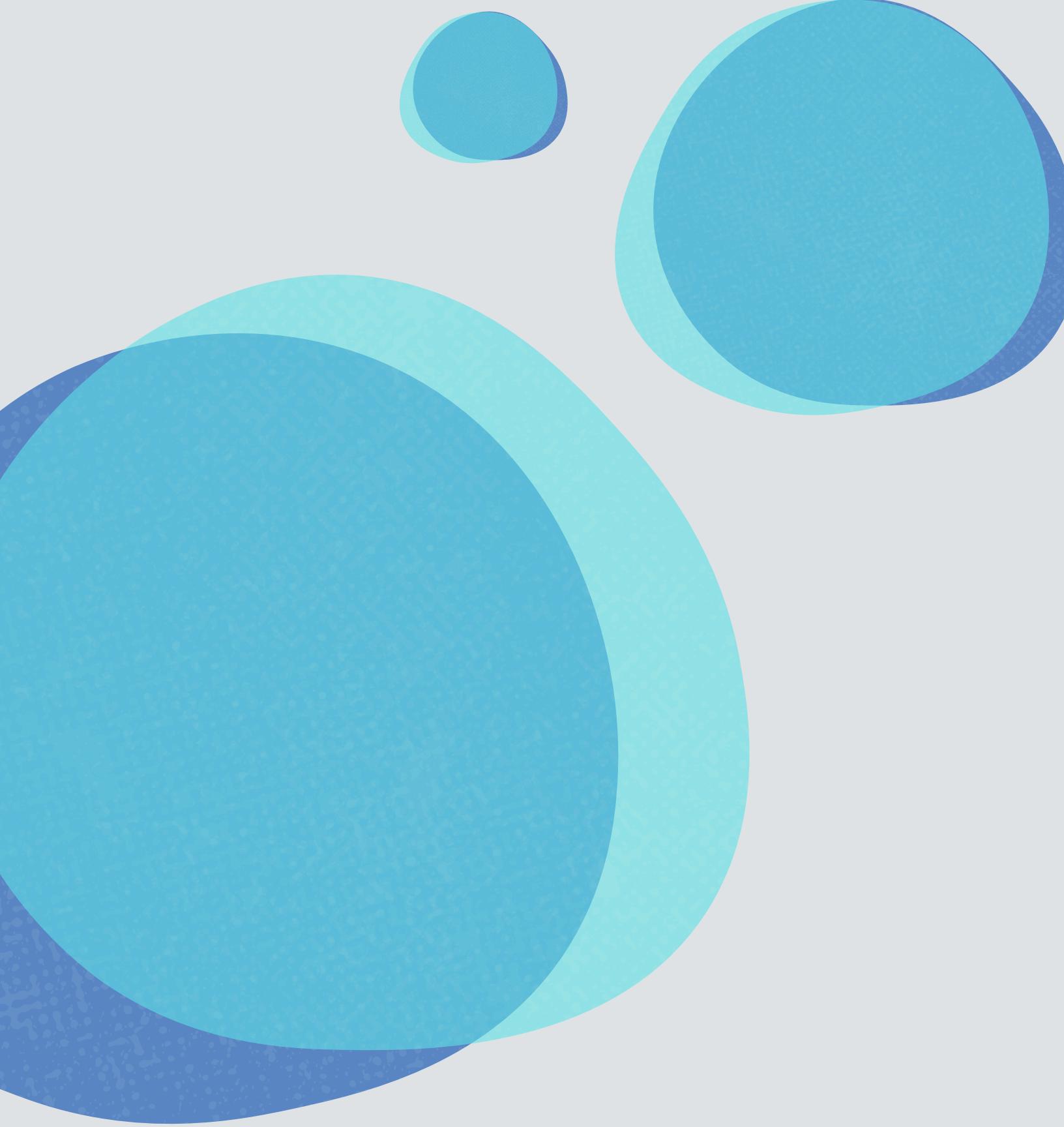


MVC: MODEL - VIEW- CONTROLLER



HISTORY OF MVC

- MVC pattern was first discussed in 1979 by Trygve Reenskaug.
- MVC pattern was first used in 1987 in the Smalltalk programming language who originally wanted it to be called Model-View-Editor.
- By the Late 1990s, the pattern was widely adopted for web applications.

WHAT IS MVC?

Model-View-Controller is a popular software pattern. Breaks up the logic of your application into three different components.

