Project Name - Meeting Planner

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

Deadline - 15 Days

Project Description -

This project should be a ready to deploy meetings scheduling system. It must have all the features mentioned below and it must be deployed on a server before submission.There should be two separate parts of the application. A Frontend developed and deployed using the technologies mentioned below and a REST API (with realtime functionalities) created using the technologies mentioned below.

Frontend Technologies allowed - HTML5, CSS3, JS, Bootstrap and Angular
Backend Technologies allowed - NodeJS, ExpressJS and Socket.IO
Database Allowed - MongoDB and Redis

Features of the Application -

1) **User management System -**
    a) Signup - User should be able to sign up on the platform providing all details like FirstName, LastName, Email and Mobile number. Country code for mobile number (like 91 for India) should also be stored. You may find the country code data on these links (http://country.io/phone.json,http://country.io/names.json)

    b) Login - user should be able to login using the credentials provided at signup.

    c) Forgot password - User should be able to recover password using a link or code on email. You may use Nodemailer to send emails. (Please use a dummy gmail account, not your real account).

2) **User Authorization system -**
    a) User can be of two roles, normal and admin. Admin should be identified with a username ending with "admin", like "alex-admin" is an admin, since it ends with "admin".

3) **User Slots management system (Flow for normal User) -**
    a) Upon login, normal User should be taken to a dashboard showing his current months', planned meetings, in the form of a calendar. Current day-cell should be selected by default.
    b) User should be able to only view his meeting slots and he should not be able to make any changes

Hint - you may use calendar modules of Angular to design the Interface. One such module which you can use for calendar UI is Angular-Calendar.

4) **User Slots management system (Flow for Admin User)-**
    a) Upon login, admin User should be taken to a dashboard, showing all normal users in a list format
    b) Upon clicking on any user, admin should be taken to user's current calendar, with current date selected, by default.
    c) Admin should be able to add/delete/update meetings on any day, by clicking on a appropriate day-cell/timeline.
    d) These details should be stored in database for every user.

5) **User Alerts management system**
    a) Normal User should also be notified in real time, though an alert if he is online, and email (irrespective of whether he is online or offline), when
        i) A meeting is created by admin
        ii) A meeting is changed by admin
        iii) 1 minute before meeting, with an option to snooze or dismiss. If snoozed, alert should come again in 5 seconds, if snoozed again, it should re-appear in next 5 seconds and so on. Once dismissed, alert should no longer appear.

6) **Planner Views**
    a) Similar to Google or outlook calendars.
    b) The view must follow the following guidelines -
        i) Planner should show only current year, past and future years to be ignored.
        ii) User should be able to change months, through an arrow button(or prev/next button), each month should show all the dates in tabular format, like in actual calendar.
        iii) Day Cells should be filled, if any meeting is kept, with a some design. There should also be a design for overlapping meetings.

iv) Upon click the day's cell, a view should pop, showing all meetings, along a 24 hr timeline, with the slots covering the exact duration of each meeting.
v) Upon clicking on a meeting, its details should pop up in another view

c) **Admin Flow**:
i) For admin, a create button should be there in calendar view, to create a meeting.
ii) Upon clicking on create button, details view should open.
iii) Once created, it should appear on the calendar view.
iv) Upon clicking on an already created meeting, same details view should open.
v) Details view should be a form
vi) Admin should be able to make changes in meeting details form, and submit.
vii) Admin should be able to delete a meeting as well, with another button
viii) Meeting details, should cover when, where and purpose. Also, by default username of the admin, who kept the meeting, should also show in non-editable format.

7) **Error Views and messages** - You have to handle each major error response (like 404 or 500) with a different page. Also, all kind of errors, exceptions and messages should be handled properly on frontend. The user should be aware all the time on frontend about what is happening in the system.

A few important conditions-

1) Design of the application must be done entirely by you from scratch using Bootstrap. You are not allowed to use templates that are already available on internet. Use of such templates will be considered plagiarism and your submission will be rejected.

2) You must follow all the industry practices taught to you in the training. You have to structure your application in the same way as taught in your training and write all the components,services,middlewares, libraries etc yourself.

3) Rate limiting - Your APIs should have pagination or rate limiting to avoid send huge chunks of messages as API response.

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. It should use modern javascript as much as possible.

2) **Responsive Design -**Your website should be full responsive. It will be checked thoroughly for all the things that have been taught to you in the level.

3) **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. You must use elements like progress bars and loaders to handle the UX better.

4) **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.

5) **Quality of Frontend application(Angular application) -** All the best practices associated with Angular must be followed.

6) **Quality of Backend application -** All the best practices associated with Advance rest API development must be followed.

7) **Documentation** - Documentation should be done properly as taught to you in training. Documentations of REST API endpoints and Socket Endpoints should be separate.

Deliverables from Candidate

1) Project description - A TXT/Doc 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.
2) Github repository link of this project should be mentioned in the TXT file.
3) You have to host the built versions of the applications on AWS and mention the URL of that application in the TXT file. URL of both Frontend and REST API must be mentioned along with documentation link(if any).

Always remember these evaluation basis, and your 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. Do not seek help from your peer learners or anyone else. Your code will be checked for plagiarism at multiple stages and if you are found to be guilty of cheating, your submission will be discarded.