Below are the list of my last three projects where front-end was developed in Angular and back-end for two projects were written in JAX-RS RESTful web services while Google's FireBase was used for the last project. All the projects used routing-module and different types of Service module of Angular.

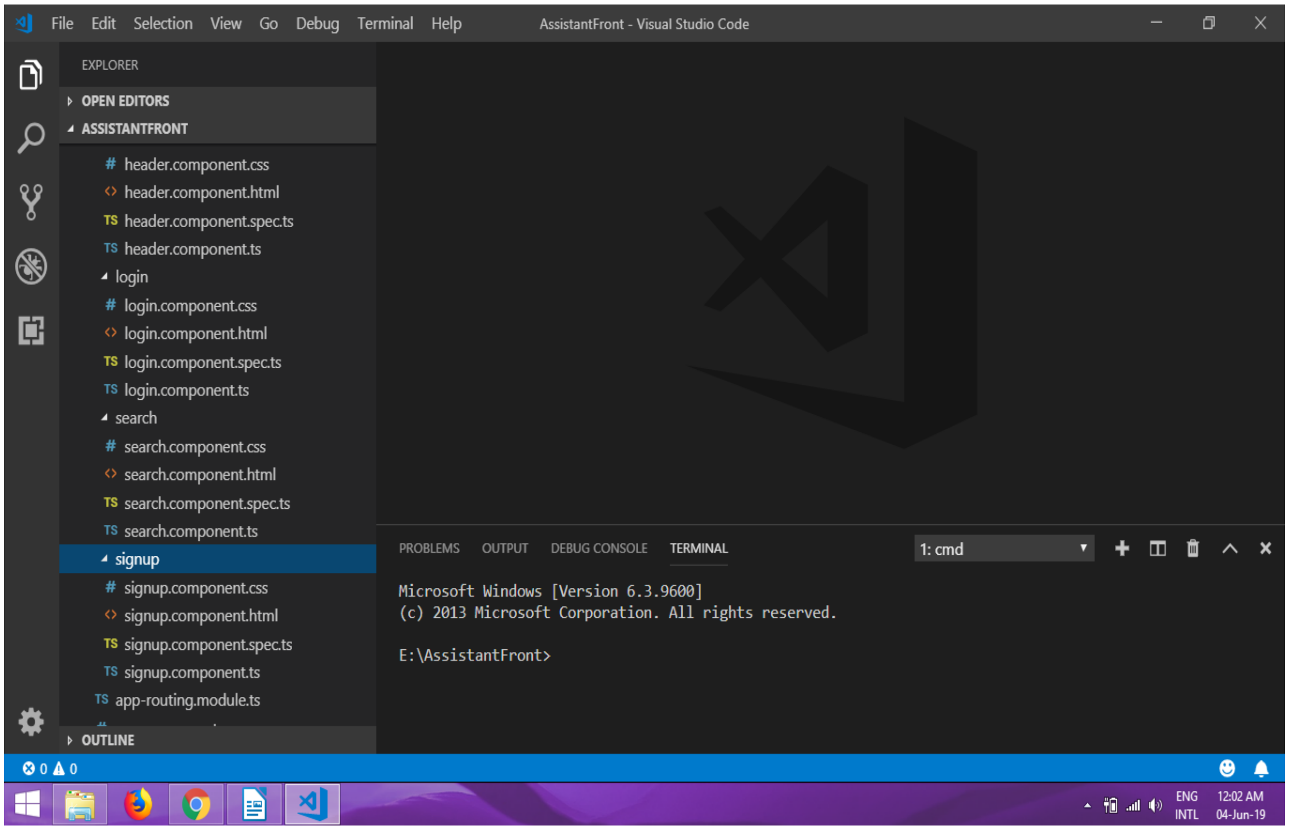
Project 1: Digital Assistant

Project title: Web Application Using Restful Micro Web Service for Query Handling

Project final build Github URL: "https://github.com/subrat211/digitalAssistant "

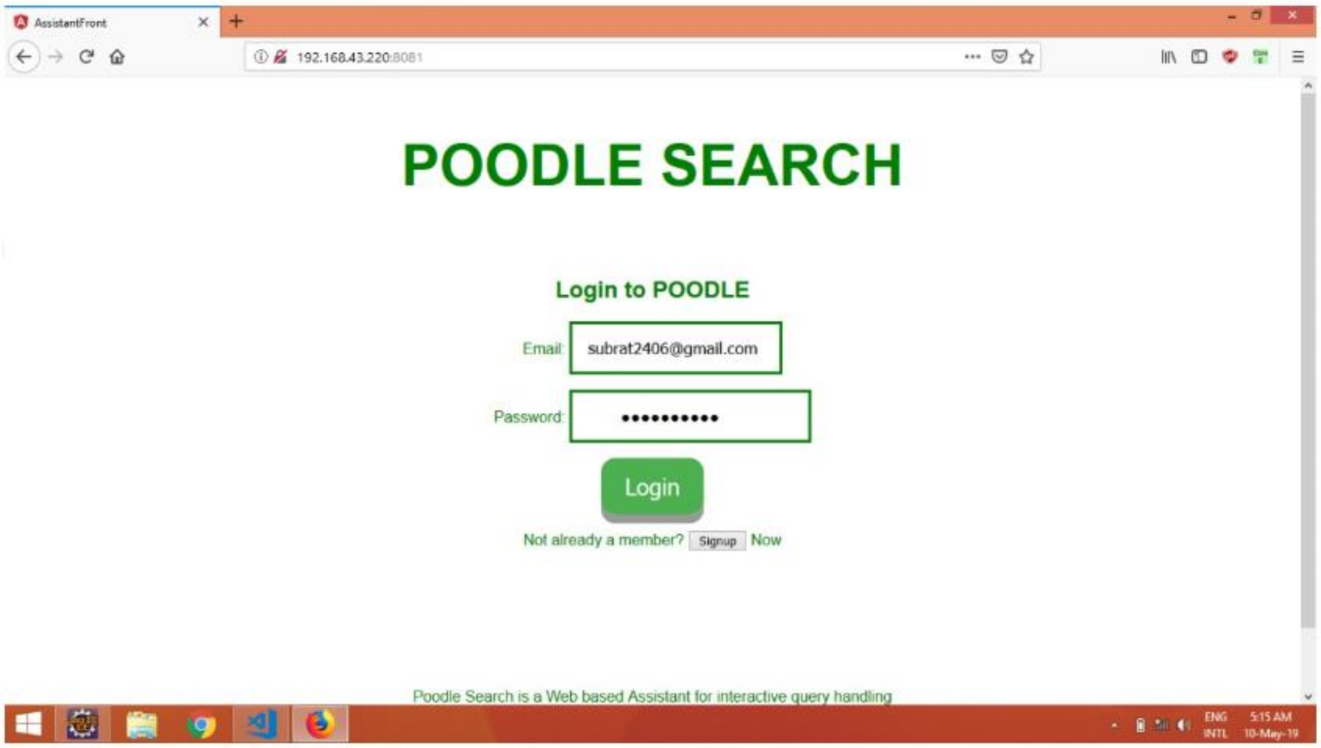
Project overview:
It is a web based "Digital Assistant" where the front-end was written in Angular 6 and back-end was written in REST Web Service. Back-end used WolframAlpha's API for answering dynamic user queries like finding the weather of a place, etc. while more static queries like greetings, fun question, etc. were handled by local SQL server. For achieving this the query was parsed to get rid of the stop words then the program decides who will handle the query either the local database or WolframAlpha Api. Testing for the back-end API's were done on POSTMAN REST API Client. Coming to