

Ping [thev.me/telegram](https://t.me/thevme) or [thev.me/whatsapp](https://t.me/thevme) if you have questions

Full-Stack Engineer @ Unit Nukleus GovTech, Kem. Digital

We have contract-for-service positions open for:

- Juniors (RM6,000-RM8,000/month)
- Mid-level (RM9,000-RM11,000/month)
- Seniors (RM13,000+/month)

Key details:

- WFH default (office space available)
 - Excellent work device provided
 - Fast-paced, building tech at the heart of government
 - High trust, flexibility, and autonomy
-

Summary of Instructions

1. Complete the form at kongsila.com/form in full
2. Complete the audition task (see below)
3. Go back to kongsila.com/form and amend your last 2 responses indicating that your task is complete (+ link to GitHub repo)
 - a. **The task must be completed by 16 March 2024, 2359hrs**
4. If successful, you will be invited for 1 interview.
If unsuccessful, you will also be notified.

AUDITION TASK

Recommended tech stack

Backend: Node, Django/Flask, GO

Frontend: React, Vue, Angular

BASE TASK (JUNIOR)

Create a **simple inventory management system**. This requires you to:

- Build APIs and front-end interfaces to show product listings as well as specific products, with navigations that you deem useful and sensible.
- Set up a database table for the products within the inventory, and another table for suppliers for products within the inventory. The fields, and relations for each database table are left to your discretion. You may use any DB you prefer, but for simplicity, SQLite (for SQL) or Mongo (for NoSQL) will suffice. Use constraints where relevant.
- Push your code to a public repository on GitHub.
- **Deploy to a public cloud of your choice (free tiers should suffice)**
- Document anything necessary in a README (+ link to live site)

Technical Specifics:

- 1) [backend] Create an API with the endpoint **/api/inventory** which lists all products in the inventory. Allow for query parameters, to help filter and sort products by the category of your choice.
 - a. Pagination must be supported for mid to senior level engineers.
- 2) [backend] Create an API, with the endpoint **/api/inventory/[product-id]** which only returns information for the product-id chosen.
- 3) [backend] Create additional endpoints:
 - a. **/api/add-inventory** to add a product into the product and supplier tables

Ping [thev.me/telegram](https://t.me/thevme) or [thev.me/whatsapp](https://wa.me/thevme) if you have questions

- b. `/api/delete-inventory` to remove a product from the product and supplier tables
 - c. `/api/update-inventory` to update information on an existing product or supplier
- 4) [backend] Create an API or management command to populate the DB with at least 1000 rows of data (i.e. 1000 products)
- 5) [frontend] In `/inventory`, display all products in a table component.
 - a. The page should have mechanisms to filter, sort products.
 - b. The page should allow users to add product(s) into the inventory.
 - c. Pagination is required for mid to senior level engineers.
- 6) [frontend] In `/inventory/[product-id]`, display the product page selected from `/inventory`.
 - a. The page should display information about the product.
 - b. The page should also allow users to update and delete products.
 - c. Bonus: Add a static image of the product

EXTENSION (mandatory for MID-LEVEL)

Building the tasks above, create a simple authorization layer, where users can choose from the following roles (reflected in all relevant interfaces):

Guest: CAN view all pages, but CANNOT add, delete, or update any product

Admin: CAN view all pages, and CAN add, delete and update any product.

The state of the user's role should persist across all pages, even after a page refresh or a tab close.

FINAL EXTENSION (mandatory for SENIOR)

Ping [thev.me/telegram](https://t.me/thevme) or [thev.me/whatsapp](https://t.me/thevme) if you have questions

- 1) Instead of having static roles, allow for dynamic roles and permission level checks. Users must be able to create a new role and assign permissions, which controls the user's ability within the system. Permissions may be described in the form:

Action	Subject
create	Product
view	Product
update	Product
delete	Product

Users must be able to assume the new role and have their permissions be reflected on every actionable item in the page.

- 2) Write simple unit test cases to test the APIs you created. The following criteria should be checked:
 - a. [/api/inventory](#) API returns 200 OK status
 - b. [/api/inventory/\[product-id\]](#) API returns 200 OK status