ALGO8 Backend Developer Assignment

- `Task 1` : Provide some example of config file separation for dev and prod environments.
- `Task 2` : Convert any callback into a promise with example code of callback and promise.
- `Task 3` : Pick any two projects out of following
    - `3A` : Show Standard techniques of authentication OAuth, JWT with refresh token implementation.
    - `3B` : Chat App with Socket IO with one-to-one and group chat. Notification on user joining, leaving, typing.
    - `3C` : Alert user by email and on simple frontend when certain metric from database cross a value using WebSocket (use of socket IO not allowed). Metric = Doc of type = 'error' > 50 in last minute. To add doc of type error use any method that suffice.
    - `3D` : Create Excel from Mongo Database Collection.

Note: Your code will be evaluated based on best practices, linting, error handling and structure of projects.

Note: Assignment submitted with copy paste of existing npm package or git repo will be rejected. But use of google search and copying useful information is encouraged.

Nice to haves

- Proper logging can earn brownie points.
- Documentation
- Proper git commit history. Create small git commits

`Task 3A`

You have to show standard techniques for authentication like OAuth, JWT with refresh token implementation. You are free to use any server (Express or other) or database of your choice. But it must be complete project with APIs like login, register.

Note: Commit your code to public repo at github or gitlab and deploy your project on platform of your choice (Heroku, AWS, GCP).

Note: Frontend is optional, but provide documentation for API routes.

`Task 3B`

Design a chat App with Socket IO with following functionalities - one-to-one between two users - public group chat. - anyone can join a group - anyone can create a group - Notification on user joining, leaving, typing in one-to-one and group chat.

Note: Commit your code to public repo at github or gitlab and deploy your project on platform of your choice (Heroku, AWS, GCP).

Note: Frontend is optional, but provide documentation for socket events.

`Task 3C`

Alert user by email and on simple frontend when certain metric from database cross a value using WebSocket. Metric could be of your choice. - Example 1: Orders are placed throughout the day on a e-commerce website. As a backend developer you are assigned a task to alert/notify some user whenever total count of orders in last minute crosses 10 orders. - Example 2: Orders are placed throughout the day on a e-commerce website. As a backend developer you are assigned a task to alert/notify some user whenever sum of order amount in last minute crosses 10000 rupees.

Note: To add orders(order=time,amount,productName) to database, you are free to use any method of choice. But make it sub-minute routine to quickly test the application.

Note: You can use any DB of choice (Mongo, SQL, Firestore).

Note: Use of socket.IO not allowed, use ws npm package or other similar package.

Note: The final url of the project must be `ws` or `wss`

Note: Commit your code to public repo at github or gitlab and deploy your project on platform of your choice (Heroku, AWS, GCP).

Note: Frontend is required for this project, but a very simple UI would suffice.

`Task 3D`

- create a database with dummy data
    - db item = { type:'', ...rest of data}
    - type can be one of following: `A | B | C | D`
- query db based on type(`A | B | C | D` or `any`) and size
- convert that data into CSV or Excel file
- upload to storage - S3, GCP Storage or any other of your choice
- return a publicly accessible url

Note: Commit your code to public repo at github or gitlab and deploy your project on platform of your choice (Heroku, AWS, GCP).

Note: Frontend is optional, but provide the API for getting url back.