the front-end HTML and CSS were the main player and as the Angular follows component approach there were separate files for HTML, CSS and TypeScript for each module. There were 5 components namely "Header", "Footer", "Login", "Signup" and "Search". The header and footer components were used in every page of the application as a static element. Each component contains their specific elements as shown

Screenshots:



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

E:\AssistantFront>
```



AssistantFront

192.168.43.220:8081

POODLE SEARCH

Login to POODLE

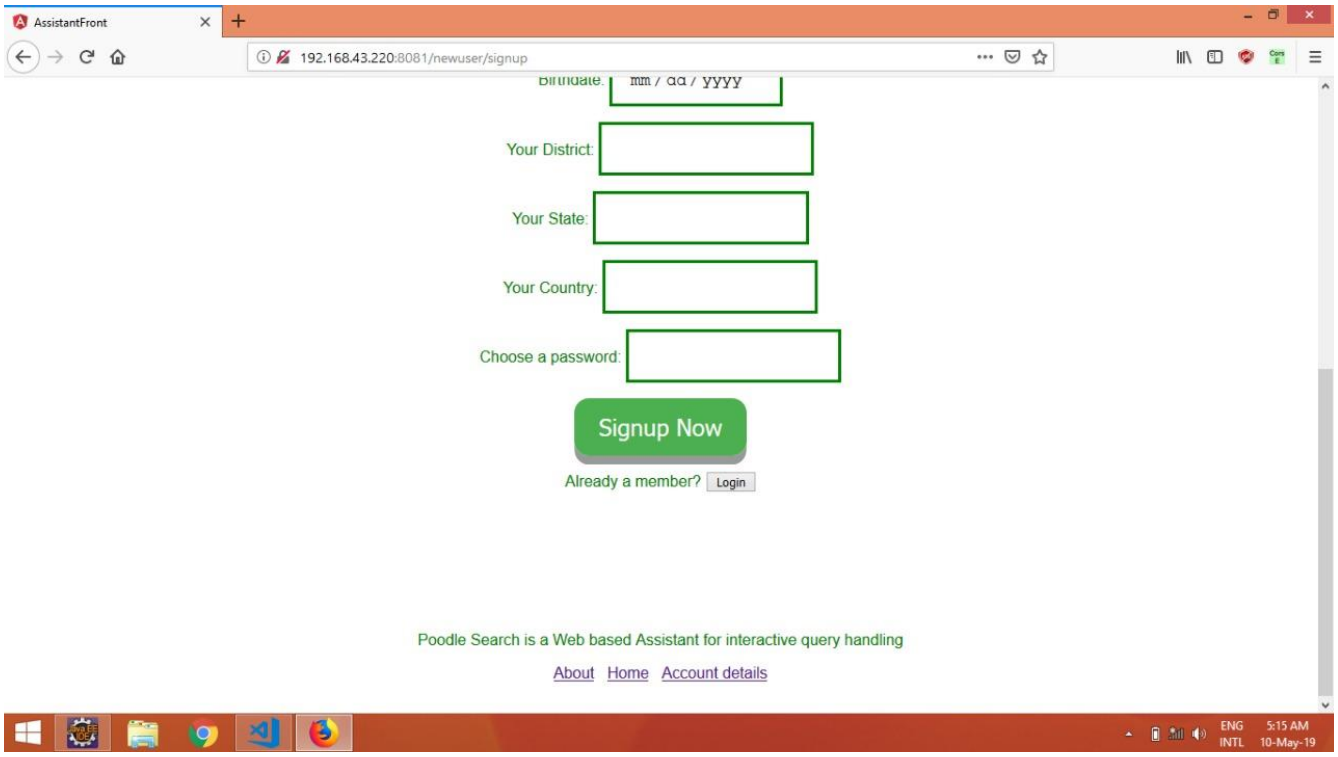
Email: subrat2406@gmail.com

Password: ••••••••••

Login

Not already a member? Signup Now

Poodle Search is a Web based Assistant for interactive query handling



AssistantFront

192.168.43.220:8081/newuser/signup

Birthdate: mm / dd / yyyy

Your District:

Your State:

Your Country:

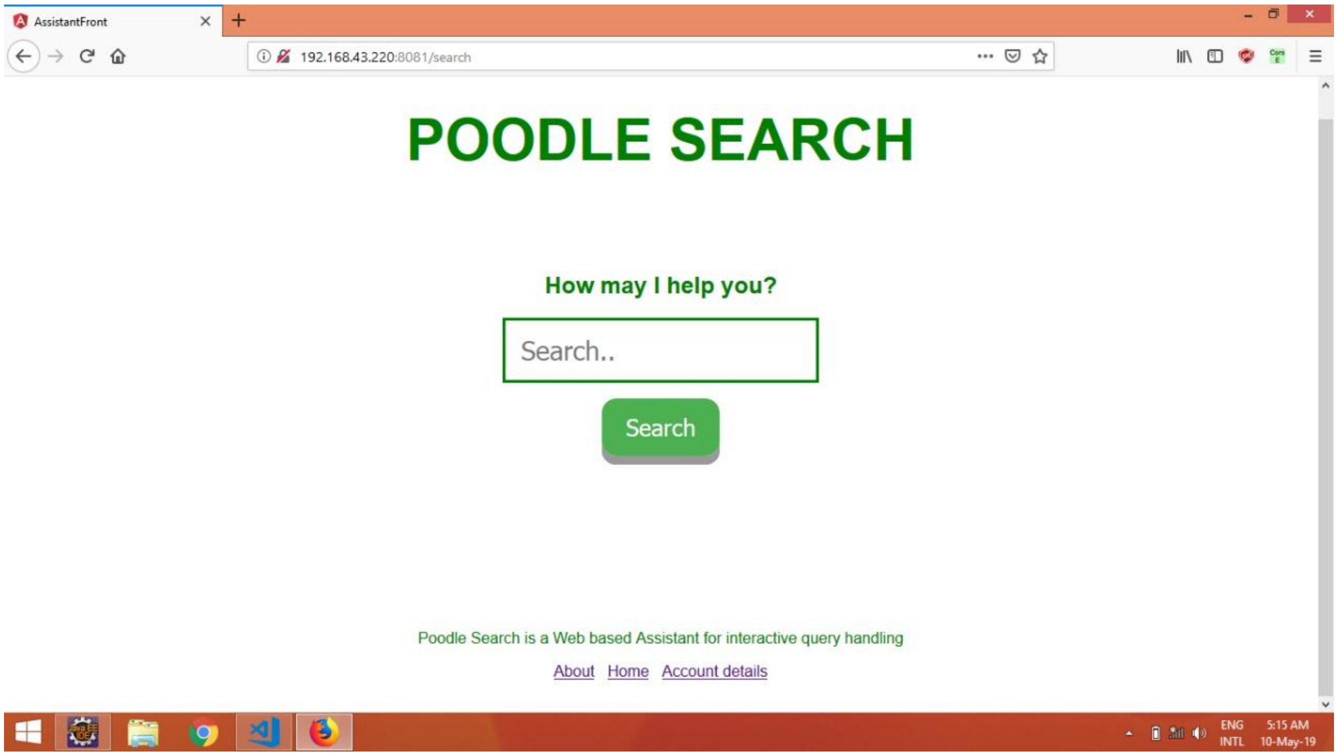
Choose a password:

Signup Now

Already a member? Login

Poodle Search is a Web based Assistant for interactive query handling

About Home Account details


AssistantFront

192.168.43.220:8081/search

POODLE SEARCH

How may I help you?

