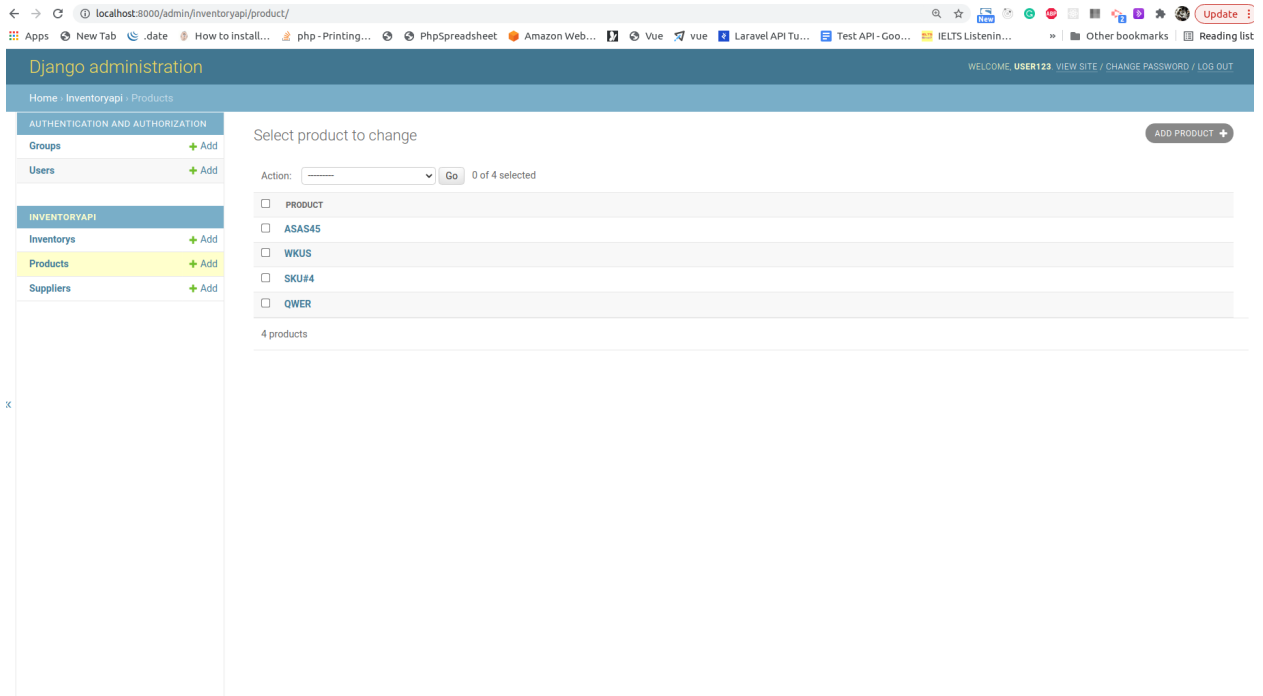


Assumptions:

1. product to inventory is a one-to-one mapping, so product id cannot be duplicated in inventory model.
2. Supplier to inventory, a foreign key is available in inventory but a supplier can supply more than one product.
- 3.No code comments are added.
- 4.No automated data is considered such as created_at and updated_at
- 5.No design patten are considered
- 6.No security is considered such as Authentication and Authorization for API calls.
7.
Admin useer name:user123
Adin password :user123
8. Apart from the main app another two apps are registered.
9. Please make sure to implement the dependencies, eg: pip insatl
- 10.No styles and front look are considered.

Answers:

- a. Three models are created (Supplier, Inventory, Product).SQLite is used



b. REST APIs, Django Rest Framework (DRF) API endpoint

URLs:

