**TBS**

Steps to run it in local enviornment!

1. npm install
2. Start a mongo server in local env
3. Edit 'server/config/adminRightsTo.js' - to define who gets admin access. Read assumption below for details.
4. Edit 'server/confilg/email-cred.js' - to be able to send email notifications through app
5. node app.js

**Ticket Based Support System!**

A SPA built to be used by any 'product owner' to resolve issues being faced by the product users.

Built to have a team of admins being able to work together in resolving tickets.

\*CHAT UI to see communication between user & admins on every ticket \*Admins can involve each other as required in a ticket.

**Assumptions -**

1. ADMIN team will have a unique domain for e-mail addresses.

For example:

If 'Edwisor' team uses this product to resolve queries of people enrolled in their program.

The Edwisor team members will get the ADMIN RIGHTS as they will register with email accounts with domain '@edwisor.com'

In 'server/config/adminRightsTo.js' we have {prodUser : 'edwisor.com'}; which can be changed to anything as per the client's domain and admin rights will be granted to them.

So, if we have email addresses like '[a@xyz.com](mailto:a@xyz.com)' and '[b@xyz.com](mailto:b@xyz.com)' and want them to have ADMIN rights.

We set {prodUser : 'xyz.com'}

**Extra Features -**

1. A Ticket can be reopened by Admins or the user to whom that ticket belongs to.
2. User can rate his experience after the ticket gets closed
3. Admins get to 'involve' other admins if needed in Ticket resolution. An email gets sent to involved Admin(s) as notification and ticket gets added to their 'assigned tickets' list as well.
4. An admin gets a view which lists TICKETS assigned to him. Ticket get asigned as soon as ADMIN comments on a ticket or another admin chooses to 'involve' him
5. A closing and reopening comment gets entered so, it can be tracked who changed status of a ticket.