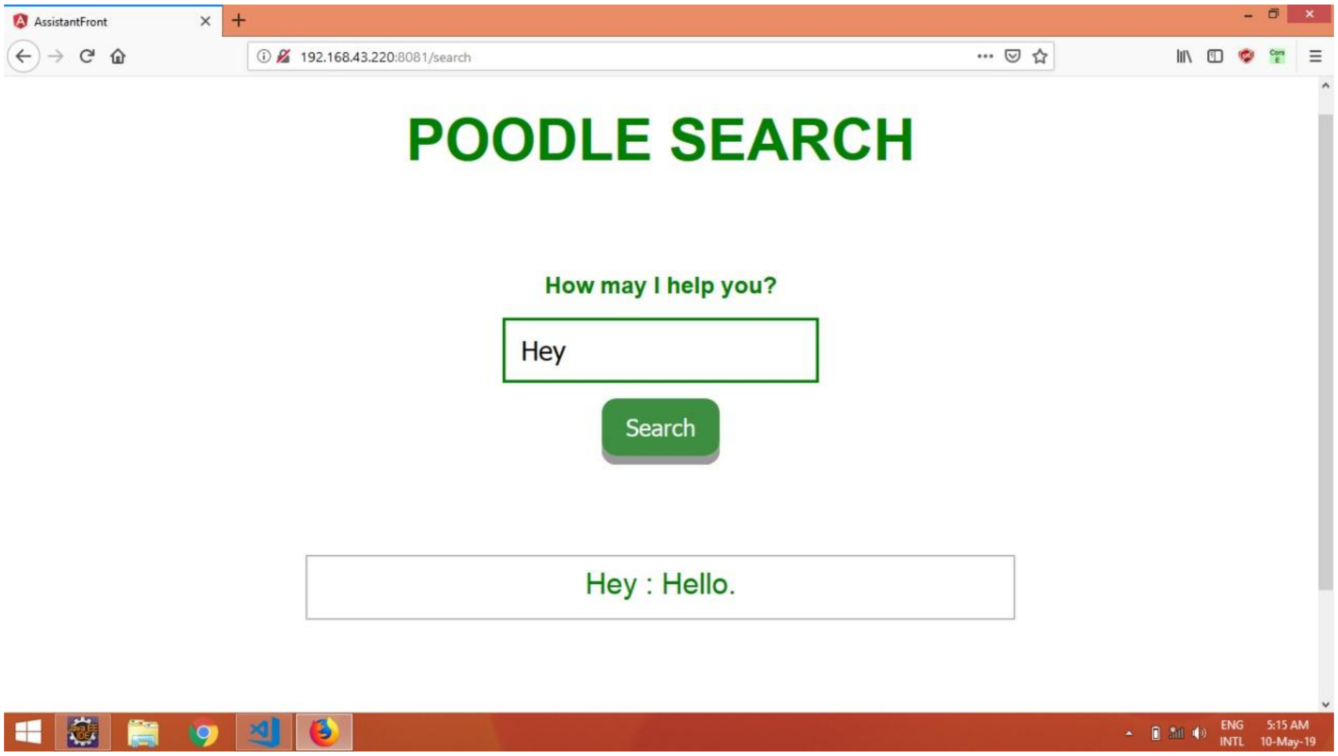
Search..

Search

Poodle Search is a Web based Assistant for interactive query handling

About Home Account details

ENG INTL 5:15 AM 10-May-19


AssistantFront

192.168.43.220:8081/search

POODLE SEARCH

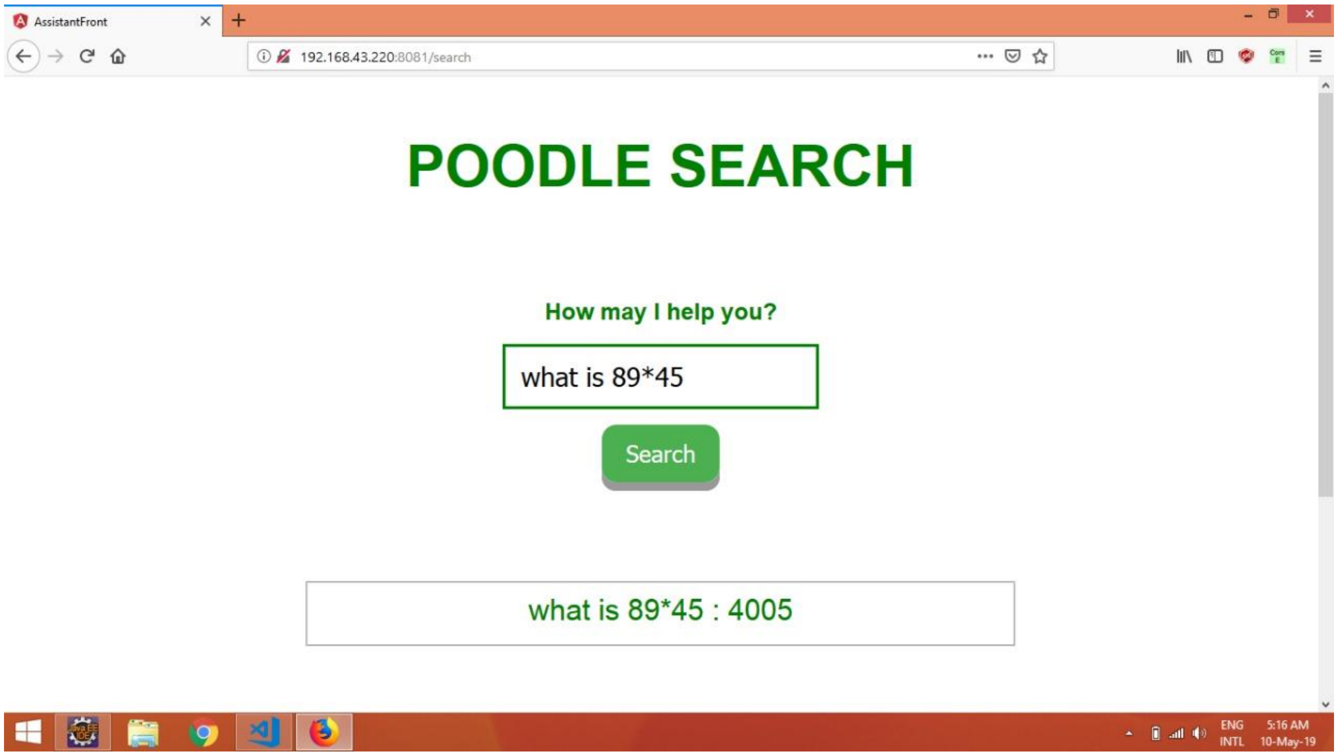
How may I help you?

Hey

Search

Hey : Hello.