http://localhost:8000/api/inventory/?name=ABC123&search_fields=name

<http://localhost:8000/>

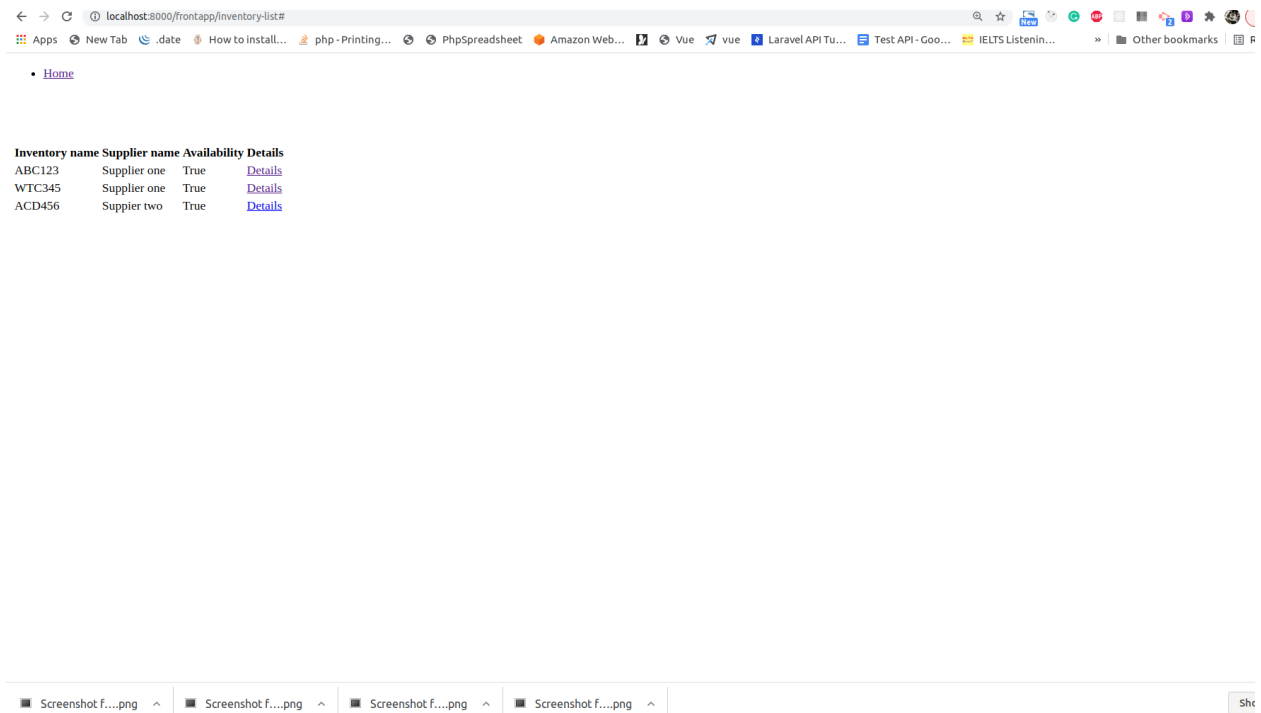
<http://localhost:8000/api/inventory/>

<http://localhost:8000/api/inventory/2/>

A screenshot of a web browser displaying the Django REST framework API Root page. The browser's address bar shows the URL 'localhost:8000'. The page title is 'Django REST framework'. The main content area features a large heading 'Api Root' and a subheading 'The default basic root view for DefaultRouter'. Below this, a 'GET /' endpoint is listed with its response details: 'HTTP 200 OK', 'Allow: GET, HEAD, OPTIONS', 'Content-Type: application/json', and 'Vary: Accept'. The response body is a JSON object containing two API endpoints: 'api/products' and 'api/inventory'. The browser's taskbar at the bottom shows several open tabs, including 'Apps', 'New Tab', and various development tools and documentation pages.

Create a view at “/inventory” to display the list of inventories and their suppliers?

