Backend Developer Assignment

Ogni software aziendale ha un supporto integrato per gli organigrammi, al fine di rappresentare gerarchie e ruoli all'interno di un'azienda. Un modo comune e conveniente per gestirli con una struttura ad albero nei database relazionali è il modello "Nested Set" (cliccare il link per ulteriori dettagli).

Struttura del Database:
le seguenti tabelle MySQL contengono un organigramma, insieme ai nomi dei ruoli in varie lingue, normalizzate secondo il modello Nested Set.

node_tree			
idNode	**level**	**iLeft**	**iRight**
1	2	2	3
2	2	4	5
3	2	6	7
4	2	8	9
5	1	1	24
6	2	10	11
7	2	12	19
8	3	15	16
9	3	17	18
10	2	20	21
11	3	13	14
12	2	22	23

node_tree_names (idNode foreign key node_tree.idNode)		
idNode	**language**	**NodeName**
1	english	Marketing
1	italian	Marketing
2	english	Helpdesk
2	italian	Supporto Tecnico
3	english	Managers
3	italian	Manager
4	english	Customer Account
4	italian	Assistenza Cliente
5	english	0brand Srl
5	italian	0brand Srl
6	english	Accounting
6	italian	Amministrazione
7	english	Sales
7	italian	Supporto Vendite
8	english	Italy
8	italian	Italia
9	english	Europe
9	italian	Europa
10	english	Developers
10	italian	Sviluppatori
11	english	North America
11	italian	Nord America
12	english	Quality Assurance
12	italian	Controllo Qualità

Specifica dei requisiti:

un'applicazione front end deve recuperare i nodi dell'organigramma e mostrarli con una visualizzazione ad albero e, quindi, dipende da un'API di backend per ottenere in modo efficiente tali dati.

Al candidato è richiesto di implementare uno script php, api.php, per restituire i nodi dell'organigramma potendo specifica da quale livello partire e permettendo la paginazione.

Lo script verrà chiamato tramite HTTP (metodo GET) attraverso un server web Apache e riceverà i seguenti parametri di input:

- node_id (numero intero, **obbligatorio**): l'ID univoco del nodo selezionato.
- language (enum, obbligatorio): identificatore della lingua. Valori possibili: "english", "italian".
- search_keyword (stringa, opzionale): un termine di ricerca utilizzato per filtrare i risultati. Se fornito, limita i risultati in tutti i nodi figli sotto node_id il cui nodeName nella lingua data contiene la parola chiave di ricerca (senza distinzione tra maiuscole e minuscole) .
- page_num (numero intero, opzionale): l'identificatore 0-based della pagina da recuperare. Se non specificato il valore predefinito è 0.
- page_size (numero intero, opzionale): la dimensione, in termini di record, della pagina da recuperare, compresa tra 0 e 1000. Se non fornito, il valore predefinito è 100.

L'API dovrebbe restituire un JSON con i seguenti campi:

- nodes (array, **obbligatorio**): 0 o più nodi che rispettano le seguenti condizioni. Ogni nodo deve contenere:
  - node_id (integer, **obbligatorio**): l'ID univoco del nodo selezionato.
  - name (string, **obbligatorio**): il nome del nodo tradotto nel linguaggio richiesto.
  - children_count (integer, **obbligatorio**): il numero di figli di **questo** nodo.
- next_page (integer, opzionale): il numero della prossima pagina se i record sono più di page_size (solo per posizioni Senior)
- prev_page (integer, opzionale): il numero della pagina precedente se page_num è oltre la pagina 0. (solo per posizioni Senior)
- error (string, opzionale): se è avvenuto un errore compilare con il messaggio dell'errore.

Vincoli:

la soluzione proposta deve controllare che tutti i parametri richiesti siano passati e validi e restituisca, in caso di errore, i seguenti messaggi:

- "ID nodo non valido" (se node_id non viene trovato).
- "Parametri obbligatori mancanti" (se un qualsiasi parametro di input richiesto non viene passato o ha valore vuoto).
- "Numero di pagina richiesto non valido" (se page_num non è un numero 0-based valido).
- "Richiesto formato pagina non valido" (se page_size non rientra nell'intervallo di validità).
- Nessun framework è consentito (solo per posizioni Senior e Mid)
- Il codice fornito deve essere conforme a PHP 7.3 e utilizzare l'estensione "pdo_mysql" per eseguire query verso il database.
- La qualità del codice non è opzionale: commenti, nomi di variabili chiari e significativi e codice PHP ben strutturato e progettato sarà altamente considerato nella valutazione finale di questo test.

Il candidato deve fornire un repository (solo per Senior e Mid, Junior possono fornire un archivio compresso) con la seguente struttura "minima":

- api.php (punto di ingresso)
- config.php (file di configurazione contenente le credenziali di accesso al database ed eventuali configurazioni) per candidati junior che utilizzano un framework questo file sarà il file di enviroment del framework.
- tables.sql (script SQL di definizione delle tabelle) per candidati junior che utilizzano un framework possono essere anche delle migration.
- data.sql (script SQL per l'inserimento dei dati) per candidati junior che utilizzano un framework possono essere anche dei seed.