ENG INTL 5:15 AM 10-May-19



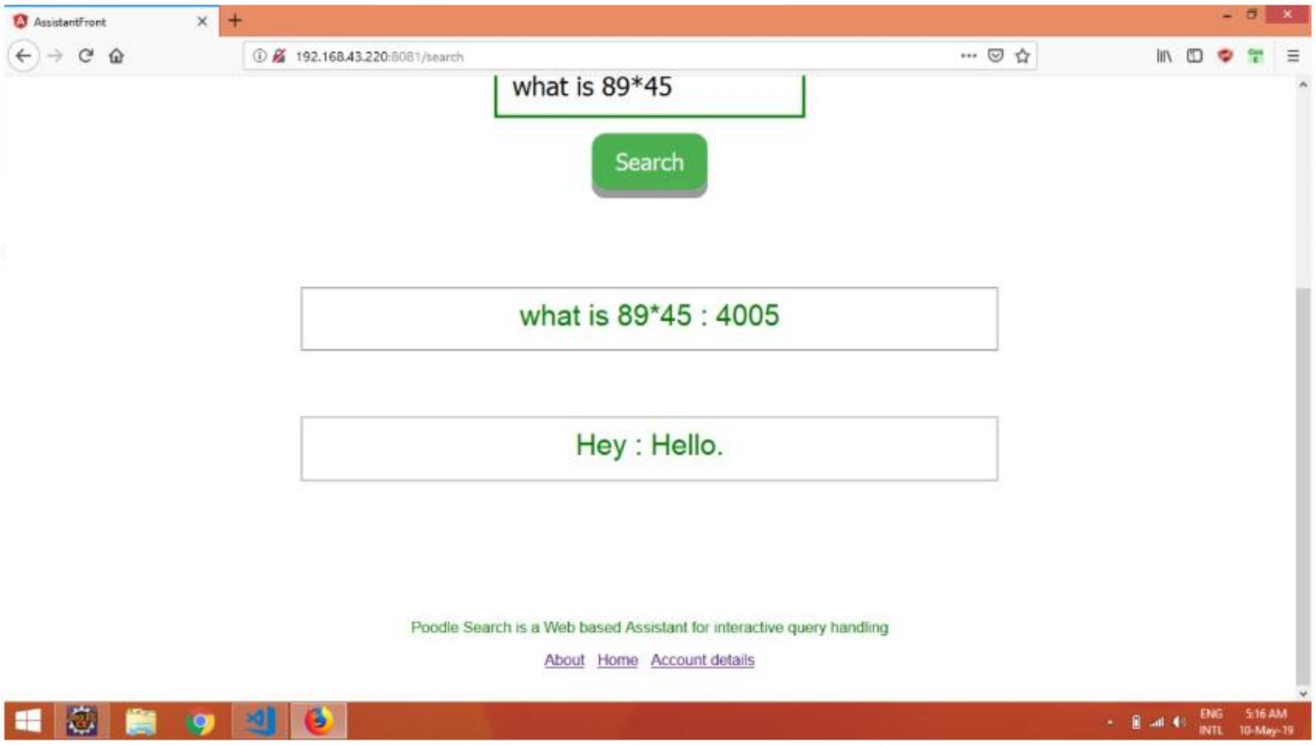
POODLE SEARCH

How may I help you?

what is 89*45

Search

what is 89*45 : 4005



what is 89*45

Search

what is 89*45 : 4005

Hey : Hello.

Poodle Search is a Web based Assistant for interactive query handling

About Home Account details

Project 2: School Attendance Monitoring System

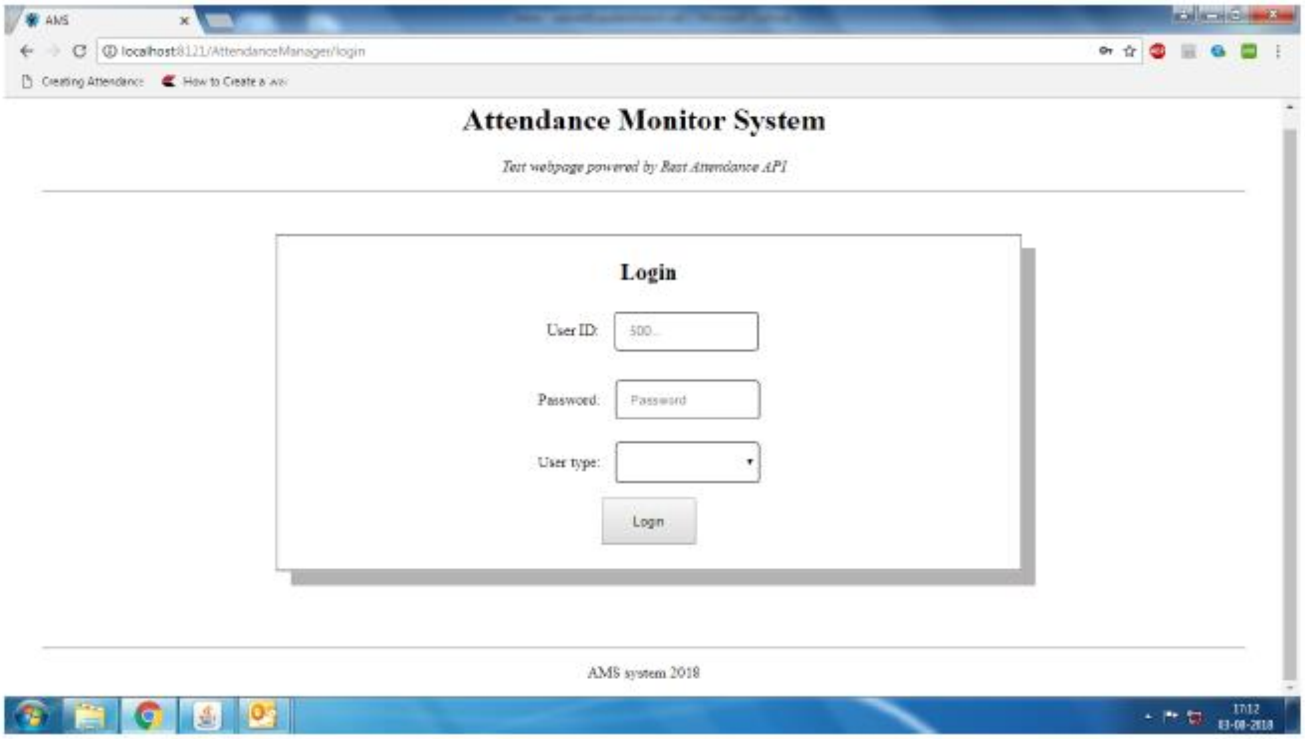
Project title: Attendance Monitor System

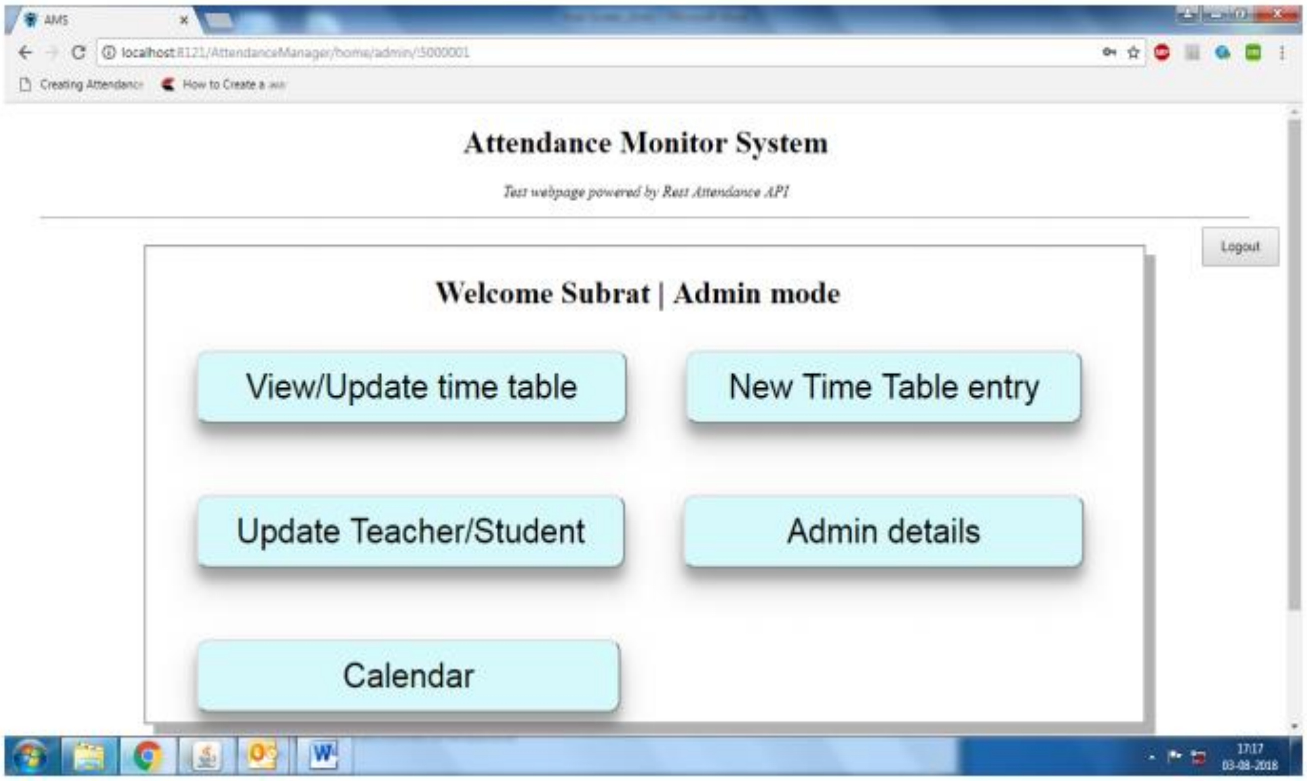
Project Report: https://drive.google.com/open?id=1J6pcSpJoJ_Pd4PBsMiBJCEH3WIHv2Bo4

Project overview:
It is a web-based Attendance Monitoring and Management system where the front-end was written in Angular 6 and back-end was written in REST Web Service. The same component based was approach was used, with component namely “Header”, “Footer”, “Login”, “Signup”, “Home”, “Details”, “AttendanceView”, “AttendanceUpdate”, “TimeTableView”, “TimeTableUpdate” and “DetailsEdit”.

