Soal Ujian untuk Calon Junior Programmer

Proyek: Aplikasi Backend Survei Pilkada

Anda diminta untuk mengembangkan aplikasi backend sederhana untuk survei Pilkada. Aplikasi ini akan digunakan untuk mengelola data kandidat, responden, dan hasil survei, serta menyimpan jawaban responden terkait dukungan mereka terhadap para kandidat.

Kebutuhan Proyek:

Manajemen Pengguna (User Management)

Buat sistem otentikasi menggunakan JWT untuk memastikan bahwa hanya admin yang bisa mengelola data kandidat, survei, dan hasil survei.

Terdapat dua jenis pengguna: Admin dan Petugas Survei. Admin memiliki akses penuh ke semua fitur aplikasi, sedangkan Petugas Survei hanya dapat mengelola responden dan memasukkan hasil survei.

Manajemen Data Kandidat (Candidate Management)

Admin dapat menambah, mengedit, menghapus, dan melihat daftar kandidat yang berpartisipasi dalam Pilkada.

Data kandidat terdiri dari: id, name, party, position.

Manajemen Data Responden (Respondent Management)

Petugas Survei dapat menambah, mengedit, menghapus, dan melihat daftar responden.

Data responden terdiri dari: id, name, age, gender, address.

Manajemen Survei dan Jawaban (Survey Management)

Admin dapat menambahkan pertanyaan survei yang berisi dukungan kepada para kandidat.

Setiap responden dapat mengisi survei yang mendukung kandidat yang mereka pilih. Jawaban disimpan dalam tabel terpisah yang menghubungkan responden, kandidat, dan hasil jawabannya.

Analisis Hasil Survei

Sistem harus dapat menghitung dukungan kandidat berdasarkan jumlah suara dari responden.

Data dukungan setiap kandidat harus ditampilkan dalam bentuk statistik sederhana (misal: total dukungan untuk setiap kandidat).

Instruksi Pengerjaan:

Autentikasi & Otorisasi

Implementasikan otentikasi menggunakan JWT untuk melindungi rute-rute yang memerlukan otorisasi, seperti rute untuk menambah kandidat, mengelola responden, dan mengisi survei.

Endpoint-Endpoint yang Diperlukan:

POST /auth/register — Mendaftarkan admin atau petugas survei.

POST /auth/login — Login untuk mendapatkan JWT token.

POST /candidates — Menambahkan kandidat (admin only).

GET /candidates — Mendapatkan daftar kandidat (accessible untuk admin dan petugas survei).

PUT /candidates/:id — Mengubah data kandidat (admin only).

DELETE /candidates/:id — Menghapus kandidat (admin only).

POST /respondents — Menambahkan responden (petugas survei).

GET /respondents — Mendapatkan daftar responden (petugas survei).

PUT /respondents/:id — Mengubah data responden (petugas survei).

DELETE /respondents/:id — Menghapus responden (petugas survei).

POST /surveys/responses — Memasukkan jawaban survei dari responden yang mendukung kandidat tertentu (petugas survei).

GET /surveys/results — Menampilkan hasil dukungan kandidat dalam bentuk statistik.

Database Design: Buatlah model database menggunakan Sequelize yang mencakup tabel berikut:

User: id, name, email, password, role.

Candidate: id, name, party, position.

Respondent: id, name, age, gender, address.

Survey\_Response: id, respondent\_id, candidate\_id, response\_text (opsional), yang menyimpan data jawaban dukungan dari responden.

Penggunaan Sequelize:

Gunakan Sequelize untuk membuat relasi antara tabel Respondent, Candidate, dan Survey\_Response.

Gunakan Sequelize Transactions untuk memastikan bahwa operasi CRUD seperti menambah kandidat, responden, atau jawaban survei dilakukan secara aman dan konsisten.

Validasi Input:

Gunakan Joi atau library validasi lain untuk memastikan bahwa data yang diterima di endpoint sesuai dengan format yang benar.

Bonus Poin:

Implementasikan Winston untuk logging aktivitas pengguna seperti login, menambah kandidat, atau mengisi survei.

Buat rute untuk mengekspor data hasil survei ke dalam format CSV.

Buat dashboard sederhana untuk admin yang menampilkan statistik hasil survei dalam bentuk visualisasi sederhana (misal: chart dukungan kandidat).

Kriteria Penilaian:

Kode yang bersih dan modular.

Kemampuan menggunakan Sequelize untuk mengelola model dan relasi database.

Pemahaman mengenai otentikasi dan otorisasi menggunakan JWT.

Kemampuan menerapkan validasi input.

Kemampuan menggunakan Express untuk membuat RESTful API dengan rute-rute yang jelas.

Implementasi fitur tambahan seperti logging dan eksport data akan memberikan nilai tambah.

Catatan:

Proyek ini dirancang untuk menguji kemampuan Anda dalam membangun aplikasi backend dengan Node.js dan Sequelize. Pastikan untuk membuat arsitektur aplikasi yang modular, dengan pemisahan yang jelas antara controller, service, dan model.

Dengan memberikan soal ini, Anda bisa mengevaluasi kemampuan calon junior programmer dalam memahami konsep penting seperti manajemen data, autentikasi, penggunaan ORM (Sequelize), serta penerapan RESTful API di aplikasi backend

Keterangan lanjut:

*https://chatgpt.com/share/ece538cd-a6f9-4b64-942b-a5d7cda8240e*