Backend System: Relevel Employee Hiring

Relevel
by Unacademy

Educator Introduction (5 min)

- Greet Students
- Enquire students if they have revised previous class content

Concept to be covered in class:

Problem Statement -

As time passes by, the unemployment rate is increasing, so to solve this issue, Relevel decided to create an application so that we all can provide jobs to the individual where we can log in and apply to a job by looking at its JD and if suitable then the companies can begin the Hiring process.

Your task is to go through the Problem Statement, Requirements and create a Relevel Employee Hiring.

Problem Requisite

- The candidate should make sure GIT, NodeJS, and MongoDB are already installed on their system before starting the development.

User Stories

Problem 1 (Estimated Time to complete ~2 Hours)

Story-1 (Define data model)

Create a clear data model for Admin, Company, Job, Student.Create mongoDB schema, for example, should be included in your code base.

- Add Appropriate validations and reference keys.
- Spend some time creating some "seed data" that covers all of the use cases you want to show.

Story-2 (Create APIs for Authentication)

Your APIs should provide the following functionality to downstream consumers of the API (in a RESTful way).

- Create appropriate APIs for /signup/:role, /login/:role.
- /signup/:role for creating a new account based on your role(Company or Individual).
- /login/:role for logging in to your accountbased on your role(Company or Individual).
- Also, have to validate profile details and login details.
- For success, return proper response codes.
- For failure, return proper response codes.
- Use an appropriate HTTP method out of GET, POST, PUT, PATCH, DELETE, etc.

Story-3 (Create APIs for the Companies)

Your APIs should provide the following functionality to downstream consumers of the API (in a RESTful way):

- Create appropriate APIs for viewing the details using id and also deleting using the same.
- There will be rules like deletion can only be done by the person who has the admin access and which type of data can be seen by everyone or only one particular type.
- Display status & response code in the response.
- The successful request should return 200 response codes.
- Use an appropriate HTTP method out of GET, POST, PUT, PATCH, DELETE, etc.

Problem 1 (Estimated Time to complete ~1.5 Hours)

Story-4 (Create APIs for the jobs)

Your APIs should provide the following functionality to downstream consumers of the API (in a RESTful way):

- Create appropriate APIs for getting the jobs opening, creating the job opening, get job by id, apply to a job, delete job opening.
- Opening can be only created by companies and apply can be only be done by students, Be open to ideas and create a well defined api.
- Display status & response code in the response.
- The successful request should return 200 response codes.
- Use an appropriate HTTP method out of GET, POST, PUT, PATCH, DELETE, etc.

Story-5 (Create APIs for the profile)

Your APIs should provide the following functionality to downstream consumers of the API (in a RESTful way):

- Create appropriate APIs for checking the role of the user(i.e ADMIN, COMPANY, STUDENT) by checking its id and return its data to the client other than the password.
- Display status & response code in the response.
- The successful request should return 200 response codes.
- Use an appropriate HTTP method out of GET, POST, PUT, PATCH, DELETE, etc.

Story-6 (Create APIs for the students)

Your APIs should provide the following functionality to downstream consumers of the API (in a RESTful way):

- Create appropriate APIs for sending, deleting all the data that can be accessed only by a student with the help of id.
- Display status & response code in the response.
- The successful request should return 200 response codes.
- Use an appropriate HTTP method out of GET, POST, PUT, PATCH, DELETE, etc.

Solution

https://github.com/Gtgstg/Employee-Hiring