In this the Student, Teachers and Admin can login from the same login page using their login id and password. They after validation by back-end will be redirected to the home page with their access view. From their every actor can perform their specific role that is of checking the timetable and attendance for the student, checking the time-table and marking student attendance according to the time table and student branch for the teacher while the admin can add new entry to time table, view old time table and update details of everyone in the system except their own. Teacher and Students can view their own details and update their phone number and address.

Screenshots:

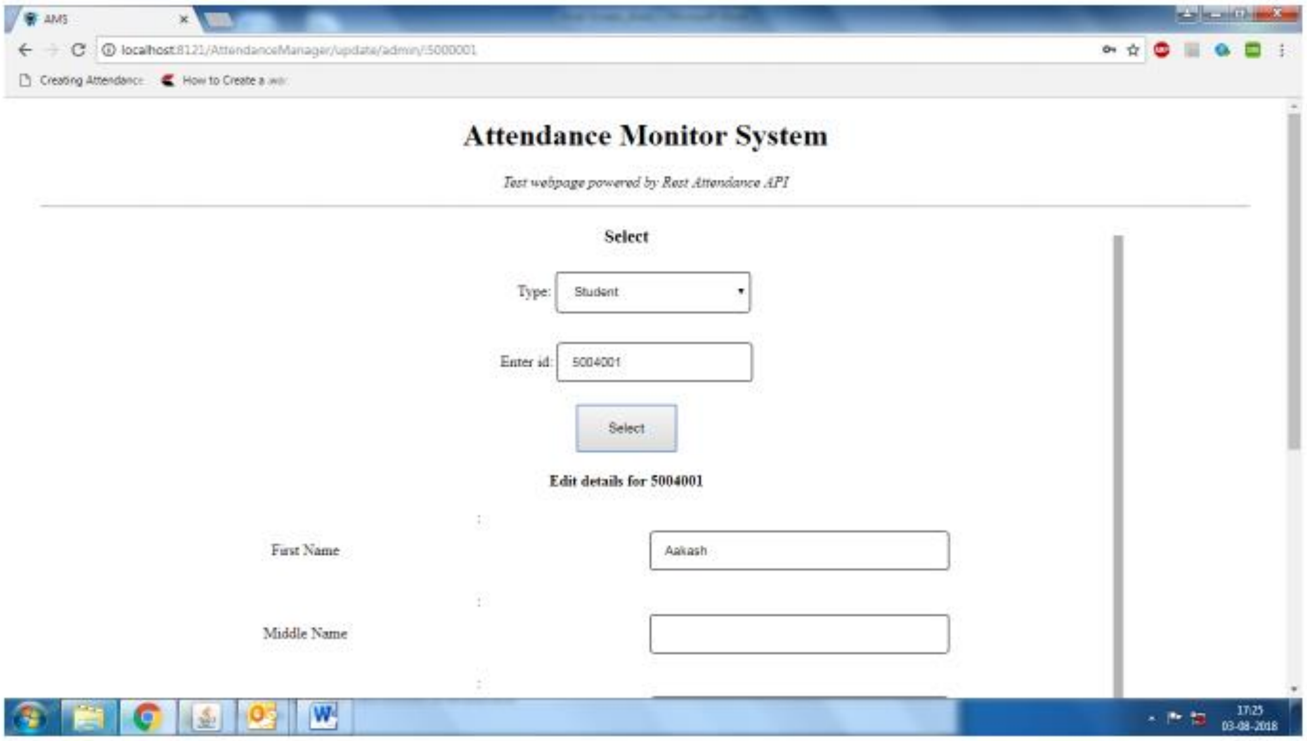