MODEL-VIEW-CONTROLLER

Model

Responsible for the data logic and structure. eg Mongoose-models and schemas

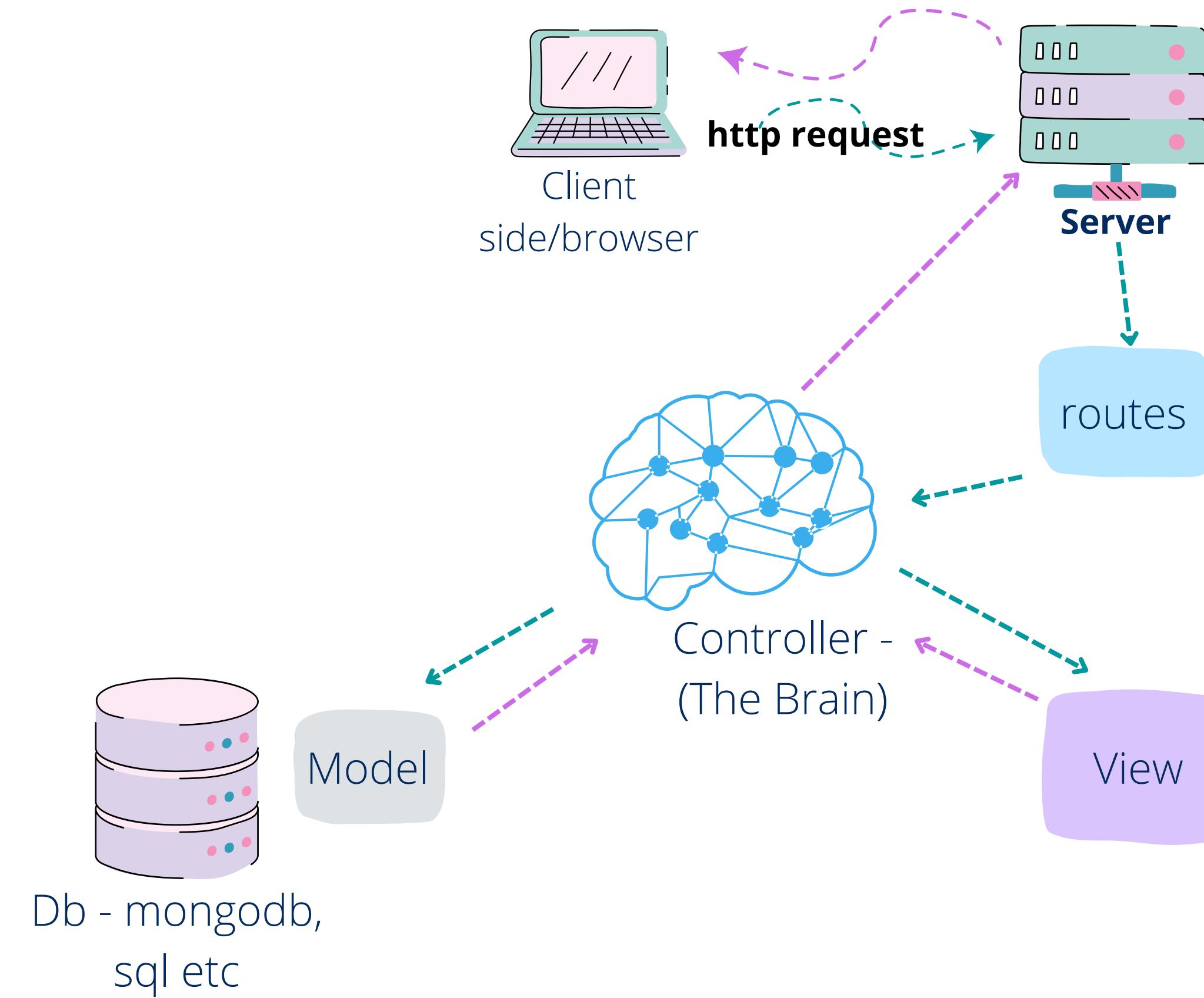
View

This is what the end user sees and interacts with in the application eg EJS, HTML, PUG

