Meteor JS Framework

Reactive framework

- Meteor JS framework
- Templates
- Events
- Sessions
- Publish & Subscribe
- Accounts

Meteor JS

- It is a **full-stack JavaScript platform** for developing modern web and mobile applications.
- It includes a key set of technologies for building connected-client reactive applications, a build tool, and a curated set of packages from the Node.js and general JavaScript community.
- Meteor allows you to develop in **one language**, JavaScript, in all environments: application server, web browser, and mobile device.
- Meteor uses **data on the wire**, meaning the server sends data, not HTML, and the client renders it.
- Meteor **embraces the ecosystem**, bringing the best parts of the extremely active JavaScript community to you in a careful and considered way.
- Meteor provides **full stack reactivity**, allowing your UI to seamlessly reflect the true state of the world with minimal development effort.

Meteor JS

- Meteor.js is an open-source platform built on Node and MongoDB. It's not just a framework, it's more than that.
- Is it comparable to Angular?
  - Angular handles just the frontend parts of application.
  - Meteor is **more of a platform** than Angular.
  - It is both server and client-side code and handles a lot of the parts of applications that we need to create manually in Angular.

Features:

- **Web and Mobile** – Meteor offers a platform for developing Web, Android and IOS apps.
- **Universal Apps** – The same code for web browsers and mobile devices.
- **Packages** – Huge number of packages that are easy to install and use.
- **Meteor Galaxy** – Cloud service for Meteor app deployment.

Advantages

- Developers only need JavaScript for server and client side development.
- Coding is very simple and beginner friendly.
- Meteor apps are real time by default.
- Official and community packages are huge time saver.

Limitations

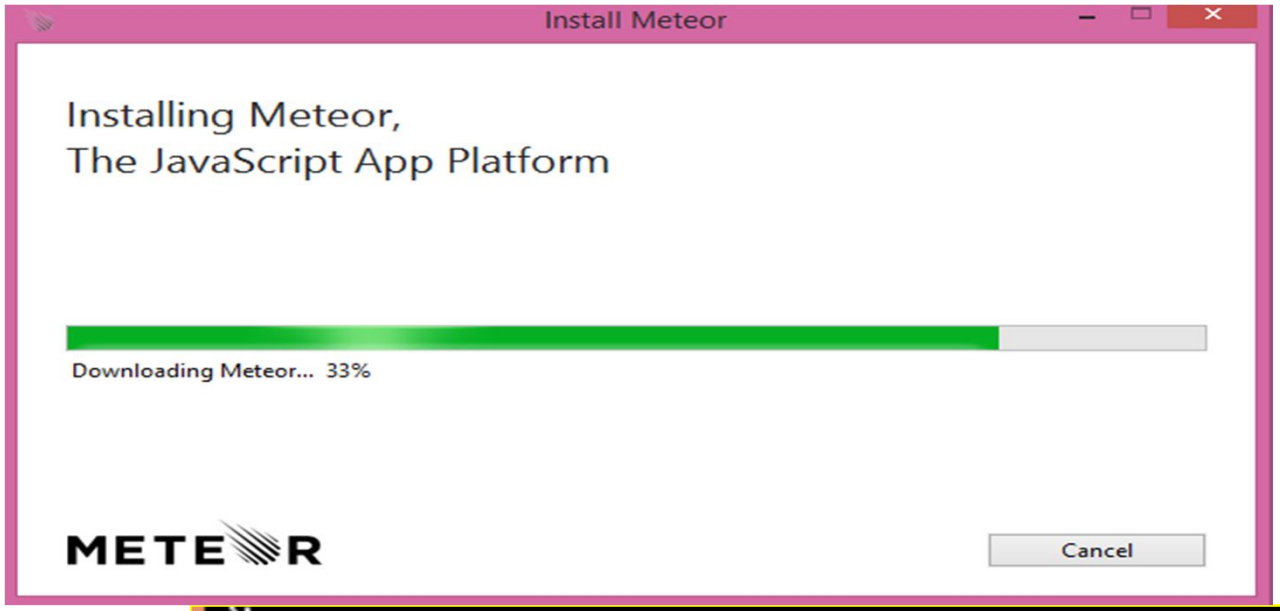
- Meteor isn't very suitable for large and complex applications.
- There is a lot of magic going on when working with Meteor, so developers might find themselves limited in some way.

Important Things to Know

- **Real time built into its core**: Meteor handles all of the real-time components so that as soon as you update something in your database, that change is made to all other connected users. Out of the box and very easy to use.
- **Full Stack Reactivity**. In Meteor, real time is the default. All layers, from database to template, update themselves automatically when necessary.
- **Built-in build system** Meteor believes that we spend too much time packaging our applications together and getting all the dependencies working together nicely. You don't have to fiddle with a Gulp configuration anymore. Meteor handles it all for you out of the box.
- Great for **single page apps and mobile**
- **Packages** are handled through Meteor's pacakage management site: atmosphere.js (can also use npm and cordova packages)
- Connecting to external services and APIs is possible using ddp: DDP, the Distributed Data Protocol, is a simple protocol for fetching structured data from a server, and receiving live updates when that data changes. We won't deal with this in our application today, but it's good to know.

