

the Master Course

{CODENATION}

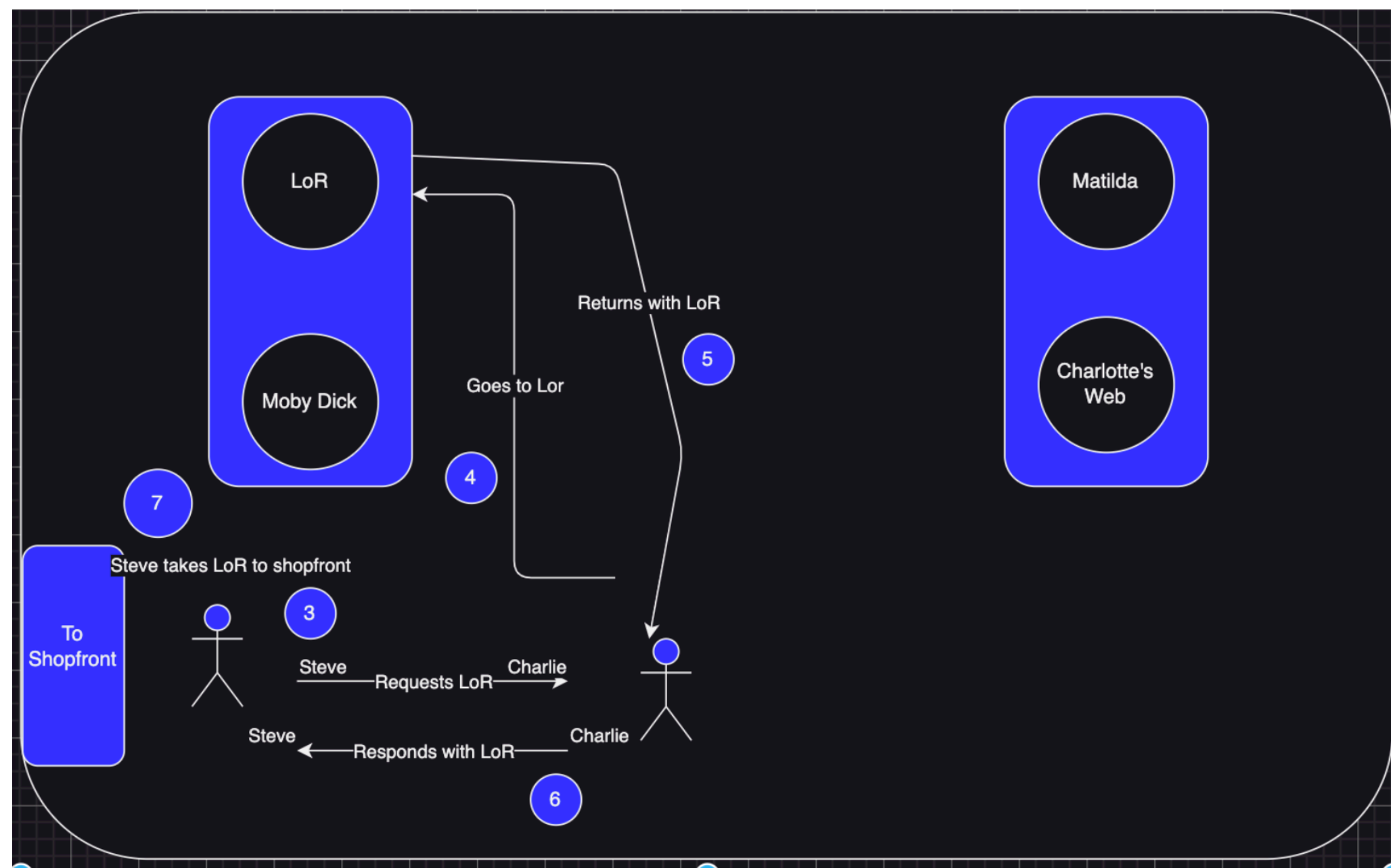
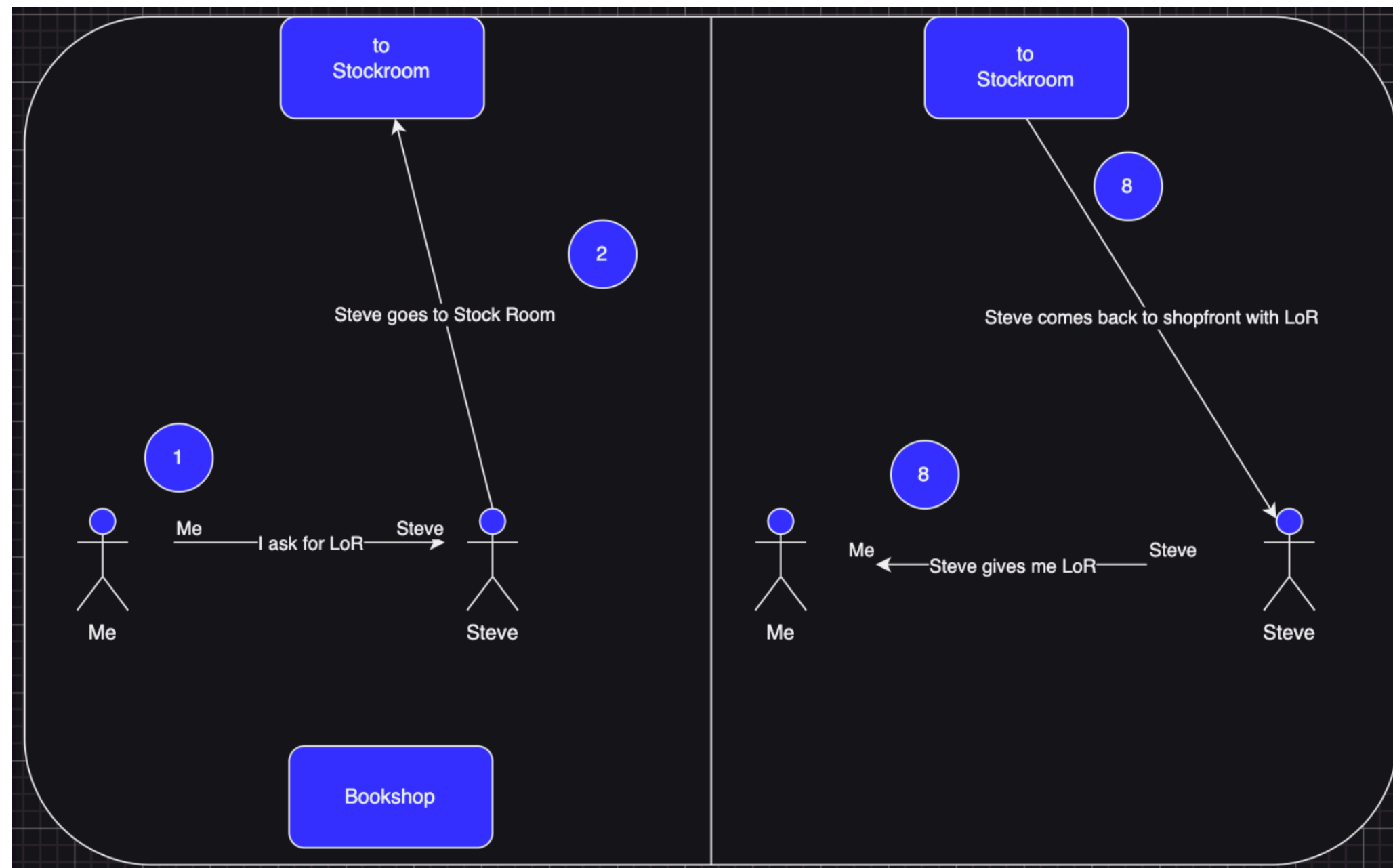
What are Routes?

