User Functionalities

1. Student
   * Login / Register
   * Test Registration
   * Give test
   * Check result
2. Teacher
   * Create, Update, Conduct Test
   * Declare Result
   * Create, Update, delete Questions
3. Admin
   * Create, Update, Remove Teacher user
   * Create, Update Remove Subject

Test Scenario

* Student accounts are created by Register
* Teacher accounts are created by Admin
* Subjects are added by Admin
* Teacher creates question bank for subject
* Teacher creates test, selects subject, maximum marks, types of question (1,2,3,4 marks), test start and end time, duration, registration time and end registration time, result time
* In backend create question paper
* Start registration form, test id for registration form
* Students fills registration forms -> generate empty answersheet
* Students go to test page, enter test id and start test
* Answers are stored and checked at end of test
* Test can be ended at timer or student ends test
* At end of test generate result
* Teacher can declare result at end of test or at decided time
* Students can check result of given tests

Frontend

1. Understand data flow and code flow in react
2. Get login/ logout page done for user
3. How to create two ports in single react app
4. Create login/logout for admin
5. Register page for student

Data Base

Reference Database

In excel sheet <database.xlsx>

Backend

Implementation details

Completed

* Created schema and model as per database v1
* Database connection to mongodb
* Public routes to student registration
* Student register service
* Login routes to userLogin
* Login service to userLogin and userDetails
* Login service to userLogin uses passport authentication
* Admin routes
* Admin service to register, remove and unblock user
* Admin service to get admin details from jwt token
* User routes to get userDetails from login service
* Admin service and routes to add, remove and unblock Subject
* Add question and answer

To do

* Jwt token logout in backend