**Assignment: Tasks App Continued**

1. In the main part of the Application finish authentication flow with Access and Refresh tokens
2. In the main part of the App also create the “Forgot Password” flow.
3. Create an additional part of the application using React, Express, and Socket.IO that allows users to chat when they’re logged in.
4. The WebSocketServer by socket.io should be added to the existing server (JS)
5. The first and the second part of the frontend of the app will be implemented by Microfrontends Architecture, as the first part is coded in JS and the second in TS.
6. Start creating the second part without the “logged in condition” until we create together in the class the example of Microfrontends.

CHALLENGE(because of Microfrontends): the logged in users could also change Tasks and see other users changing the Tasks – in real time. Pay attention that the users should see and change only the Tasks of their GROUP, it means that we should add a Group model  
to DB or GROUP property to the User Model. Create CHAT option – the button that says “Click if you want to join the group…” Invitaion by EMAIL is a great option too.

Details:

1. In the main part of the Application finish authentication flow with Access and Refresh tokens:
   1. login at the server: create AccessToken and RefreshToken  
                                     save RefreshToken in DB  
                                     send the both tokens to FRONTEND (you can send  
                                     them both in the body or AccessToken in the body, and  
                                     RefreshToken as a cookie)
   2. at the FRONTEND we can keep AccessToken in memory, in session storage and in local storage and the same with Refresh Token, but if it comes as a cookie, it stays a cookie
   3. at the BACKEND we add to the routes that need AUTHORIZATION we add middleware of verify() - which checks AccessToken  
                                                      and, if we decided so - refreshes RefreshToken
   4. at the FRONTEND: when we should get data from BACKEND then needs AUTHORIZATION (or perform action that needs AUTHORIZATION) we send   
      request with header Authorization: ‘Bearer <AcccessToken>  
            and  if the AccessToken is expired and we did not send the tokens   
            together - we send now the RefreshToken.  
            The backend should refresh the both tokens, or, if RefreshToken is   
            expired too - LOGOUT

1. In the main part of the App also create the “Forgot Password” flow
   1. FRONTEND: we should be able to get from “login” page to “forgot password” page
   2. FRONTEND: “Forgot Password” page - fill email and “submit” (send to BACKEND request to reset the password)
   3. BACKEND function forgotPassword:   
                   – gets email from the request body  
                   – checks if the user exists  
                   – if exists we create resetToken with short expire time  
                   –  saves the token in DB, and send email with link to the ResetPage  
                       which contains (link contains) the ResetToken and UserId
   4. FRONTEND: the user gets message that the email was sent
   5. USER clicks on the link in the email and gets to RESET PASSWORD page
   6. FRONTEND Reset Password page: – user fills twice the same password  
                                                                  – clicks Submit  
                                                                  – the page sends request to change   
                                                                     password to the server, and the   
                                                                     request contains: password, userId  
                                                                     and the resetToken
   7. BACKEND ResetPassword function: – check resetToken and that it is not   
                                                                       expired  
                                                                    – patch the new password and delete   
                                                                       the token  
                                                                   – send “SUCCESS” to FRONTEND
   8. FRONTEND gets “SUCCESS”, shows “SUCCESS” message and the LOGIN page