

**Full Stack Web Developer**

Technical Test

**=========================**

**Candidate Name: [replace with your name and surname]**

**Date of test: [ 05-09-2024]**

**Duration: [2.5 Hours]**

**

**Instructions:**

* This test is divided into four sections: Database Design, Back-End Development, Front-End Development, and Problem Solving.
* Complete each section within the allotted time.
* Provide clear and well-documented code at each step.
* You may use any front-end framework (React, Angular, Vue.js, etc.) and any back-end framework or language (Node.js, Django, Flask, PHP, etc.) of your choice.
* You are free to download any necessary software to set up your development environment.
* Ensure that you include a README file with detailed instructions on how to set up and run the application.
* You can use the internet to read documentation, but using any AI tools is forbidden.
* **Save your work under your name in a zip file on the desktop** when you have completed the test.

**System Description:**

You are tasked with developing a **Ticketing System** designed to connect employees with IT support staff.

When an employee needs to repair an IT item, they should use the ticketing system to submit a request. The system will automatically generate a unique ticket number and ask the employee for the following information:

* **Employee name**
* **Location** (HQ, Shatila, Nabaa, Tripoli, Bekaa)
* **Problem description**
* **Problem type** (hardware or software)
* **Device type** (Laptop, Mobile, Printer, Access Point)

If the problem is software-related, additional information will be required:

* **OS version**
* **Affected software application** (Office, Adobe Acrobat, Chrome, X-Gate, P&L, QuickBooks, BOB)
* **Error code**, if available
* **Screenshot**, if available

If the problem is hardware-related, additional information will be required:

* **Device Serial Number (SN)**
* **Picture**, if available

Upon submission, the system should assign a status of "pending" to the ticket, along with the date and time. The ticket will be assigned to an IT specialist based on the location of the problem. The system will display the ticket information to the employee, including the name of the assigned IT specialist.

There are three IT specialists:

* **Firas** - Responsible for HQ
* **Wael** - Responsible for Bekaa and Tripoli
* **Hussein** - Responsible for Shatila and Nabaa

IT specialists can:

* Review all tickets in the locations they are responsible for.
* Search for specific tickets by number, employee name, SN (for hardware tickets), device brand (for hardware tickets), or error code (for software tickets).
* Filter tickets by status, device type, or problem type.
* Change the ticket status to "pending," "in process," or "completed."

**Question 1: Database Schema Design (30 minutes)**

**Task:**

* Design a relational database schema for the ticketing system, including tables for Tickets, Employees, IT Specialists, and Centers. Establish proper relationships between these tables.

**Requirements:**

* Use the inheritance principle to differentiate between hardware and software tickets.
* Ensure the schema supports querying tickets by various criteria (e.g., ticket number, employee name, SN, error code).
* Implement foreign keys to maintain referential integrity.

**Deliverables:**

* Provide the SQL script for creating the database schema.
* Include a brief explanation of your design decisions.

**Question 2: API Creation (30 minutes)**

**Task:**

* Create an API endpoint that allows an employee to submit a new ticket. The API should handle the request and insert the appropriate data into the database.

**Requirements:**

* Implement the API using any back-end framework or language (Node.js, Django, Flask, PHP, etc.).
* Ensure that the API is secure, handles errors gracefully, and returns meaningful responses.

**Deliverables:**

* Provide the API code, along with any supporting files.
* Include documentation on how to test and run the API.

**Question 3: Form Creation (30 minutes)**

**Task:**

* Develop a simple form using any front-end framework (React, Angular, or Vue.js) that an employee can use to submit a ticket.

**Requirements:**

* Implement the form using any front-end framework.
* Ensure that the form is responsive and user-friendly.
* Validate the form inputs before submission.

**Deliverables:**

* Provide the code for the form and any supporting files.
* Ensure that the form is functional and handles the dynamic fields based on the problem type selection.

**Question 4: Problem Solving (30 minutes)**

**Task:**

* Provide a brief outline of how you would implement a search function for IT specialists to search tickets by status, device type, problem type, or ticket number.

**Requirements:**

* Describe the logic and database queries you would use to implement this search functionality, considering performance and scalability.
* Provide example SQL queries or pseudocode that demonstrate how you would search for tickets using the specified criteria.
* Briefly explain any strategies you would employ to ensure that the search functionality is efficient and scalable, particularly in a system with a large number of tickets.

**Deliverables:**

* Write a clear and concise outline explaining the approach you would take to implement the search functionality.
* Include sample queries or pseudocode.
* Discuss performance considerations.

Good luck!

Use indexing multi index for multi columns

Also elastic search to search using ms

And we can use limit to paginate the data