# Part 01

## /\* app.js \*/

Creates a connection to the database using the specified credentials and database name.

Creates the http server connection with request and response parameters.

If the servers gets a POST request, change the path to the specified case (e.g. /add, /newProduct, /delete).

If there is a POST request, the server must deal with those.

When the user inputs data into a form/text box, that data gets sent to the web server to be processed and stored.

request.on() method binds an event to an object.

When the user clicks the "Buy" button to purchase an item, the console logs the information.

The cartID increases by 1 for each item added, and the products in the cart are called from the database to be listed.

An error is thrown if that row doesn't exist in the database.

The end method finishes sending the request.

When the user clicks the "New Product" button, the related form is unhidden, now displayed to the user.

The submitted form sends the data to the server, adding it to the product JSON array.

When the user clicks "Show Cart", it displays the items that they added to their cart.

When the "Delete" button is clicked, the item gets removed from their cart, and from the cart array (if the code is fixed).

When the user clicks the "Login" button, the related form is unhidden, now displayed to the user.

Entering their name/password, a call is made to the database checking if that user exists.

If the user doesn't exist, an error is thrown stating such.

If the user exists, the login will be successful, with a message being logged to the console stating as such.

When the user clicks the "Checkout" button, they are prompted to login, if not done already.

Their order information is seny to the database to be stored, including customerID, and saleDate.

If successful, further order details are stored in the database, including orderID, productID, and quantity.

When the user clicks the "Register" button, the related form is unhidden, now displayed to the user.

If the user already exists in the database, a message will show.

If the user does not exist, their name, password, and address will be sent to the database to be stored in the Customer table.

There is a database query to create the Products table, with the data specified in SQL syntax.

After creating the table, information is added to it.

If the table already exists, a message will state that.

The getProducts() endpoint queries the database, retrieves the records from the database, and converts the data into a JSON array.

Marshalling: Process of converting or formatting data between the client and server.

## /\* shop.html \*/

Contains the CSS styling and layout.

Contains the scripts required to perform the functional tasks.

Starts the ajax call, which is a jquery library used on the client-side browser.

The ajax call sends a request to the server, gets a response back in an array, and displays the information in the DOM.

Each time a form needs to be shown to the user, the specified form is unhidden, becoming visible. Other forms are hidden.

When the DOM is fully downloaded, the browser calls the ready function.

Client-side programming manipulates the DOM to show stuff on the screen.

When getProducts() is called, the ajax call is sent, sending a request to the server, getting a response back in an array, then displaying the Products table/JSON array in the DOM.

# Part 02