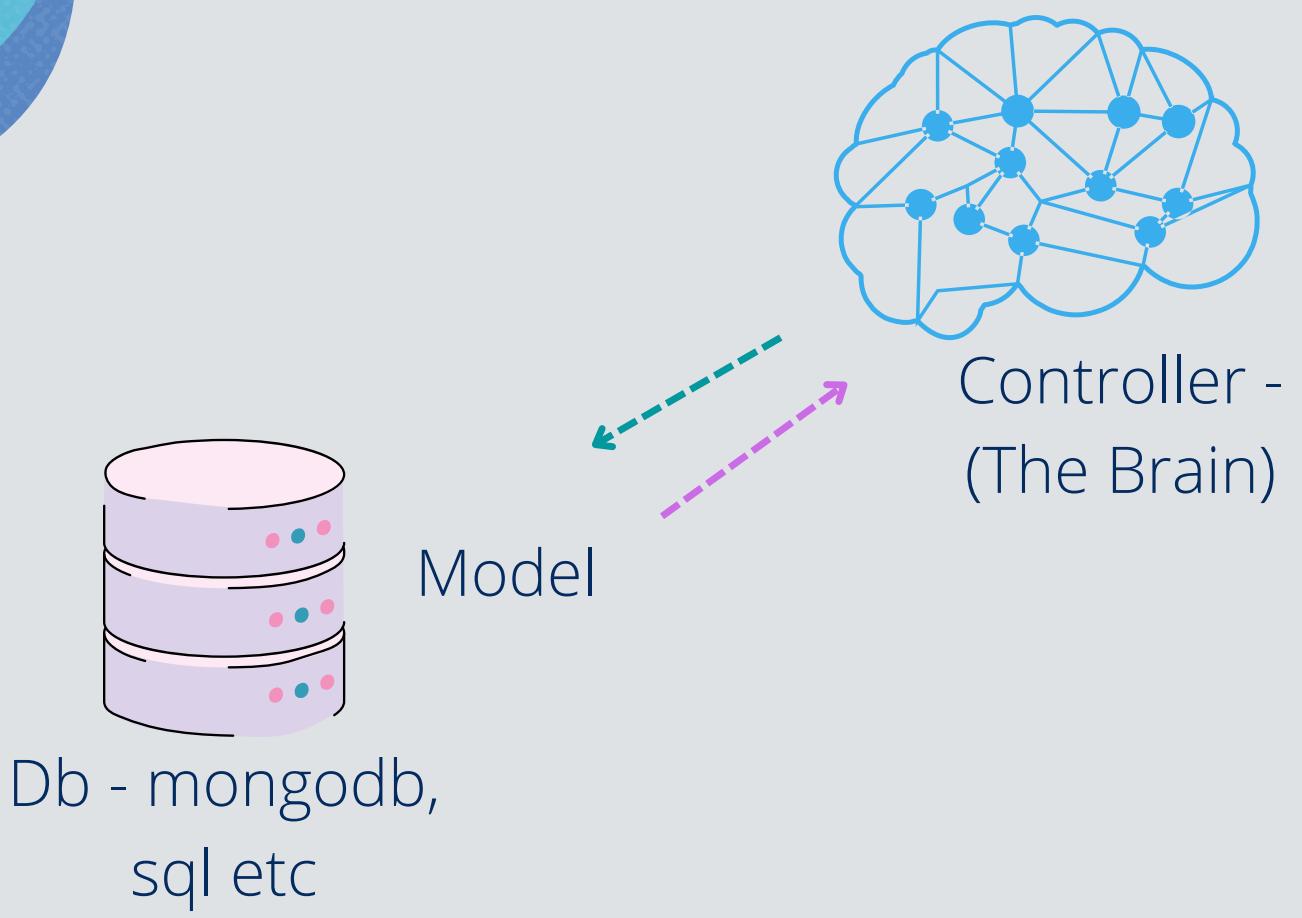
Controller

This is the control-centre (brains) or middleman behind the application and communicates with the Model and View.