AMS
localhost:8121/AttendanceManager/login
Creating Attendance
How to Create a
Attendance Monitor System
Test webpage powered by Rest Attendance API
Login
User ID:
500...
Password:
Password
User type:
Login
AMS system 2018



Attendance Monitor System

Test webpage powered by Rest Attendance API

Welcome Subrat | Admin mode



Attendance Monitor System

Test webpage powered by Rest Attendance API

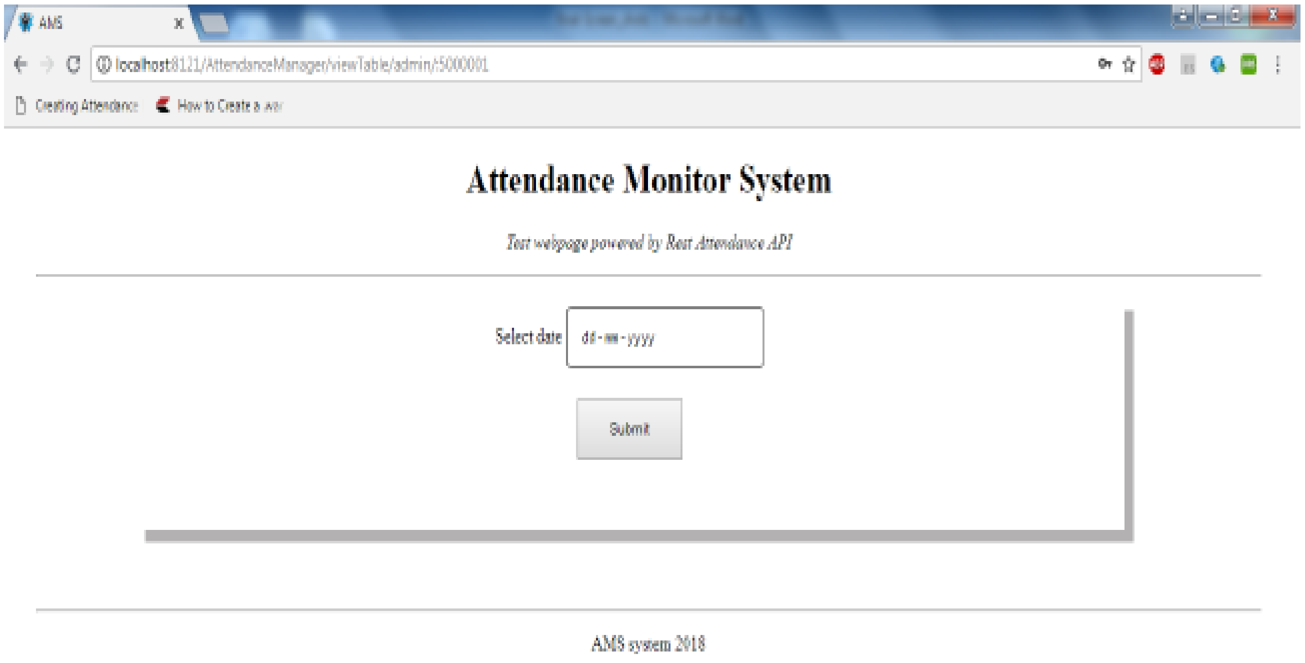
Select

Type: Student

Enter id: 5004001

Edit details for 5004001

First Name	:	Aakash
Middle Name	:	


localhost:8121/AttendanceManager/viewTable/admin/:5000001

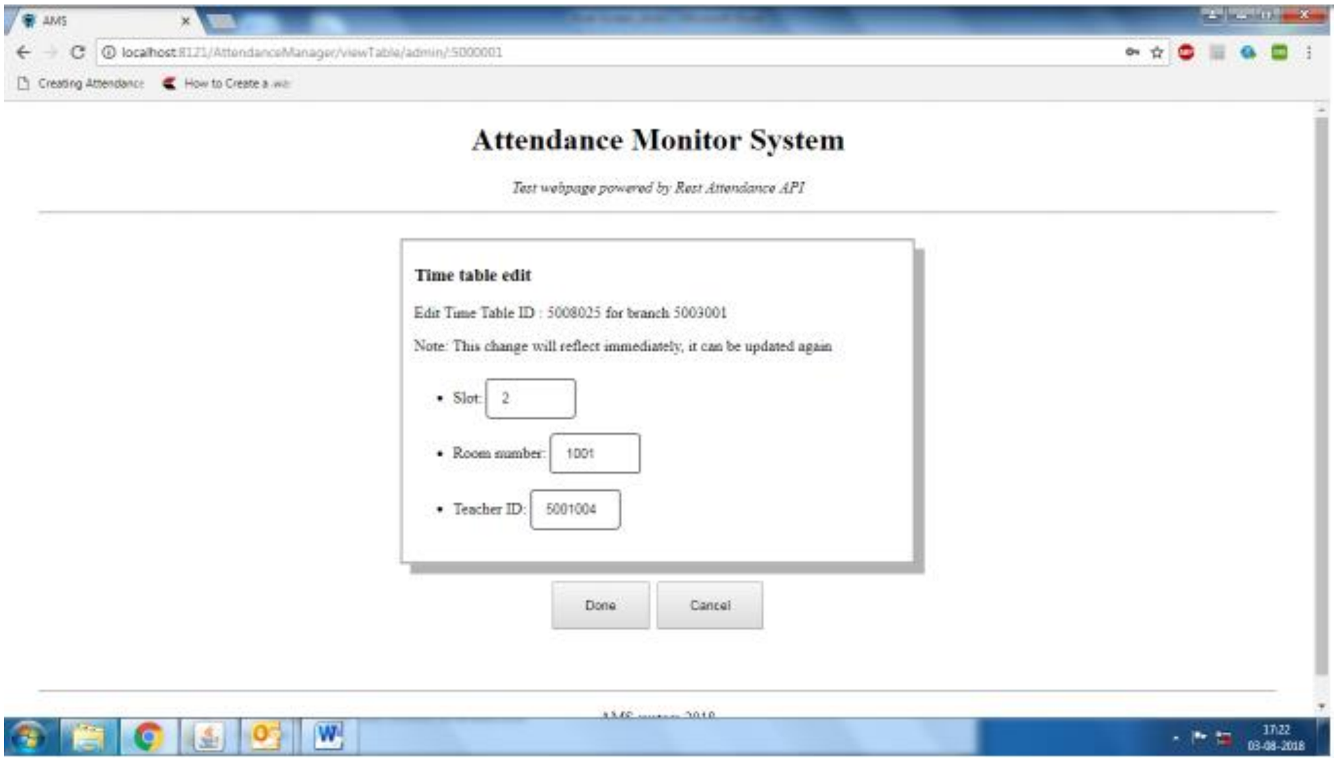
Attendance Monitor System

