# MEETING NEST (Meeting Scheduling System )

**Project Description**

This application is a meeting scheduling system. The meeting can only be scheduled by the admin and the update is sent to the user via an email .

Since it is a MEAN Stack application,there are two separate parts of the project.

1. FRONTEND APPLICATION
2. BACKEND APPLICATION

* The frontend part is developed using Angular,Bootstrap,CSS,HTML 5,JS
* The backend part(REST API with real time functionalities) is developed using Node JS,Express JS
* Nodemailer: It is a module for Node js applications to send emails.
* Database: MongoDB and Redis Server

APPLICATION MODULES :

1. Dashboard Module
2. User Module
3. Meeting Module
4. Shared Module
5. Page-not-found Module
6. Server-error Module

Each module contains components based on the functionalities of the application.

COMPONENTS :

1. DASHBOARD MODULE

User-dashboard Admin- dashboard

1. USER MODULE

Login Sign up Reset-password Verify-email Forgot-password

1. MEETING

Create Meeting Update Meeting

1. SHARED MODULES

User-details my-nav

1. SERVER-ERROR

Server-error

1. PAGE-NOT-FOUND-ERROR

Page-not-found

**Features of the Application**

1) User management System –

a) Signup - User signs up on the platform providing all details like FirstName, LastName, Email and Mobile number. Country code for mobile number (like 91 for India) .

b) Login - user logins using the credentials provided at signup.

c) Forgot password - User recovers password using a link or code on email.

Nodemailer is used to send emails. (Dummy gmail account is used enabling less secure apps option)

2) User Authorization system

1. User has two roles:

Normal and Admin.

Admin is identified with a username ending with "admin".

In this application it is “rohit-admin”.

3) User Slots management system (Flow for normal User) –

a) Once a normal user logins, he/she is taken to a dashboard showing the current months planned meetings in the form of a calendar.

Current day-cell is selected by default.

1. User is able to view the meeting slots related to him/her only.
2. A normal user cannot make any changes in the dashboard.

Angular-calendar is used to design the calendar UI.

4) User Slots management system (Flow for Admin User)-

a) Upon login admin is taken to a dashboard that shows all normal users in a list format

b) Upon clicking on any user, admin sees user's current calendar with current date selected by default.

c) Admin can 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

1. Normal User gets notified in real time via an email (irrespective of whether he is online or offline)

Notifications are sent when

1. A meeting is created by admin
2. ii) A meeting is changed by admin
3. iii) 1 minute before meeting, with an option to snooze or dismiss.
4. If snoozed, alert comes again in 5 seconds.
5. Once dismissed, alert should no longer appear.

6) Planner Views

a) Similar to Google or outlook calendars.

b) The view follows the following guidelines –

1. Planner shows only current year.
2. ii) User(Both admin and normal user) can change months, through an arrow button(or prev/next button)
3. Each month shows all the dates in tabular format like in actual calendar.
4. Day Cells are filled if any meeting is kept with a design.
5. Upon clicking the day's cell, a view pops up that shows all the meeting details along a 24 hour timeline with the slots covering the exact duration of each meeting.
6. Upon clicking on a meeting, its details pops up in another view
7. c) Admin Flow:
8. i) For Admin, a schedule meeting button is there in calendar view to create a meeting.

ii) Upon clicking on create button, details view opens

iii) Once the meeting is created it appears on the calendar view.

