Project Name - Ticket Based Support system using MEAN Stack

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

A generic ticket based support system for resolving queries on a platform

Project Description -

Support is an essential feature for any platform, and dedicated support is best approach in case you are really concerned about the user experience of your platform.

The Aim of the project is to create an online ticket based support system, just like the one present on edwisor.com which should be usable by any kind of platform to get support queries from their users and resolve them.

This project should be a Single Page application with separate backend and frontend. The backend should have REST APIs with proper documentation. Take documentation from your previous project as reference.
You are allowed to use any kind of angular module, libraries or tool you want. APIs have to be developed by you.

Frontend Technologies allowed - HTML 5, CSS, Javascript , Jquery and AngularJS

Backend Technologies allowed - NodeJs, ExpressJS, MongoDB.

You are free to use any libraries/modules in backend and frontend both.

Features of the platform -

1) **Ticket Raising panel - User facing**
2) **Ticket Resolution panel - Admin**

1) ## Ticket Raising panel - user end

a) A view to login and Signup.
b) A View to create a ticket. Get all the necessary information like name, email, phone number, Query title and Query details. **File upload is optional.**
c) A View to view all queries raised by the person.
d) A view to show the details of a particular query. It should include the original question as well as the answer from admin and person in form of a conversation (chat like UI). This view should also have the option to set the status of the ticket to ‘open’ or ‘closed’ depending on whether the query is resolved or not.

2) Ticket Resolution Panel - Admin end

a) View to Display ticket by status - This should list all tickets received by the support system. There should be a drop down menu to filter through the status of ticket. Ticket can be of status ‘open’ or ‘closed’ depending on whether the query is resolved or not.
b) A view to show the details of a particular query. It should include the original question as well as the answer from admin and person in form of a conversation (chat like UI). This view should also have the option to set the status of the ticket to ‘open’ or ‘closed’ depending on whether the query is resolved or not. The answer created here

Additional Features -

1) On Status change of ticket, the person should receive an email notification.
2) When the person receives the answer or the admin receives the reply, an email notification should be sent to the person concerned.
3) For the sake of simplicity, treat the Admin as a user of the system. Don’t create special backend for admin.

A few important points -

1) Run the APIs in POSTMAN once to see the response format. That will enable you to easily use that in your Angular code.
2) The frontend should be single page with well defined View, controllers, directives and services.
3) Backend should follow MVC format and should have properly defined middlewares and libraries. Authentication should be done using JWT.
3) Follow modern design guidelines while creating the view.
4 Admin can be less intuitive, but try to make user experience as intuitive as you can

---

Evaluation Basis

This project will be evaluated on following basis -

1) **Design and UI of the project** - Your website should amazing in terms of colors and placement of objects. Being Simple yet elegant is the key here. Research over internet in case you don't know the kind of color choices you should make.

2) **Responsiveness of the website -** You will be given major credits for responsiveness. A non-responsive website will not be accepted as correct answer. You are free to use custom css and frameworks (like bootstrap) to ensure responsiveness.

3) 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.

4) **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.

5) **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.

6) **Quality of Angular application -** You have already been trained about optimal practices to create an angular application. You must follow them in your application to get better skill score.

7) **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) Compressed Folder containing your code.

2) 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.

3) Github repository link of this project mentioned in the TXT file

Create a folder containing all three deliverables. Compressed all these into a ZIP/RAR format.

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.