5. Now we get the following structure:

book 1 cart

— src

— features

— .gitignore

— composer.j son

— vendor

**Writing extension code**

Copy the Cart, Storageinterface, and Sessionstorage classes from the Unit testing with PHPUnit recipe.

Finally, we get the following structure:

book 1 cart

— src

storage

I— SessionStorage.php I— Storageinterface.php Cart.php

— features

— .gitignore

— composer.j son

— vendor

**Writing extension tests**

1. Add the book/cart/features/bootstrap/bootstrap . php entry script:

<?php

defined('YII\_DEBUG') or define('YII\_DEBUG', true); defined('YII\_ENV') or define('YII\_ENV', 'test');

require\_once DIR . '/../../vendor/yiisoft/yii2/Yii.php';

2. Create the features/cart.feature file and write cart testing scenarios:

Feature: Shopping cart In order to buy products As a customer

I need to be able to put interesting products into a cart

Scenario: Checking empty cart Given there is a clean cart Then I should have 0 products Then I should have 0 product And the overall cart amount should be 0

Scenario: Adding products to the cart Given there is a clean cart When I add 3 pieces of 5 product Then I should have 3 pieces of 5 product And I should have 1 product And the overall cart amount should be 3

When I add 14 pieces of 7 product Then I should have 3 pieces of 5 product