MVC PATTERN FLOW



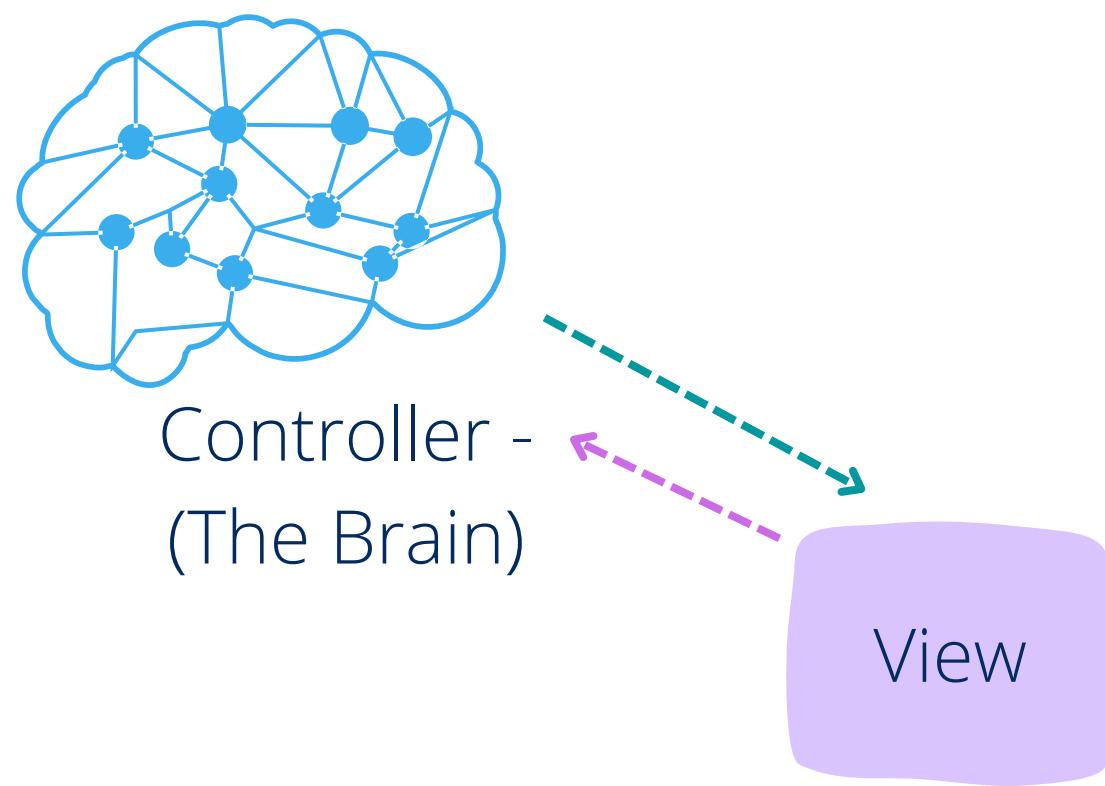
Model



- Mongoose is a wrapper that sits on top of Mongodbs and can be used to make things easier. It also sits very well with MVC.
- In Mongoose the Model is a wrapper that sits on top of the schema.
- The Mongoose schema defines the structure of the document, default values, validators, etc., whereas a Mongoose model provides an interface to the database for creating, querying, updating, deleting records, etc.
- The model's job is to simply manage the data and talks directly to the controller and the database.

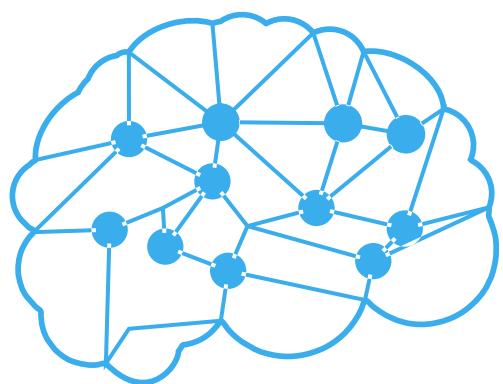
VIEW

- The View component is responsible for all of the visual html aspects of the application.
- Language templates such as EJS, React and Pug are stored in the view folder and talk directly to the controller.
- The controller then passes data to the view template from the model and vice versa
- The view renders the data before passing back to the controller which in turn sends html back to the server generating a response to the browser.



ROUTES

- But wait.... before we head on to the controller, there's an extra bit we need to talk about. Let's talk about routes. In order to locate the controller, we need to set up our routes. we will use express router as an example.



