



WireApps

Back-End Software Engineer

Back-End Technical Assessment - Node.js

Prepared for: WireApps Ltd
Prepared by: *Pivithuru Jayakody – Software Engineer*
Document: *Back-end Technical Assessment - Node.js*



Document

This assessment is part of the interview screening process for Back-end Software Engineers. To face this assessment, a candidate will have completed the first interview panel. Upon successful completion of the assessment, the candidate will be guided through the next stage of the interview process.

Scenario

There is a pharmacy, where the main people are the owner, managers, and cashiers. They need a system to make their business processes more efficient by reducing their workload that has to be performed manually.

This new system should be able to:

- Authenticate Users using Username and Password
 - ❖ Must include Name, Username, Password, Role (Feel free to add any other appropriate fields)
- Manage Inventory of Medication
 - ❖ Query existing medicines
 - Must include Name, Description, and Quantity (Feel free to add any other appropriate fields)
 - ❖ Insert and update records
 - ❖ Soft delete medication records
 - ❖ Delete records
- Manage Customer Records
 - ❖ Query existing customers
 - ❖ Insert and Update customer details
 - ❖ Soft delete customer records
 - ❖ Delete customer records
- Allow actions based on User Roles
 - ❖ All users can query medication records and customer records



- ❖ Only the Owner can insert, update, soft delete and (permanently) delete medication records and customer records. Managers and Cashiers should not be able to insert or (permanently) delete any data.
- ❖ Managers can update and soft delete medication records and customer records, but they cannot insert or (permanently) delete data.
- ❖ Cashiers can update medication records and customer records, but they cannot insert, or delete data



Deliverables

Note: You do not need to develop a frontend application like a dashboard.

- Deliver the following:
 1. REST API running on Node.js
 - You may use a framework like Express, NestJS or FoalTS
 - You may create a SQLite database to store test data (or you can use mock data extracted from a file.)
 2. ER Diagram including the data models in the system
 3. Postman collection of the API endpoints
 4. README file inside your repository
 - Include instructions on how to run the project locally

How to deliver:

- Upload your codebase to GitHub as a public repository and share the URL.
- You may take a maximum of 3 days to complete this task