Project Name: MEETING PLANNER

1. **Source code:**
    - Repository: https://github.com/rinkeshpatel22/Mean-stack-app1-meeting-planner
    - git: https://github.com/rinkeshpatel22/Mean-stack-app1-meeting-planner.git

2. **Deployed url:**
    - Front-end: http://rinkesh.co.nf/MeetingPlanner
    - Back-end API docs: http://rinkesh.co.nf/MeetingPlannerAPIdocs

3. **Overview:** An application that allows us to register as an admin and plan a meeting with other users who are registered as a normal user with track of meeting schedule via email and online notifications.

4. **Technologies used:**
    - Front-end:- Angular 7, HTML, CSS, JS, Bootstrap.
    - Back-end:- NodeJS, ExpressJS, Socket.io
    - Database: MongoDB

5. **Features of the application:**
    - Users can register themselves as an admin or normal user.
    - Admin:
        - On login, Admin will be redirected to admin-dashboard.
        - On admin-dashboard, admin can view list of all normal users.
        - On click of any user from the list, admin can view previously scheduled meetings with that user in calendar.
        - Admin can create a new meeting with that selected user.
        - Admin can also update and delete meetings.
    - Normal User:
        - On login, the user will be redirected to user-dashboard.
        - On user-dashboard, a calendar with current day selected will be displayed.
        - User can view his/her meeting list by clicking of day-cell of the calendar.
        - On click of any meeting from the list of meetings, a popup will be displayed with the details of the meeting.
    - Notifications:
        - On meeting create, update and delete an admin and normal user will get notification via email.
        - If normal user is online than the user will get online notification when any meeting created, updated or deleted.

    - 1 minute before meeting, admin and normal user will get reminder with snooze and dismiss option. If sneezed, alert will re-appear in next 5 second again and again until it dismissed or meeting start time passed.

6. **Main modules of the application:**
    - User management
        - Signup
        - Login
        - Forgot password
        - Reset password
        - Admin flow
        - Normal user flow
    - Meeting management
        - Create
        - Update
        - Delete
        - Get meetings by inviter – (get only those meetings created by specified admin)
        - Get meetings by invitee – (get only those meetings created for specified normal user)
        - Get meetings by inviter and Invitee– (get only those meetings created by specified admin and for specified normal user)
    - Email management
        - Send welcome email on signup.
        - Send password reset link when requested
        - Send password change notification
        - Send Meeting create notification.
        - Send Meeting update/delete notification.
    - Socket.io management
        - Establish socket connection between front-end and back-end
        - On meeting create, update or delete send online notification to the user who is the invitee of the meeting.
    - Authorization
        - Create authToken in backend and send it in successful login response to front-end.
        - Front-end pass authToken in every authorized api call
        - API verify the authToken before responding the request, if authToken is correct and not expired then response successfully otherwise reject the request.

7. **Back-end code folder structure**
   - **<u>App</u>**
     - **Controllers**
       1. meetingController.js
          - ✓ Functions for create, update, delete and get meeting.
       2. userController.js
          - ✓ Functions for signup, login, change password etc.
     - **Libs**
       1. checkLib.js
          - ✓ Library to check null or empty value
       2. emailLib.js
          - ✓ Library to send email notifications
       3. loggerLib.js
          - ✓ Library to log errors and info while executing the application.
       4. passwordLib.js
          - ✓ Library to compare password
       5. responseLib.js
          - ✓ library to generate api response in a unique pattern
       6. socketLib.js
          - ✓ library for online meeting updates and notifications via socket.io
       7. tokenLib.js
          - ✓ Library to manage authorization token
       8. validationLib.js
          - ✓ Library to validate request parameters
     - **Middlewares**
       1. auth.js
          - ✓ Library to check the request whether it is authorized or not by verifying the authToken.
     - **Models**
       1. auth.js
          - ✓ Database schema for Auth table
       2. meeting.js
          - ✓ Database schema for Meeting table
       3. user.js
          - ✓ Database schema for User table
     - **Routes**
       1. Meeting.js
          - ✓ API request routes for meeting create/update/delete/read.
       2. User.js

        - ✓ API request routes for user signup, login, change password etc.
- **Config**
    1. appConfig.js
        - ✓ Configurations of the application like server port, database connection url, api version etc.

8. **Front-end modules (lazy loaded modules)**
    - **App module**
        - **User-management module**
            1. signup component
            2. login component
            3. forgot-password component
            4. password-reset component
        - **Admin module**
            1. admin-dashboard component
        - **User module**
            1. user-dashboard component
        - **Shared module**
            1. calendar component (shared for both admin and normal user)
            2. meeting component (shared for both admin and normal user)
            3. header component (shared throughout the application)
            4. loader component (shared for both admin and normal user)

9. **Front-end code folder structure**
    - **app**
        - **admin**
            1. admin component
                - ✓ display and manage admin dash-board
        - **services**
            1. app service
                - ✓ Functions of all API calls
            2. error-interceptor service
                - ✓ Intercept API response if got error and handle major error (404, 500).
            3. socket service
                - ✓ Functions to manage socket connection, emit and listen live notifications.
        - **Shared**
            1. Calendar component
                - ✓ Display calendar and manage user events.
            2. Header component

        - ✓ Display header with logout link, handle logout function.
    3. Loader
        - ✓ Display loader while the front-end page is awaiting for the response from API.
    4. Meeting component
        - ✓ Display meeting details in form, for admin it provides functionalities to create, update and delete meetings.
- **User**
    1. User component
        - ✓ Display and manage normal user dashboard.

- **User-management**
    1. signup component
        - ✓ display, manage and validate signup form
    2. login component
        - ✓ display, manage and validate login form
    3. forgot-password component
        - ✓ display and manage forgot-password form
    4. password-reset component
        - ✓ display and manage password-reset form

- **assets**
    1. countires.json
        - ✓ contains list of countries with codes
    2. countryCodes.json
        - ✓ contains list of phone codes with country codes