Project Name - Live test taking system

(you can also give it a name of your choice)

Deadline - 16 Days

A Multiple choice test taking system in which tests are tracked live and analytics are generated based on that.

Project Description -
This project should be a ready to deploy test taking system. It must have all the features mentioned below and it must be deployed on a server before submission. It must be a single page application.

Frontend Technologies allowed - AngularJS for dynamism and SPA and any technologies/framework allowed for frontend design.

Backend Technologies allowed - NodeJs, ExpressJS, MongoDB, SocketIO, WebSockets.

Features of the platform -

1) **User management system**
2) **User Test management system**
3) **User Test taking system**
4) **Test listing admin**
5) **User analytics in admin panel**

1) **User Management System -**
    a) User should be able to sign up using email, gmail and facebook.
    b) User should be able to login to the system through email and password combination or using Gmail/Facebook
    c) Forgot password functionality should be there to reset password.

2) **User Testing management system -**
    a) Once the user logs into the system, he should see a dashboard containing the statistics of all tests he has taken. The statistics may include the

number of tests taken, average score and percentage growth etc. You are free to think and use more metrics, graphs etc.

b) Dashboard should also contain the lists of tests the user has taken and every item in this list should be clickable. On clicking this item, a Test Result view should open which contains the details of test result.
c) There should be a “take a test” option in menu from which user can go to test taking page.
d) On test taking page, user should see a list of tests he can appear for along with a button to start that test.

3) User test taking system -

a) Once user starts the test, he should first see an instructions screen containing. It may also contain the rules of the test.
b) Once the user reads the instructions and accepts the rules (single accept button), The test timer will start and the screen should display the test questions and options associated with it.
c) User should be able to choose only one option as answer for every question.
d) The test should have a time limit. The test window must automatically close once the timeout occurs irrespective of how many questions have been answered. The system should submit the answers automatically.
e) If the user completes the test before the time ends, he should see a submit window which will submit his all answers. In case of timeouts, this window must appear automatically.
f) The system must keep a track of how much time a user is taking for answering each question. Your models should be designed accordingly.
Hint - questions and their options should be stored in one model which is not user dependent, whereas the answers should be stored with both testId and user details because the answers are related to users.
g) You may use socketio, ws or similar modules for live tracking the user progress.
h) On submission of test, show the result to student. Tell him the number of correct answers and percentage of marks obtained.

4) Test listing Admin

a) Admin should be able to create tests in the system
b) Each test should have a set of questions, each question containing at least 4 options and overall time limit of the test.

c) Admin should be able to create, edit, delete and view any tests, question or option.
d) While creating options for any question, admin should be able to set a correct answer. This answer (flag) will actually help in automating the test evaluation process.

5) **User analytics in admin**
   a) Admin should be able to view details of users registered in the system
   b) Admin should be able to view overall performance of the user in all his tests.

A few important points -

1) Run the APIs in POSTMAN to test them properly.
2) Backend should follow MVC format and should have properly defined middlewares and libraries.

Evaluation Basis

This project will be evaluated on following basis -

1) **Quality of JavaScript code -** Your application's Javascript code should be optimized to be readable with proper indentation and comments. It should be broken down into functions for better maintainability and it should not contain any logical bugs.

2) **Intuitive Thinking -** You have thinking intuitively and make the platform as easy to understand as possible. You have think about all the possible error cases and you have to handle them by giving alert messages to user.

3) **Originality of code -** Your code will be checked for plagiarism and if it's not original, it will be discarded with a negative skill score.

4) **Quality of frontend application -** Your frontend application should look good in terms of UI and UX.

5) **Quality of Backend application** - You have already been trained about how to code optimal REST apis using ExpressJS. You must follow the same guidelines in your application to get better skill score.

Deliverables from Candidate

1) A TXT file containing the description of your project and all your assumptions. It should also describe the features of the project and also any extra features that you have coded to get extra marks for intuitive thinking. It should also contain your registered name, email id and mobile number.
2) Github repository link of this project mentioned in the TXT file
3) Url of server where the application has been hosted should be mentioned in the TXT file.

Always remember these evaluation basis, as you do have a deadline. And your aim is to meet the deadline.

Warning - Do not submit incomplete projects or projects that are not running. They will result in negative skill score. Always use support system in case of queries.