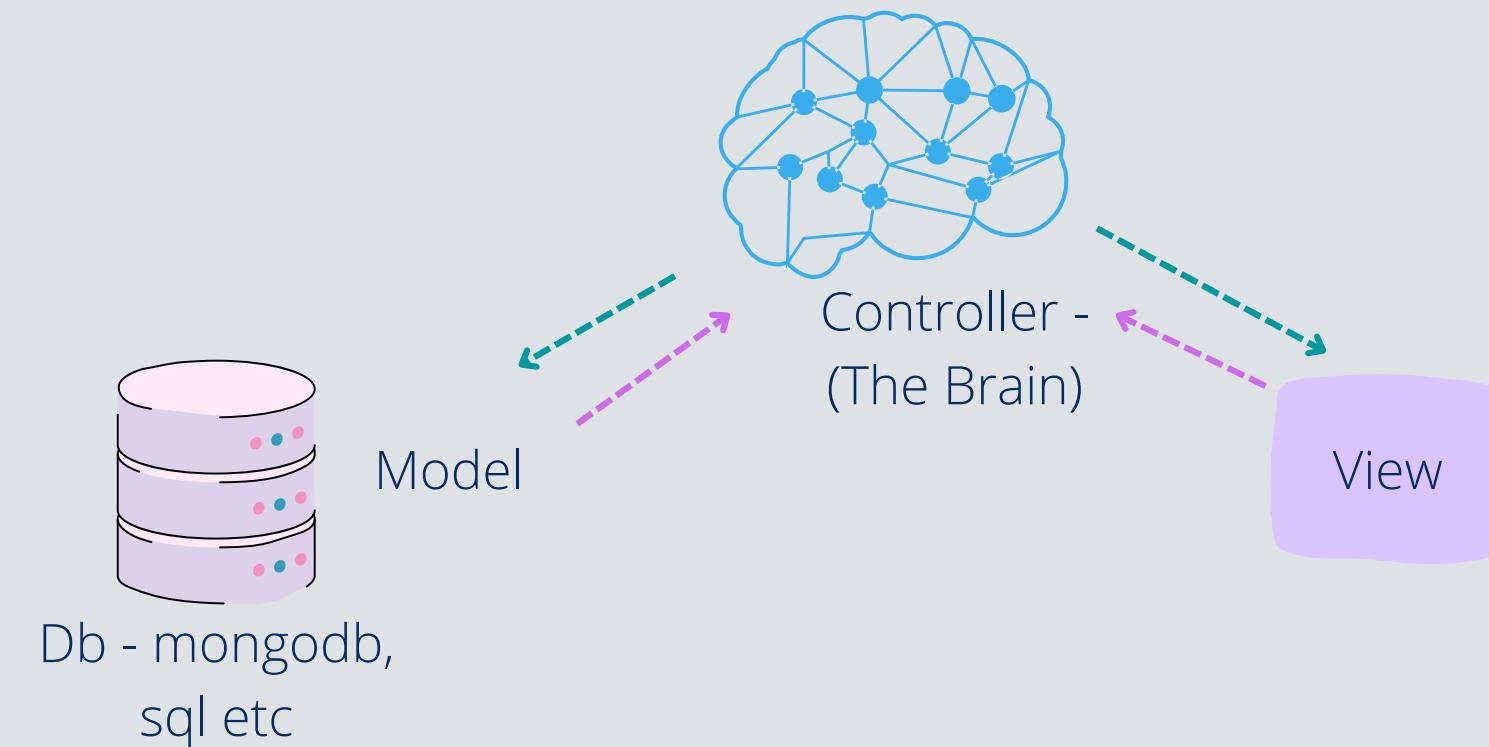
routes



- Express router sets the route e.g.
`router.get('/todos/:id', todoController.exploreTodosById);`
- It determines which controller to use based on the http request method which uses a route param and a callback method or function to locate the controller.
- It routes directly to the controller and nothing else.

CONTROLLER

- The Controller is 'The Brain', 'The control centre'.
- It is the middleman, dealing with the complex logic and does all coordination between the model and view.
- As an example if there was a change made by the view to any part of the db eg a post, this would be handled and processed by the controller who would then send it off to the model.
- In the same example. The model then passes the relevant info back to controller who sends it on to the view to render.



SUPPORTING INFORMATION

The Model View Controller Pattern – MVC Architecture and Frameworks Explained

<https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>

MVC Architecture – What is a Model View Controller Framework?

<https://www.freecodecamp.org/news/mvc-architecture-what-is-a-model-view-controller-framework/#:~:text=The%20MVC%20pattern%20was%20first,the%20programming%20language%20Small%20Talk.>

Express Tutorial Part 4: Routes and controllers

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/routes