Exam Portal

Auth levels

1. **admin**

    - can include and remove any teacher from the examination portal at will.
    - can also update the details regarding any teacher, which can potentially affect their activities or/and their stance, as a tier of authority.
    - can include and remove any subject on which an exam may be conducted by the teachers.
    - admin has the final say in regard to any subject upon which an exam may be conducted. however, teacher is granted full authority as to the actual conductance of the exam.

2. **teacher**

    - has authority over the number and strength of the questions that would be included in the test paper.
    - can include, remove and update the questions from the test paper, wihtout restriction, given that the test is still in the creation stage.
    - teacher has root control on the start and end time pertaining to every single test, conducted by them, which would be terminated automatically after a set time duration, if not terminated directly by the teacher themselves.
    - can view the complete stats (under construction), as per the test as a standalone event, which sports every single detail and nuance of the test.
    - they have the authority to send the test link upon creation of the said test.

3. **student**

    - can access the test, posted at any given time, provided that teacher has mailed them the unique test ID.

Tech Stack :

1. **Nodejs** : Premium server supported with nodejs runtime, built, capable of handling potentially huge number of requests in real time, with latency, virtually, non-existent.
2. **Expressjs** : A web application framework, which provides the intricate middleware, integral to the layered functionalities, hosted in the application.
3. **React** : Javascript library, to construct the user interface fragments, configured to provide optimum control flow to any end-user.
4. **MongoDB**: Real time database management operations, with hosted mongodb cluster, to enable operations pertaining to access and post events of data (identity and marks information when applicable).

Applications :

- this functional exam portal, can be implemented by any standard, exam conducting body, being able to adapt to the specific needs (should they exist), backed by the varied functionalities it is bundled with.

Steps to initiate the application

1. **Get a live and running instance of MongoDB**

    1. Go to mongodb.com
    2. create an account
    3. create a database cluster

    4. you will get a connection string for that cluster
    5. paste that connection string as it is in the `backend/config/default.json` under the `mongodb: { connectionString: "<paste token here>" }`

2. **Configure Sending Emails**
    1. Enter your email credentials in the `backend/config/default.json` file under the key of `mail-credentials` where userid is your email address and password is generated by email console
        1. Your google account needs to have 2FA on that mail
        2. Go to security ==> "Signing in to Google" section ==> App Passwords ==> Generate a password ==> Copy the generated password and paste in the password field (of your app)
    2. Add your email in the `backend/services/mail.js` file (line number 16), which says <\EMAIL_HERE>, replace it with your email
    3. You can also change the name by which you send emails, by replacing the "Exam Portal" in line 16 of `backend/services/mail.js` by the name of your choice

3. **Create the admin in the app**
    1. Give the admin credentials in the `backend/services/tools.js` file (line 10 - 14). Credentials include name, emailid, contact and password (in line 7)
    2. uncomment the line number 117 in file `backend/app.js` and start the app using the script `yarn start` inside the `backend` directory
    3. Once it shows the message "User Created", stop the app, using the keyboard shortcut `Ctrl + C` or `Cmd + C` or kill the running terminal
    4. Comment the line number 117 in file `backend/app.js` as it was earlier, then start the server again (`yarn start` in the backend directory)