Project Name - Expense Splitter

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

Deadline - 15 Days

Project Description

This application will help in splitting of amount between a group of people.
Let's say a group of 5 people go to a restaurant. One person paid the bill and he can split the bill among those 5 people through this application.

Take Splitwise (you can find it on web and also on android) as a reference application.

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

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).

Expense Management System -

Group Management-

1. Users should be able to create groups and add different people into it.
2. When a user is added to a group he can see that group name in his dashboard.
3. User can go into the group and see all the expenses added in the group.
4. People should not able to exit a group.

Expense -

1. Users can go to a group and create an expense.
2. Expense should have an amount field, name of the expense, who are the people involved, who paid, when it was created, updated or deleted etc.
3. User can select only the people available in the group to create an expense in a group. They cannot select people from a different group.
4. Anybody in the group should be able to add, edit or delete any expense. Editing means they can change the amount, who paid, who are people involved etc.

Expense History -

1. Whenever an expense is added or updated you should create a history of what has happened to the expense.
2. From the expense view user should see the history of the expense as well.
3. For each history it should say which person did that action.
    a. E.g Rahul create this expense.
    b. Manan updated the amount from 300 to 400
    c. Amir was added the expense
    d. Shakti was removed from the expense etc

Amount Pending Functionality -

1. This should show for a specific group which person should get money and who should pay for it.
    a. E.g From Rahul, Manan and Amir, Rahul paid Rs. 300 for a restaurant bill. So Amir and Manan should pay Rs. 100 each to Rahul.
    b. Suppose another person Shakti was added to the expense. Now the amount each person should pay is Rs. 75.
    c. So Rahul will get Rs. 225 from Amir, Manan and Shakti.
    d. Now another expense is added of Rs. 100 and its paid by Amir. So Rahul, Manan and Shakti should pay Rs. 25 to Amir.
    e. So Amir has to pay (75 - 25 = 50) to Rahul. Shakti will pay Rs. 75 to rahul and Rs. 25 to Amir and same for Manan.

Realtime -

1. For each add or update or delete of an expense people involved in the expense should be notified through email and browser notification if a user is online.

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.

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.