Install Meteor

- Download the official meteor installer from this page
- https://www.meteor.com/install OR
- download for windows: https://install.meteor.com/windows



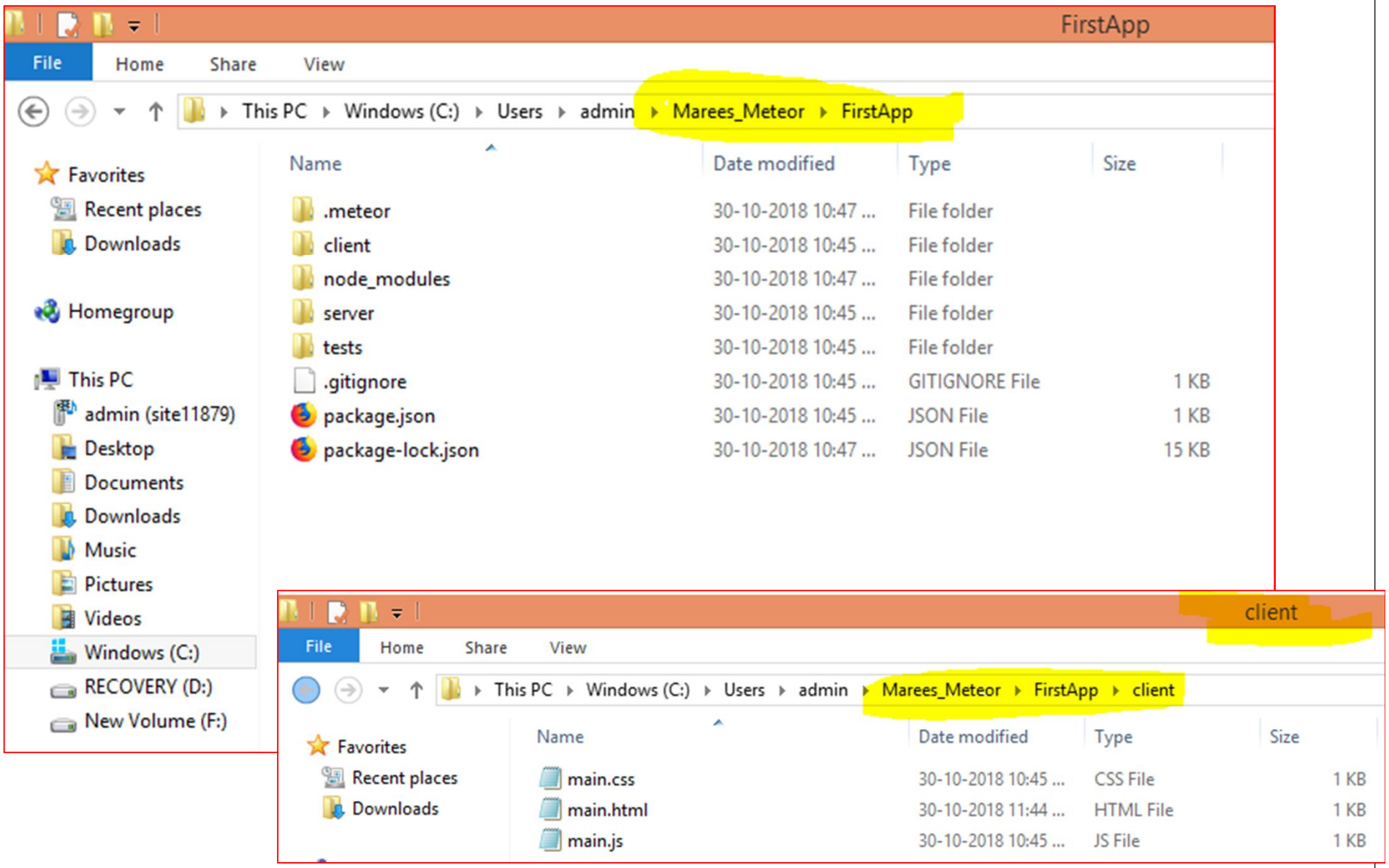

```
C:\Users\admin\Marees_Meteor>meteor
run: You're not in a Meteor project directory.

To create a new Meteor project:
  meteor create <project name>
For example:
  meteor create myapp

For more help, see 'meteor --help'.
```

```
C:\Users\admin>meteor --version
Meteor 1.8

C:\Users\admin>
```

Step 1 - Create the App/project

```
C:\Users\admin\Marees_Meteor>meteor create FirstApp
Created a new Meteor app in 'FirstApp'.

To run your new app:
  cd FirstApp
  meteor

If you are new to Meteor, try some of the learning resources here:
  https://www.meteor.com/tutorials

To start with a different app template, try one of the following:

  meteor create --bare    # to create an empty app

  meteor create --minimal # to create an app with as few Meteor packages as poss
ible
  meteor create --full    # to create a more complete scaffolded app
  meteor create --react   # to create a basic React-based app


C:\Users\admin\Marees_Meteor>
```

