ASSIGNMENT FOR BACKEND DEVELOPER

# Instructions

1) Use NodeJs for Backend and MongoDB for Database.
2) Use Proper commenting and Naming conventions should be meaningful.
3) Reusable code functionality would be expected.
4) MVC Pattern should be followed.
5) ECMAScript 2015 version would be preferred.
6) The code should be efficient and follow best practices.
7) Always Use ‘mongoose’ instead of ‘mongodb’ npm package.

# Before Start Working on it,

1) Create a public GitHub repository.
2) Share the repository with us.
3) For Every concept completion should be pushed in GitHub with proper commit message. For bug fixing also, should have proper commit message.

# Assignment

We have to create Restful APIs for User Profile Management, with functionalities,

(Note : You can use your own flexibility in splitting or grouping APIs)

1. Add Admin
2. Update Admin
3. View Admin
4. Add User
5. Update User
6. View User

# Model

(Note 1 : Both Admin and User holds same model)

(Note 2 : All Admin and User details should be stored in one place in database)

(Note 3 : * Represents Mandatory Field)

(Note 4 : Naming Conventions are up to you.)

1. First Name*
2. Middle Name
3. Last Name*
4. Email*
5. Password (Encrypted)*
6. Role (Admin or User)*
7. Department
8. Created Time*
9. Updated Time*

# Add Admin or User

1. All Mandatory fields should be present.
2. Password should have 6 to 12 characters.
3. Confirm password should also be present in adding User/Admin.
4. Created Time and Updated Time should be auto-generated.
5. Admin can add Admin/User. User can add user only.

# Update Admin or User

1. Admin/User can edit any field. If 1 field given means, that particular field should update.
2. Admin/User cannot edit password (Once created password, it remains same).
3. Created Time and Updated Time should be auto-updated and it cannot be manually edit by any User/Admin.
4. An Admin can edit Admins/Users and himself. A User can edit users and himself.

# View Admin or User

1. Admin/User can view any field. If 1 field given means, that particular field should view. If nothing mentioned means, all details should display.
2. Password should not be display for any User/Admin.
3. An Admin can view Admins/Users and himself. A User can view Users and himself.

# Conditions

1. Without Authentication no one can view, add or edit any users.
2. JWT Concepts should followed.
3. All APIs should be protected with Cross Origin Restriction.

# Additional (Unrelated to current Assignment)

1. Try to Understand the basic concept of ‘MQTT’ Protocol.

# Greetings

Hope you understand the document thoroughly.
See you soon. All the Best.
Thank you.