mount( App, {  
      content: <AboutPage />  
    })  
  }  
})
```

A close-up photograph of a person's hands typing on a silver laptop keyboard. The person has dark skin and is wearing dark nail polish. The laptop is open, and the keyboard is in focus. In the background, there is a blurred scene of a hackathon, with another person's hand visible on the right and a laptop screen on the left. The word "HACKATHON" is overlaid in white, bold, sans-serif capital letters in the upper center of the image.

HACKATHON

# Judging criteria

- Design
  - How many designs do you have?
  - How detailed and clear are your designs?
- Hardware
  - Were you able to integrate hardware with a demo app?
  - How many things can you get your arduino to do?
- App Development
  - Quality of your infrastructure based on your problem
  - How many features were you able to implement?

# Hackathon Schedule

- 2:00pm - 2:15pm - **Introduction & Question**
- 2:15pm - 4pm - **Work Time**
- 4:00pm - 5:00pm - **Foosball & snacks**
- 5:00pm - 7:00pm - **Work Time**
- 7:00pm - 8:00pm - **Dinner & Games**
- 8:00pm - 9:30pm - **Work Time**
- 9:30pm - 10:00pm - **Judging**