



## Office Reservation App:

In this project you will develop an office reservation app. This app should have CRUD operations for Organizations, Offices and Desks that are working with a Mongo database. Only a logged-in user can use these CRUD operations. In this app, you should register a user with an endpoint. You should develop a Jason Web Token (JWT) refresh and access token mechanism for handling user security. We expect you to implement this application with some features as detailed below:

### Features required to complete this project:

- Project should be implemented in NodeJs
- Project design is up to your creativity
- Project should be implemented with Typescript
- Use Schemas with Zod for validation
- Use NodeJs security with any library you prefer that enables refresh and access tokens to implement user login
- There will be roles for each user like (SUPER\_ADMIN, ADMIN , USER)
- SUPER\_ADMIN will be able to list all organizations, create organizations and offices.
- ADMIN should be able to create Users for **his or her** organization.
- There should be a Swagger UI documentation for each endpoint.
- There should be a cron job that deletes desk reservations at the end of each month
- You should deploy your application to AWS EC2 instance so it can be accessible from internet

### Useful Links:

- <https://expressjs.com/>
- NodeJs Express Tutorial:
  - ▶ Node.js and Express.js - Full Course
- NodeJs Crash Course:
  - ▶ Node.js Crash Course Tutorial #1 - Introduction & Setup
- Nodejs Typescript MongoDB Tutorial:
  - ▶ Build a REST API with Node.js, Express, TypeScript, MongoDB & Zod
- What is JWT?
  - ▶ What Is JWT and Why Should You Use JWT
- NoSql vs Sql
  - ▶ SQL vs NoSQL or MySQL vs MongoDB
- NodeJS Auth Tutorial
  - ▶ Build an Authentication API with Node.js, TypeScript, Typegoose, ExpressJ...



- Spring Boot Swagger UI Tutorial
  - ▶ Automatically Generate Swagger Docs With ExpressJS & NodeJS
- Spring Boot AWS EC2 Deployment
  - ▶ Deploy Node app on AWS EC2 Amazon Linux 2

## PROJECT DETAILS

Each Organization should have a offices and each office should have desks  
There should be create, update , delete , getList and getDetail endpoints for each Entity(Organization, Office, Desk,User).

These endpoints should be secure that can only be accessed with a valid access token and valid role.

Users should be able to reserve a desk with a specific time slot.

Each endpoint should be documented in Swagger-UI.

Each request or response should use a Zod Schema.

Your application should be accessible from the internet with a specific port number.