React

* Responsible for the client side – browser side
* User interface
* Based on Javascript – it’s a javascript library
* Cannot be used for storage – on browser side

NodeJS/ExpressJS

* Backend, server side
* Web application that runs on a dedicated machine/server somewhere on the internet
* These two pieces are used to write Javascript code that runs on a server, detached from client
* More performance intensive side
* Communicates with client side via HTTP request/responses (Ajax)
* Responses are handled in the client without reloading the page on the client.
* Handled through JSON format (used to exchange text data, numeric, structured data)

MongoDB

* Stored on Database Server
* Used for persistent data storage (not files)
* Node/Express talks to the database server (mongoDB)
* We do not send requests from the client side to the db because these queries would include our db credentials

A diagram of a server

Description automatically generated

React (Front End Client)

* Here, we build a React SPA (Single page application)
* In charge of re-rendering everything in the browser
* Routes (with react-router-dom)
* In charge of Rendering react components based on a path the user enters (through a router)
* State Management (Hooks, Redux)
* In charge of re-rendering the parts of the screen that needs to change
* Components + Styling (CSS)

A diagram of a company

Description automatically generated

The Backend (Server)

* Decoupled Ends
* Backend == API
* Two kinds of APIs you can build – REST API or GraphQL API
* Different in terms of how the requests are sent to the API should be formatted
* REST by far the most common and used form (Combination of different URLs or paths [get,post,patch,delete]
* GraphQL – one URL and one HTTP verb, typically a POST request… one endpoint that accepts query commands
* The body of the request contains a query expression describing the operation you want to perform.
* In both cases, you talk to the Node/Express app and that app will then do something based on the action that is triggered

A diagram of a server

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a web site

Description automatically generated

A diagram of a graph

Description automatically generatedA diagram of a graph

Description automatically generated

REST API is stateless and decoupled from any front end, meaning the API can be used with your SPA and any client can talk to it. (if you later build a mobile application, can talk to the same API)

A screenshot of a graphql

Description automatically generated