{ C O D E N A T I O N }

Routes

Imagine your visiting a bookshop...

You ask Steve the shopkeeper for a copy of
Lord of the Rings



I visit a bookshop. I meet Steve, the shopkeeper in the shopfront.
Charlie is in the stockroom, dealing with things in there.

- 1) I ask Steve the shopkeeper for a copy of LoR (I interact with a web page e.g. button click)
- 2) Steve the shopkeeper goes to the stockroom (A request is made by the client to the server)
- 3) Steves asks Charlie to get a copy of LoR (the request reaches the server)
- 4) Charlie walks to the shelf where LoR is kept (an API route is followed to a particular controller, where a DB operation is performed)
- 5) Charlie returns with LoR and gives it to Steve (the DB responds to the API with the correct data)
- 6) 7) 8) Charlie passes LoR to Steve (the API responds to the client with the requested data)
- 9) Steve hands me the LoR (the requested data is displayed in the browser with I the user can see and interact with)

Routes

**The bookshop is an analogy for how the
internet works, and how we interact with
websites/web applications**

Routes

What is a User

A user is us. A person using the website/
web app. In our bookshop, 'Me' is the
user.

Routes

What is a Client?

A client is a computer that sending a **request**, for example your computer when you access the internet.

Your computer requires **data** to be able to navigate a website/app, and so sends **requests** to a **server** to get data.

For example, if you want to see a particular book, a **request** with the **data** that you input (book title) will be sent to a **server** and a will receive a **response**.

Your access to the website will now change based upon the **response** that the **client** (your computer) receives from the **server**.

In our bookshop example, Steve is the **client**.

Routes

What is a Server?

A server is a computer that accepts **requests** from a **client** and returns a **response** to a **client**.

A **server** 'listens' for **requests** for a **client**. Once a **request** is received, the **request** data will be sent down a particular '**route**' in the code to the required destination.

The **server** can then use the **request** data to perform an operation, such as send data to and/or from a **database**.

The server can then package any data required and send it back to the **client**.
In our bookshop, Charlie is the **server**.

Routes

What is a Route?

A **route** is a pathway written in code;

"https://michaels-awesome-books-website/books/lordoftherings"

Here, we see that the address of the website is 'Michaels-awesome-bookswebsite'.

The route that the **request** for 'Lord of the Rings' needs to travel down is **'/books/lordoftherings'**

At the end of this **route**, as per our example, a **database** operation is run to **'GET'** the book.

Routes

What is a Route?

The the data of the book is then sent back by the **server** to the **client** via a **response**.

In the example above, the **route** is the path that Charlie walks down to get 'The Lord of the Rings'.

Routes

What is a Database?

A **database** is a place where data is stored.
It is essentially an electronic filing system where data is stored.

The data could be anything from login details, books titles, user likes on a post, and many more!

A **database** is accessed with what are called CRUD operations;

CREATE
READ
UPDATE
DELETE

Routes

What is Express

Express is a Node-based **Javascript API** framework.

Basically, **Express** is a set of pre-written code that can help to more easily write code for an **API** that will run in a **server**.

We can create the **server**, set it to listen for **requests**, create **routes**, perform **CRUD** operations on a **database** and return **responses**.