**Lecture 117**

**Adding the product ID to the path**

* Refer code 01-extracting-dynamic-params

**Lecture 118**

**Extracting Dynamic Params**

* Refer code 01-extracting-dynamic-params
* In such cases where there is a dynamic route and specific route , we have to place the specific route first

**Lecture 119**

**Loading Product Detail Data**

* Refer code 02-loading-product-detail-data.

**Lecture 120**

**Rendering the Product Detail View**

* Refer code 03-rendering-the-product-detail-view

**Lecture 121**

**Passing Data with POST Requests**

* we can also add an include add to cart ejs and put that entire code into that include here and then simply well include that include. So instead of having that code here in index.ejs, we can use the ejs tag with the minus to add include and then we go to well up one level and then into includes and then add to cart and we use that same code also in the product detail page here and also in the product list page. So there I also want to replace this form with my include. Now let's see if that still works, if I now go to products, now we get this error because if you have an include in a for loop or in a loop in general as we have it here, we're looping through all the products and then product is a local variable available in that loop only, then in include, included in the loop unfortunately doesn't get that variable by default but you can pass it to that include here and you can do so by adding a second argument to the include function where you again pass an object and simply add that variable again. So here this product is what will be available in include and on the right side, you include or you add the value you have available in this file, so product in the loop is now passed to product in the include. This of course needs to be done for all our include, so also in the index.ejs file and not for the one in the product detail though because there, it's not inside a loop, so it's available globally and therefore works.
* Refer code 04-adding-a-cart-model

**Lecture 122**

**Adding a Cart Model**

* Refer code 04-adding-a-cart-model.
* A static method is created for adding the products since there is only one cart for the user. We don't want to create a Cart instance everytime we want to add a new product to cart.

**Lecture 123**

**Using Query Params**

* Query params can be accessed from req.query object

**Lecture 124**

**Prepopulating the edit product page with data**

* Refer code 05-pre-populating-the-edit-product-page

**Lecture 125**

**Linking to the Edit page**

**Lecture 126**

**Editing the product data**

* Refer code 06-editing-the-product-data

**Lecture 127**

**Adding the product-delete functionality**

* Refer code 07-adding-the-product-delete-functionality

**Lecture 128**

**Deleting cart items**

* Refer code 08-deleting-cart-items

**Lecture 129**

**Displaying cart items on the cart page**

**Lecture 130**

**Deleting Cart items**

* Refer code 09-fixing-a-delete-product-bug

**Lecture 131**

**Fixing a delete product bug**

* Refer code 09-fixing-a-delete-product-bug
* Docs: <https://expressjs.com/en/guide/routing.html>