Test webpage powered by Rest Attendance API

Select date dd-mm-yyyy

Submit

AMS system 2018


localhost:8121/AttendanceManager/viewTable/admin/:5000001

Attendance Monitor System

Test webpage powered by Rest Attendance API

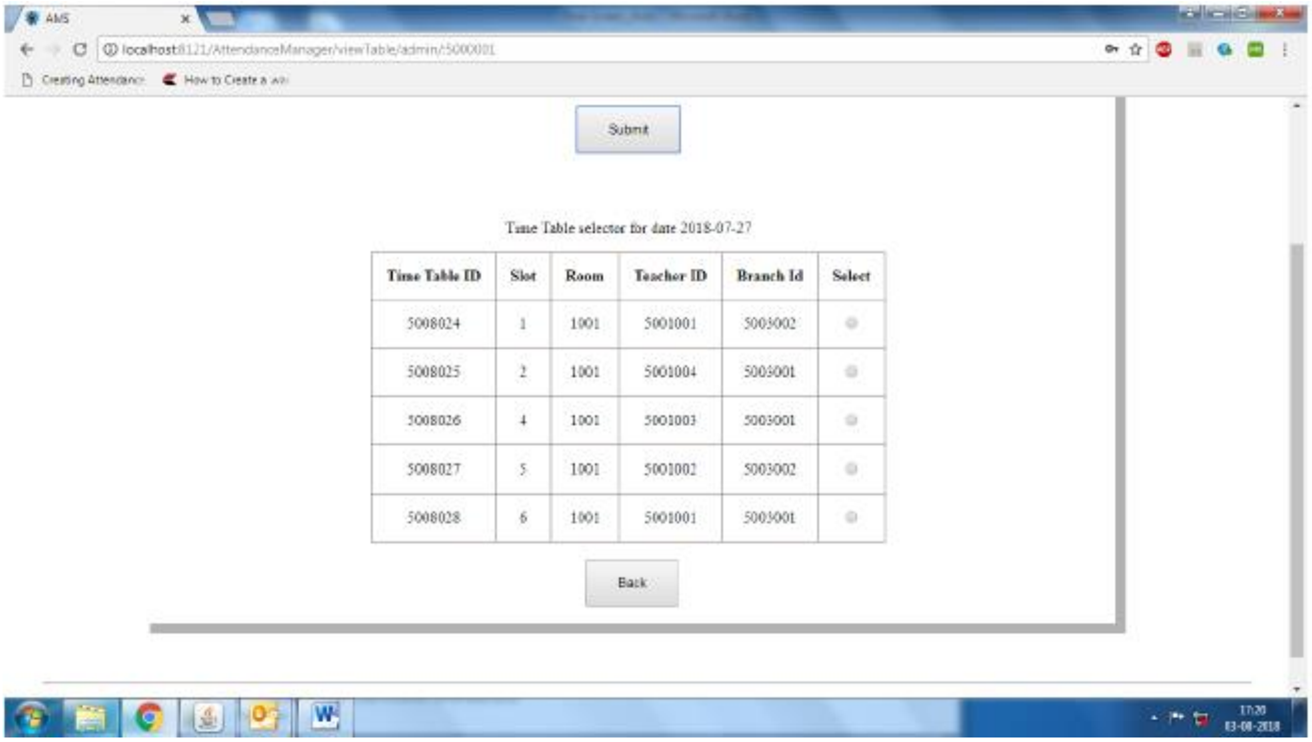
Time table edit

Edit Time Table ID : 5008025 for branch 5003001

Note: This change will reflect immediately, it can be updated again

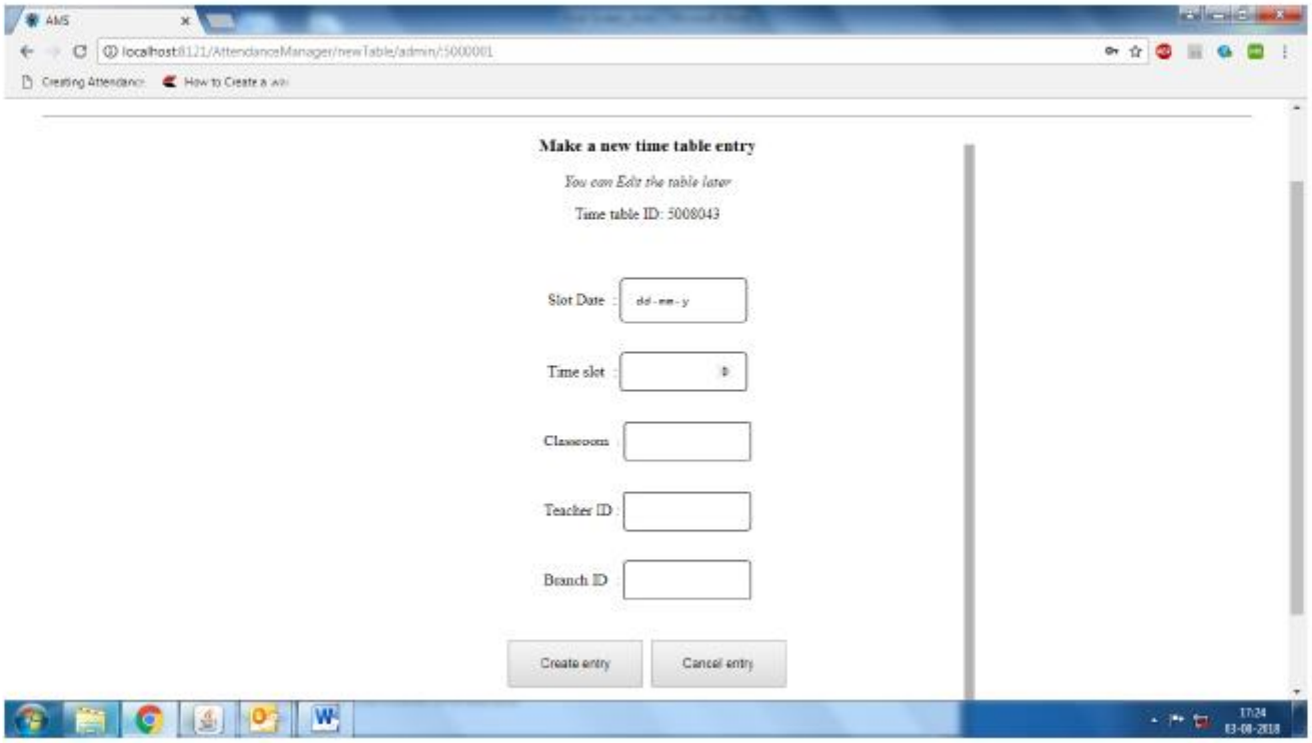
- Slot: 2
- Room number: 1001
- Teacher ID: 5001004

Done Cancel



Time Table selector for date 2018-07-27

Time Table ID	Slot	Room	Teacher ID	Branch Id	Select
5008024	1	1001	5001001	5003002	○
5008025	2	1001	5001004	5003001	○
5008026	4	1001	5001003	5003001	○