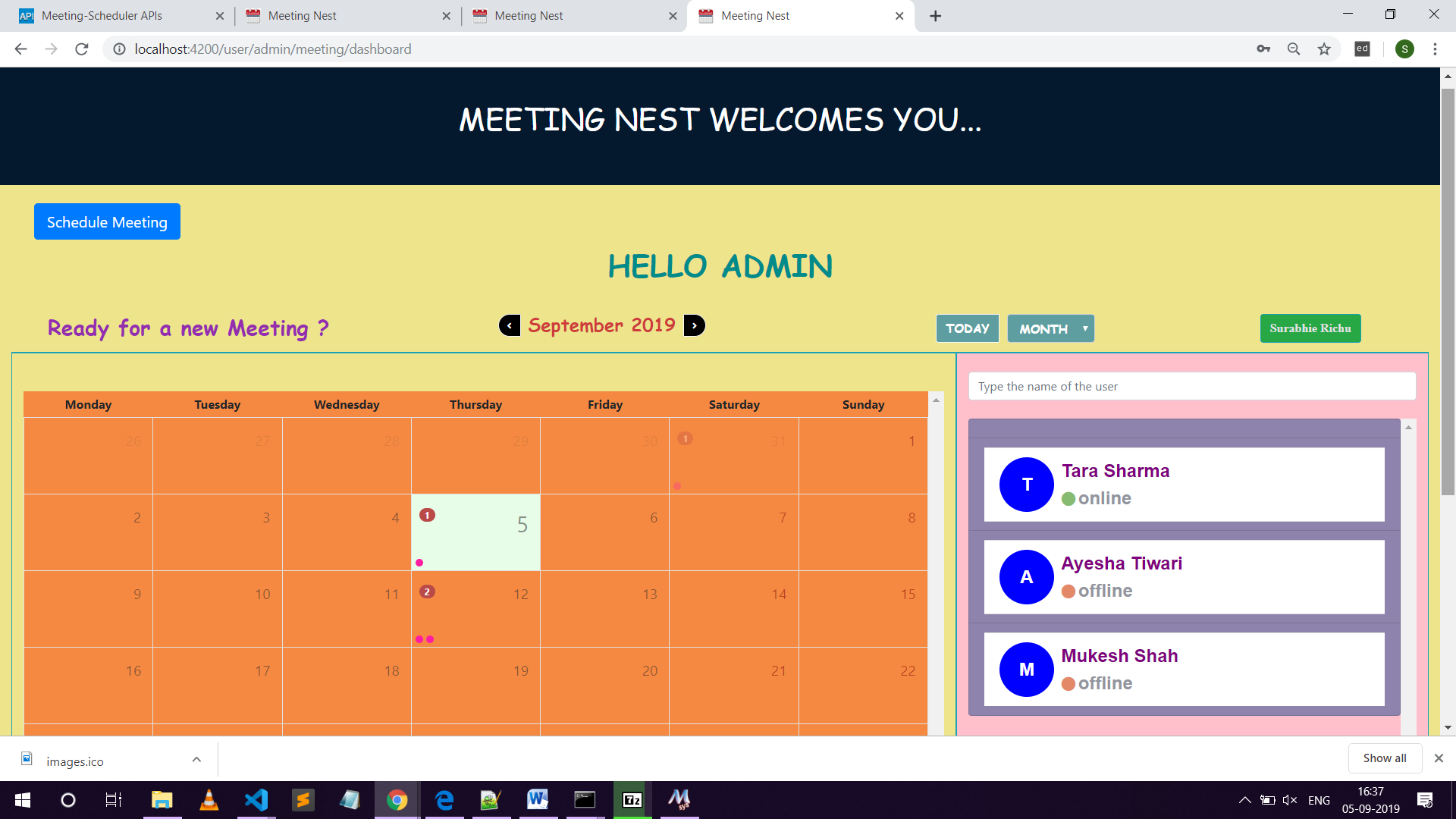
iv) Upon clicking on an already created meeting, same details view opens

1. Details view is a form
2. Admin can edit the meeting details and reschedule the meeting.
3. Admin can delete a meeting as well with a delete button
4. Meeting details covers when, where and purpose. Also, by default username of the admin, who kept the meeting, is mentioned in non-editable format.

