



Laravel Basic Tasks

Larastore

Duration: 5 Hours



Create an online store to manage the Products, this application should contain 3 main pages [View Products, Add Products, Update Products] Remove Products is just an action without a page. follow the below steps this will help to put you on the right way.

1. Create the Product Model

Create a Model called Product make sure to follow Laravel framework naming conventions, and make sure that your Product model includes **fillable attribute**:

- o id
- o product_name
- o product_description
- o product_price

2. Create the Products Controller

After creating the model, we need to create a Products Controller this controller **should include** all basic Methods.



1. index: to redirect the user to the landing page
2. create: to create a new product [redirect to the related view page].
3. Store: to store the product coming from the view
4. edit: to update a specific product[redirect to the related view page].
5. Updated: to store the updated values
6. destroy: to delete a product.

Verb	URI	Action	Route Name
GET	/products	index	products.index
GET	/products/create	create	products.create
POST	/products	store	products.store
GET	/products/{product}	show	products.show
GET	/products/{product}/edit	edit	products.edit
PUT/PATCH	/products/{product}	update	products.update
DELETE	/products/{product}	destroy	products.destroy

3. Create the Categories Model

Create a Model called Category make sure to follow Laravel framework naming conventions, and make sure that your Category model includes **fillable attribute**:

- o id
- o category_name
- o category_description



4. Create the Categories Controller

After creating the model, we need to create a Categories Controller this controller **should include** all basic Methods.

7. index: to redirect the user to the main category page
8. create: to create a new category [redirect to the related view page].
9. Store: to store the category coming from the view
10. edit: to update a specific category[redirect to the related view page].
11. Updated: to store the updated values
12. destroy: to delete a category.

Verb	URI	Action	Route Name
GET	<code>/categories</code>	index	categories.index
GET	<code>/categories/create</code>	create	categories.create
POST	<code>/categories</code>	store	categories.store
GET	<code>/categories/{category}</code>	show	categories.show
GET	<code>/categories/{category}/edit</code>	edit	categories.edit
PUT/PATCH	<code>/categories/{category}</code>	update	categories.update
DELETE	<code>/categories/{category}</code>	destroy	categories..destroy



5. Create the related views page

You will create four pages for our CRUD application, the first page contains a data grid showing the records from the products database table. It also has actions for each record displayed in the grid, you may choose to view its details, update, or delete it.

You'll also add a create button on the top of the data grid that can be used for creating new records in the product's table.

6. Create the data migration.

This is important step for our application this migration will help me to create the database tables inside Laravel by artisan commands, the database schema should match the model attributes also make sure the database tables should be plural

****Important: Don't make the tables manually**

7. Edit .env file.

In this file, you should edit all the information related to the database to make the connection.

8. Make the relation inside the model

Edit the model files and ensure the relationship is made inside the model

9. Make the corresponding edits for the migration files.

Edit the migration and ensure the relationship is made inside the migration.