<http://localhost:8000/frontapp/inventory-list>

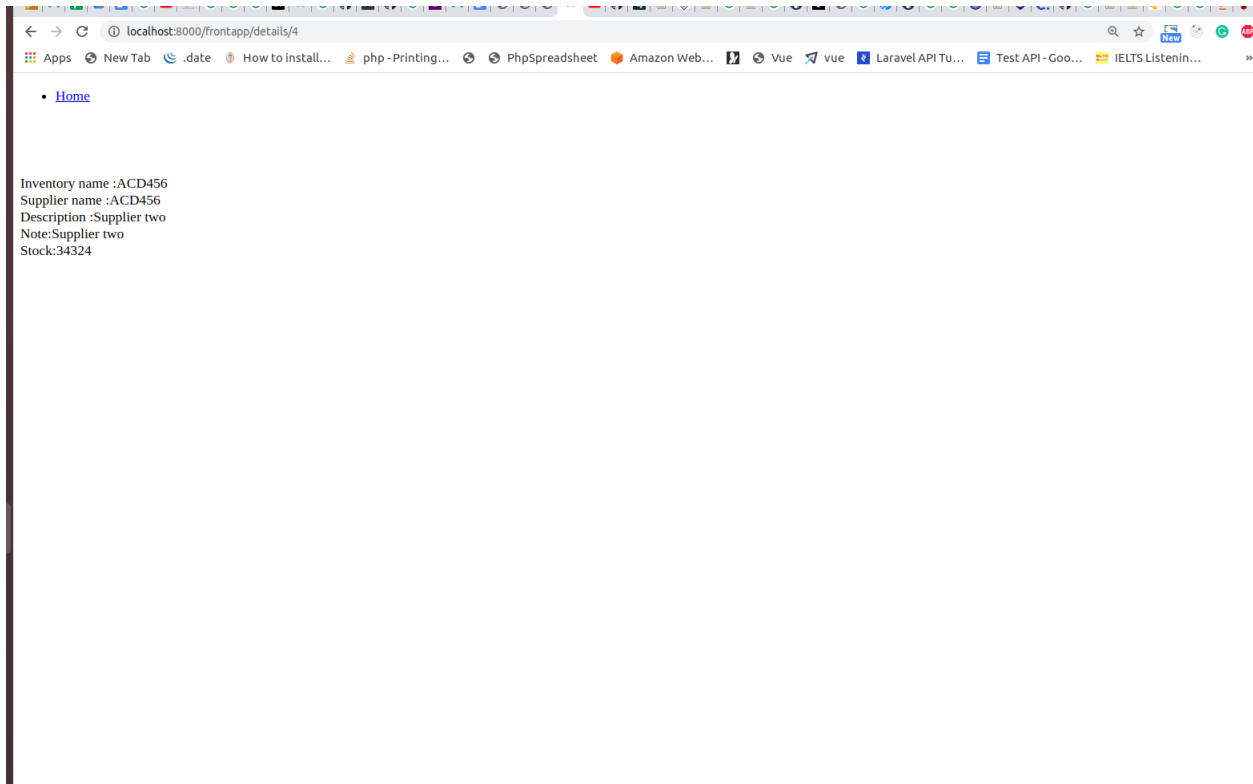


c)

c. Create a view at “/inventory/<id>” to display a selected item from the list page using only views and template rendering to show the item.

Note:I have not included the thumb image.

<http://localhost:8000/frontapp/details/4>

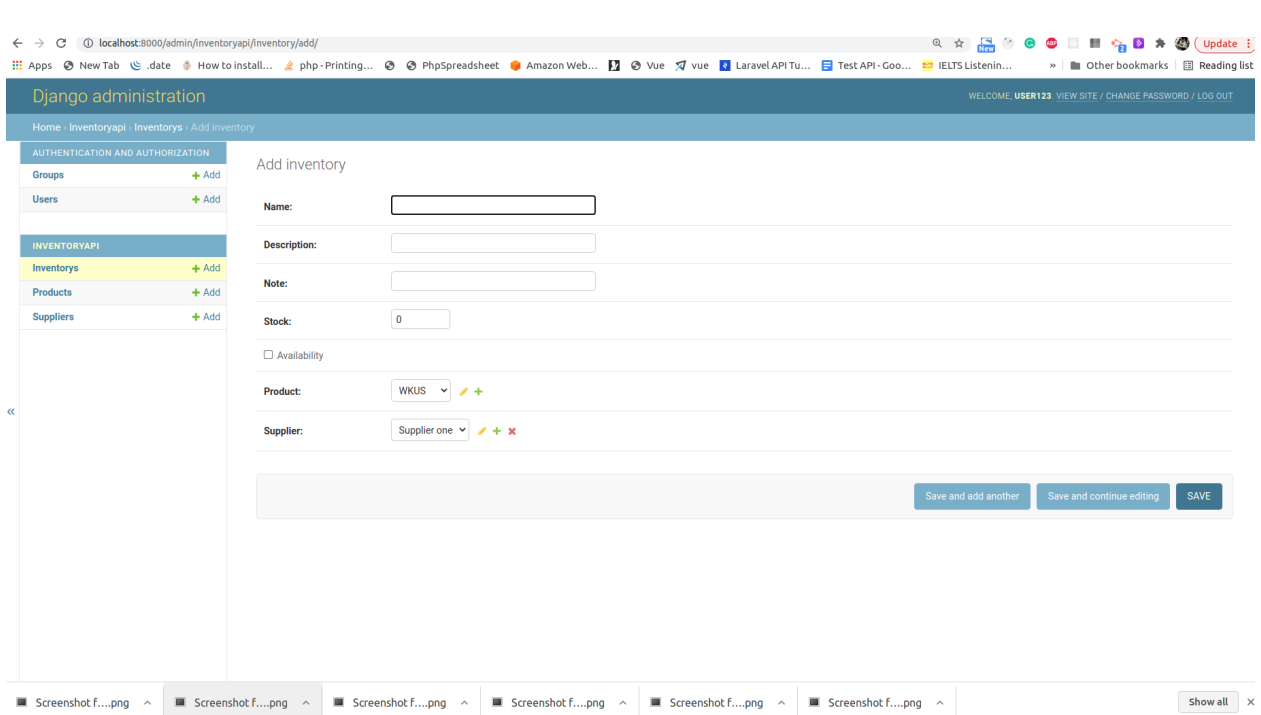


d. Add the inventory model to the Django's built-in admin panel to enable CRUD management of inventory (Add, update, remove stock). Create a superuser account to access the admin panel.

<http://localhost:8000/admin/inventoryapi/inventory/add/>

Admin useer name:user123

Adin password :user123



e.) unit tests, `inventory/frontapp/tests.py`.

The `"/api/inventory"` => `'http://localhost:8000/api/inventory/'`

The `"/inventory"` page => `'frontapp/inventory-list'`

The `"/inventory/<id>"` => `'frontapp/details/2'`