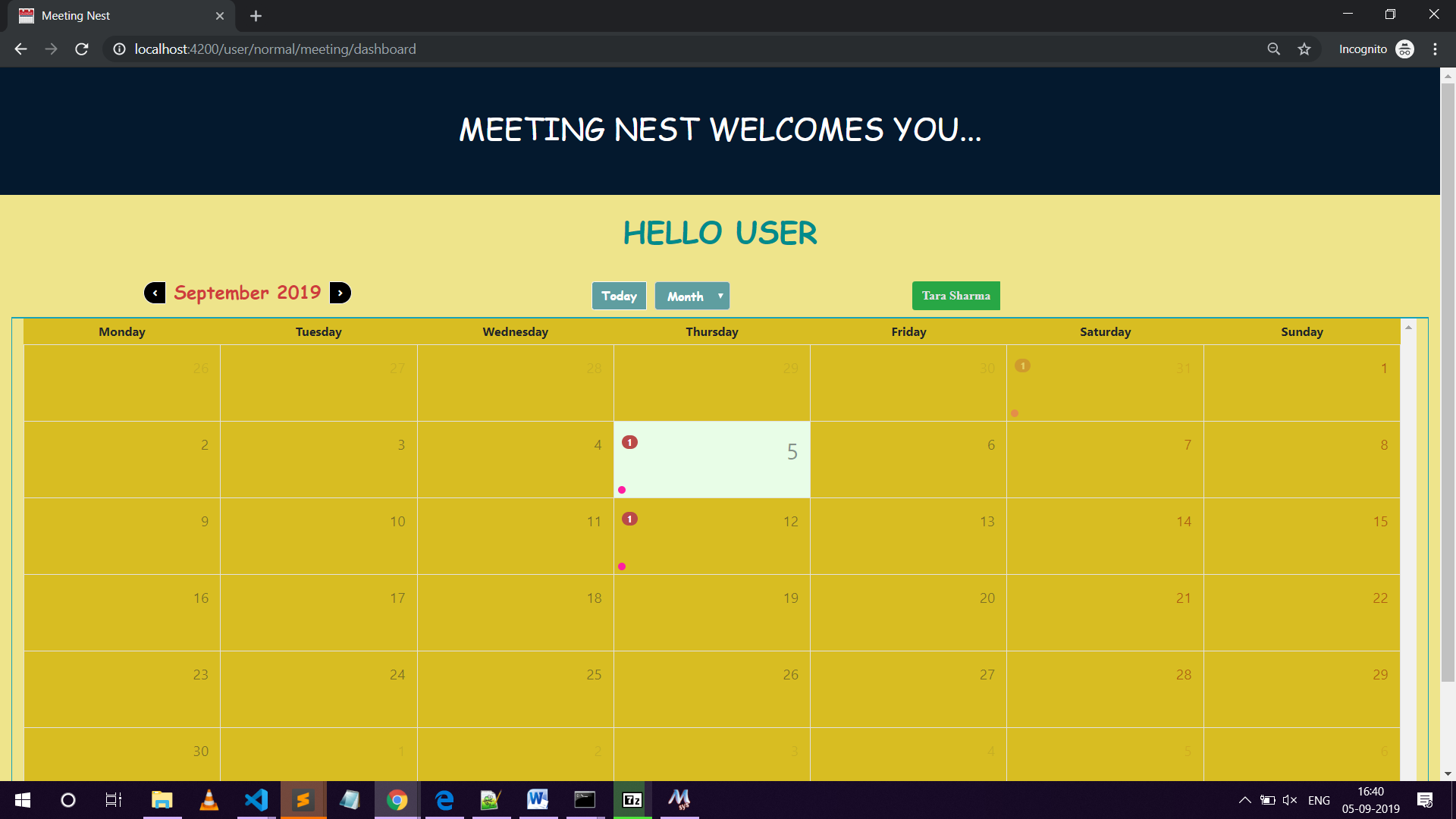
7) Error Views and messages - Each major error response is handled (like 404 or 500) with a different page. Also, all kind of errors, exceptions and messages are handled properly on frontend. The user is aware all the time on frontend about what is happening in the system

SCREENSHOTS:

1-Admin dashboard



2-user dashboard



GIT REPOSITORY LINK :

1. <https://github.com/Surabhie/Meeting-Nest-Frontend.git>
2. <https://github.com/Surabhie/Meeting-Nest-Backend.git>

SUBMITTED BY:

SURABHI