



# Blockchain Research Lab

## Assignment 3

Deadline : 14 Nov 2022

Total Marks : 400

### Problem Statement :

Make the assigned project.

| Trainee Name       | Project                |
|--------------------|------------------------|
| Aditya Raj Sharma  | Authentication backend |
| Prakhar Maheshwari | Authentication backend |
| Gracy Gupta        | Authentication backend |
| Simran             | Authentication backend |
| Harsh Saxsena      | Authentication backend |
| Jaswant Kushwaha   | Authentication backend |
| Jatin Shankar      | Authentication backend |
| Yash               | Authentication backend |

### Project Description :

- Authentication System:**
  - A user collection should atleast contain following fields :
    - email
    - nickname
    - password : *hashed*
    - role : enum( [“user”, “admin” ] ) , default: user
    - last\_logged\_in
  - Following APIs (routes) must atleast be present :
    - /signup (method : POST) :**
      - method :** POST
      - description :** for signup
    - /login (method : POST) :**
      - method :** POST
      - description :** for login; should return an authorization token.
    - /password/reset**
      - method :** POST
      - body :** { email }
      - description :** It should send an email with password reset link of format “*https://hostname/password/reset/<token>*”
    - /password/reset/:token**
      - method :** POST
      - params :** token
      - body :** { new\_password }
      - description :** If the token is verified then it should reset the password
    - user/nickname**
      - method :** GET
      - accessible only to the logged in user**
      - fetches nickname of user
    - user/nickname**
      - method :** POST
      - accessible only to the logged in user**
      - description :** sets new nickname
    - admin/delete/:email**
      - method :** GET
      - params :** email
      - accessible only to the admin**
      - description :** deletes the user
    - admin/make\_admin/:email**
      - method :** GET
      - params :** email
      - accessible only to the admin**
      - description :** makes any user admin

- You must use middlewares.
- Bonus++ marks for implementing authentication in your preveious project.
- Bonus marks for integrating the APIs (routes) with a frontend.
- Bonus marks for making an API documentation.
- Bonus marks for implementing validations.
- Use NodeJS, express, mongoDB, mongoose, jwt, nodemailer, passport (optional)
- Follow REST apis format.