Check it folder



Step 2 - Run the App

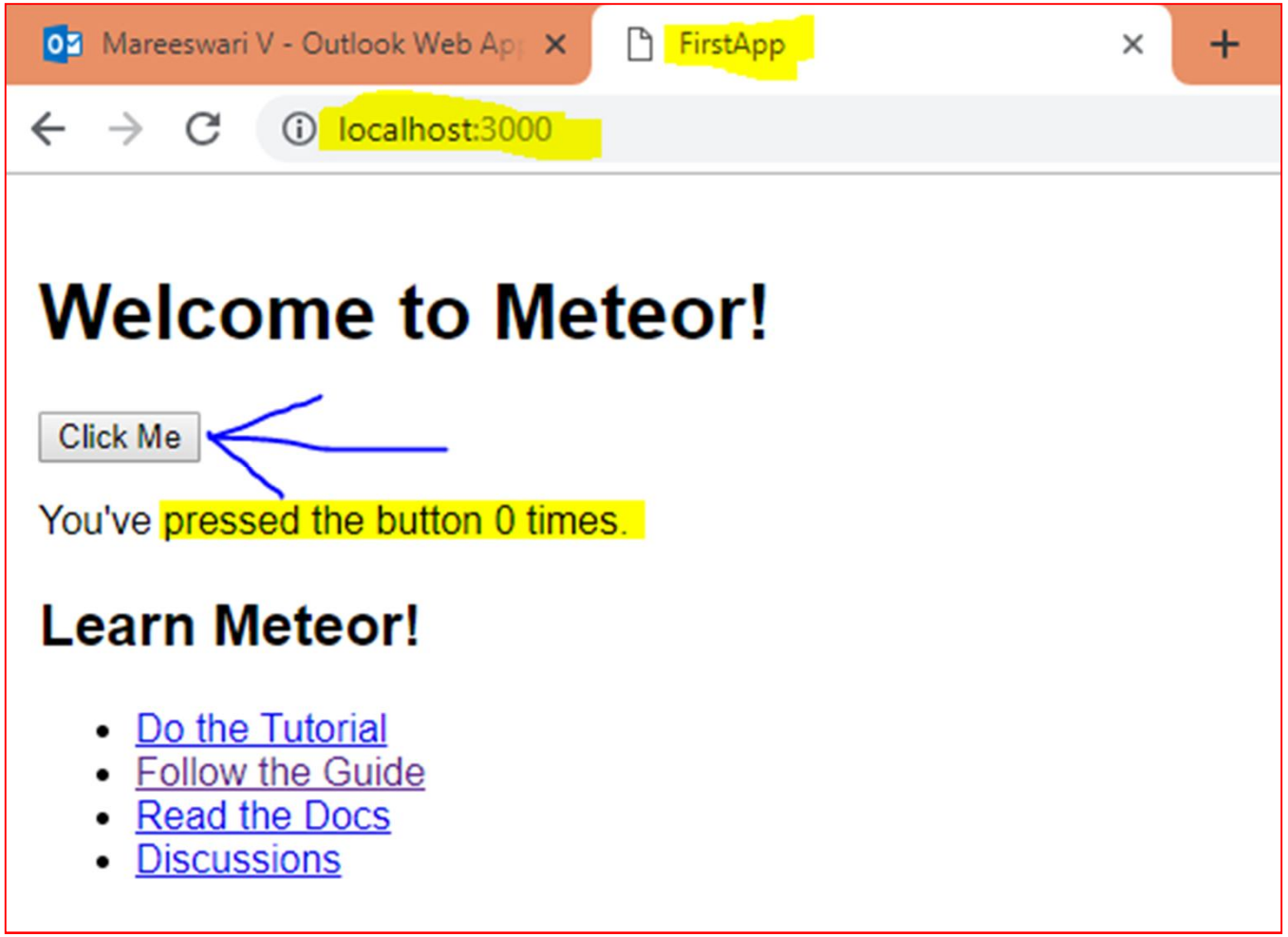
- Once you create a app, just cd into that project and run the application with:

```
C:\Users\admin\Marees_Meteor>cd FirstApp

C:\Users\admin\Marees_Meteor\FirstApp>meteor
[[[[[ C:\Users\admin\Marees_Meteor\FirstApp ]]]]]

=> Started proxy.
=> Started MongoDB.
=> Started your app.

=> App running at: http://localhost:3000/
   Type Control-C twice to stop.
```

- This will **grab the necessary packages**, **bundle all the css and js**, **start your application using a Node server**, and make it viewable in your browser.

Step 3 - Verify the Result

- open the **http://localhost:3000/** address



```
| .meteor
| client/                       // all the code for our client and browser
    |----- components/          // we'll be creating components for our application
      |--- poll-form.css
      |--- poll-form.html
      |--- poll-form.js
      |--- poll.css
      |--- poll.html
      |--- poll.js
    |----- app.body.html        // layout for our entire app
    |----- app.head.html        // document head for entire app
    |----- app.js                   // the overall js for our layout
    |----- app.css                   // the overall css for our layout
| collections/              // here we'll store our mongo models
    |----- polls.js         // defining our mongo collection
| server/                       // code for our server
    |----- bootstrap.js // adding sample data on app startup
```

meteorApp creation