Breakdown and Estimation of Laravel Test - Developnow

Assumptions

- There will be a set of APIs those needs to be developed that mimics the ecommerce
- Developer is free to modify the to add new columns in to mimic the ecommerce without missing the points on test

Goals

- To develop test application and submit it by Monday, July 04, 2022 with unit test.

Requirements

	User Story	Task	Estimation in Story point	Notes
1		Setup task • Install Laravel Project in latest version • Install swagger and others • Setup php unit configuration for the database and others • Setup Repository Coding pattern	1	
2	As a admin I should be able to get login as a user	Create database migration files to admin information • use single source of truth for database names • use soft-deletes feature • use existing migration file do not need to write new files	1	
3		Create POST API `/api/v1/auth/tokens` to get login • make use of sanctum and make separate guard so that separation of concern become clear while working with customer login	1	
4	As a admin I should be able to manage the products categories	Create database migration files to store the information of the product categories • use single source of truth for database names • use soft-deletes feature • create model and mange needful relationships • column names as • name(string)	1	
5		Create GET API `/api/v1/product-categories` to get the list of products • include filters over there • make use cursor based pagination	1	
6		Create POST API `/api/v1/-categories` to create the product	1	
7		Create GET API `/api/v1/-categories/{id}` to get the detail of the	1	
8		Create PUT API `/api/v1/-categories/{id}` to update the product	1	
9		Create DELETE API `/api/v1/-categories/{id}` to delete the product		
10	As a admin I should be able to manage the products	Create database migration files to store the information of the product • use single source of truth for database names • use soft-deletes feature • create model and mange needful relationship • columns names as follows- • name(string) • price(int) • category_id(int) • image(string)	1	Rate must be integer: but must be able to store float amounts. Make use of accessors and mutuators to do so.

11		Create GET API `/api/v1/products` to get the list of products • include filters over there • make use pagination	1	
12		Create POST API `/api/v1/products` to create the product	1	
13		Create GET API `/api/v1/products/{id}` to get the detail of the	1	
14		Create PUT API `/api/v1/products/{id}` to update the product	1	
15		Create DELETE API `/api/v1/products/{id}` to delete the product		
16	As a client I should be able to get login as a customer	Create database migration files to customer information • use single source of truth for database names • use soft-deletes feature • use existing migration file do not need to write new files • columns would be like • first_name(string) • middle_name(string) • last_name(string) • email(string) • mobile_number(string) • credit_card_number(string) • total_transaction_amount(float) • total_transaction(int)	1	
17		Create POST API `/api/v1/customers/auth/tokens` to get login • make use of sanctum and make separate guard so that separation of concern become clear while working with admin login login	1	
18	As a customer I should be able to manage my cart	Create database migration files to store the information of the cart items • use single source of truth for database names • create model and mange needful relationship • columns names as follows- • product_id(int) • customer_id(int) • quantity(int) • price(float) • status	1	
19		Create GET API `/api/v1/carts` to get the list of products • include filters over there • make use pagination	1	
20		Create POST API `/api/v1/carts` to create the product	1	
21		Create GET API `/api/v1/carts/{id}` to get the detail of the	1	
22		Create PUT API `/api/v1/carts/{id}` to update the product	1	
23		Create DELETE API `/api/v1/carts/{id}` to delete the product		