5008027	5	1001	5001002	5003002	○
5008028	6	1001	5001001	5003001	○

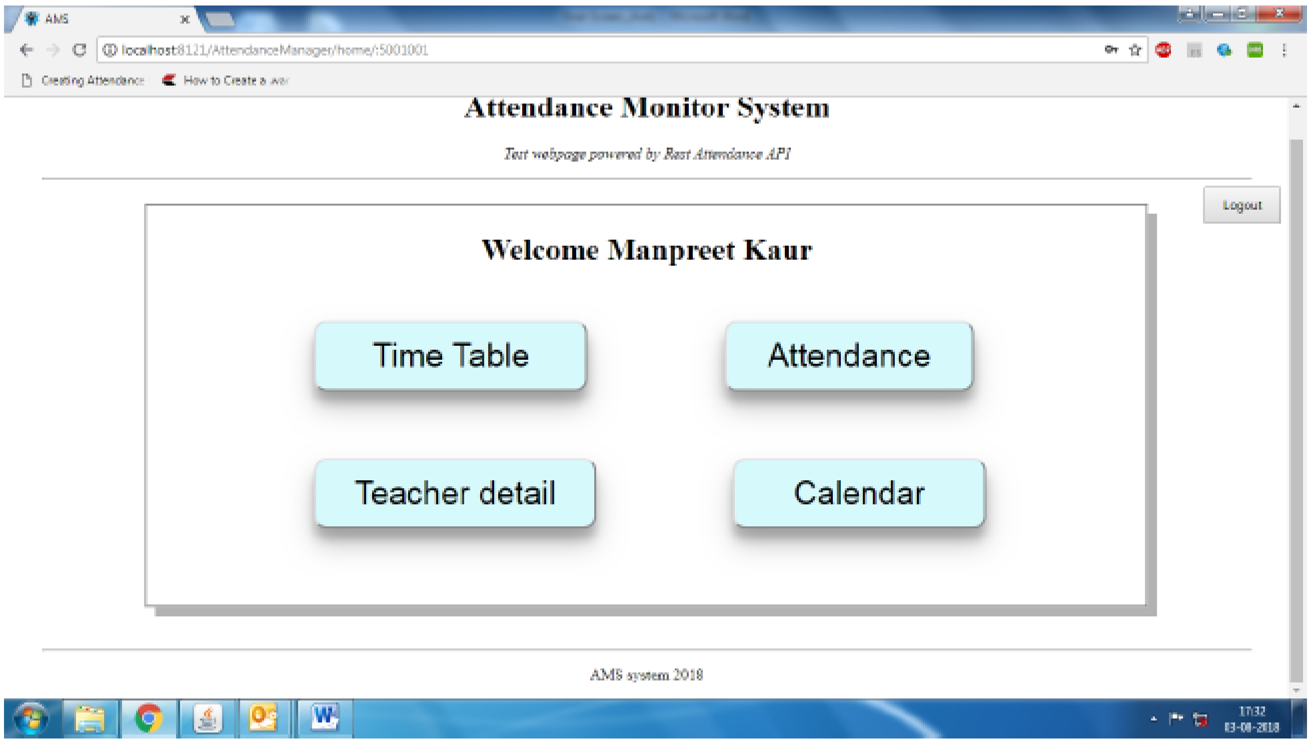


Make a new time table entry

You can Edit the table later

Time table ID: 5008043

Slot Date :	dd-mm-y
Time slot :	
Classroom :	
Teacher ID :	
Branch ID :	



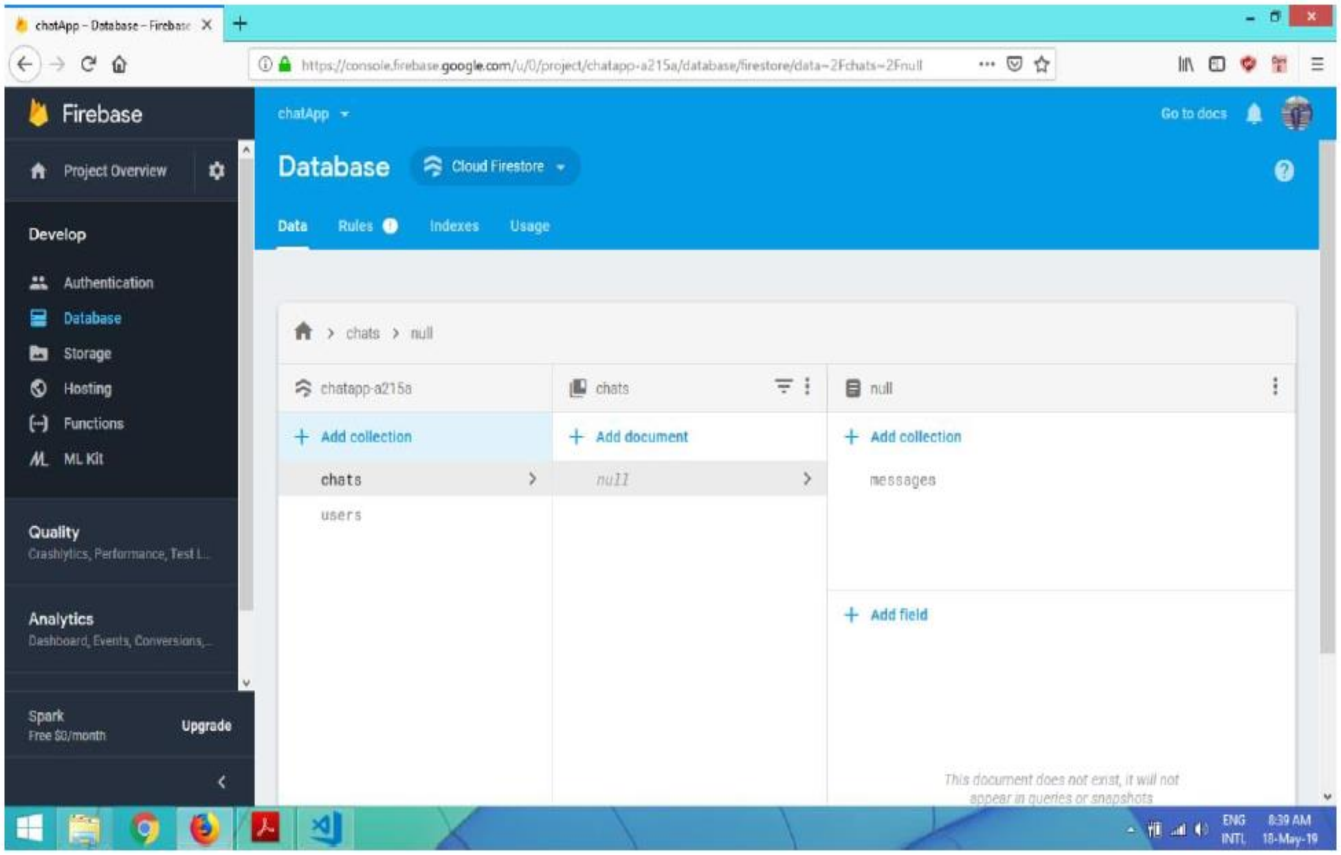
Project 3: Web based chatting application

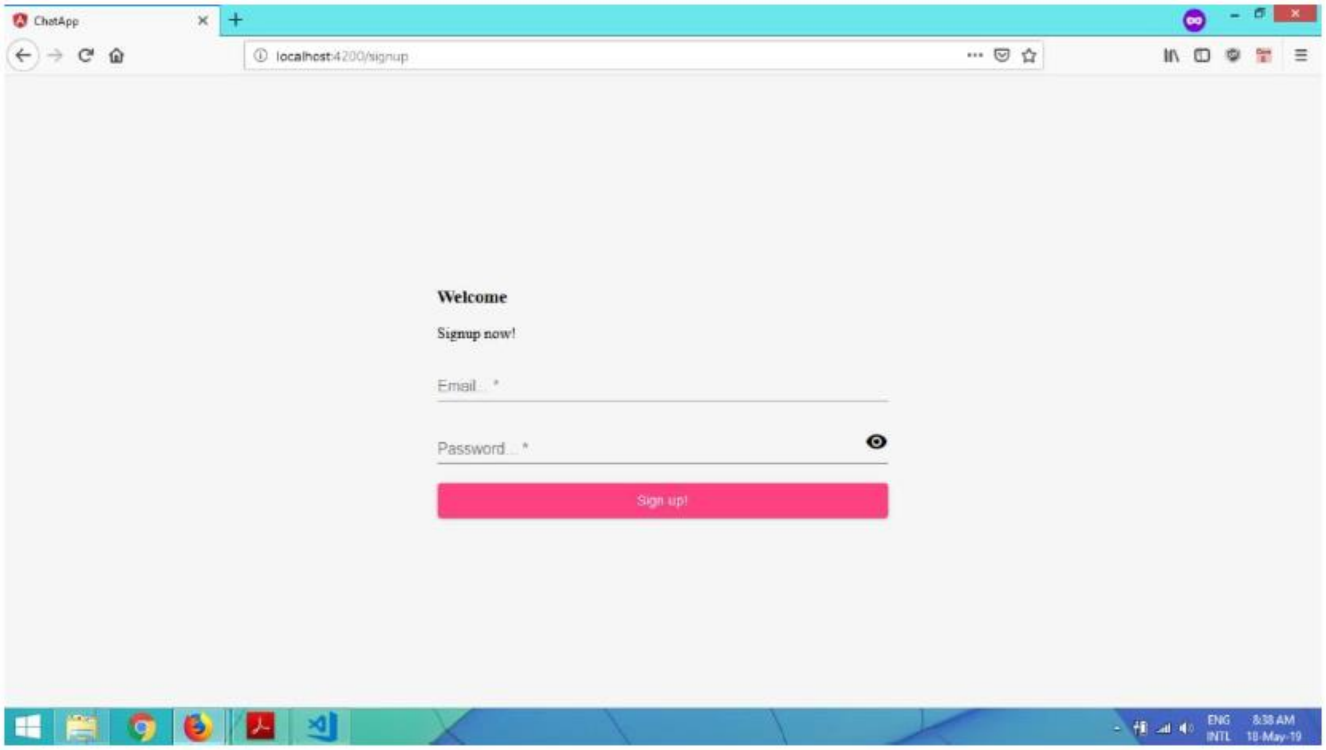
Project title: Chat App

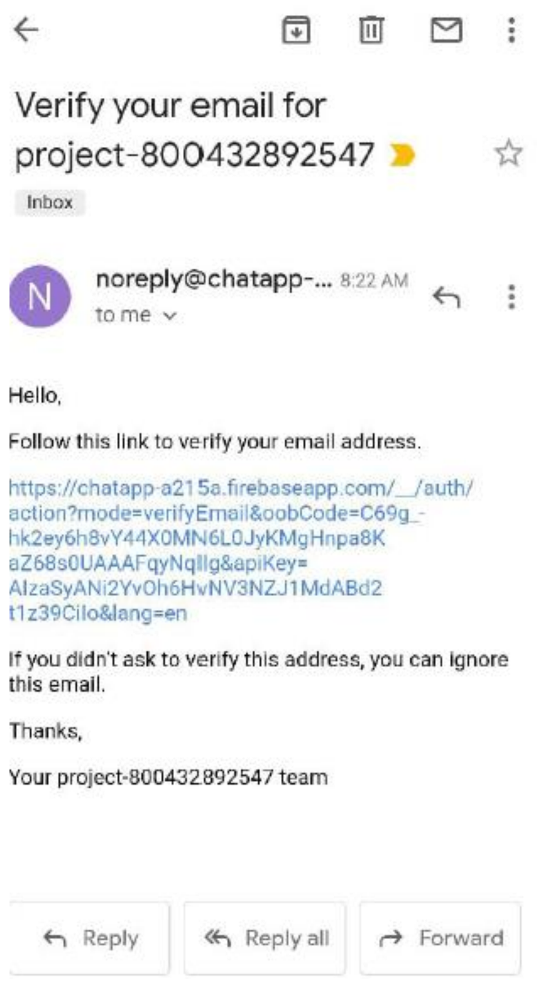
Project Overview:
It is a web-based chatting application where user can either login or create their account and chat with the people on the network. It uses Firebase as it's back-end, so it is the only project that uses a NoSql database. It uses Firebase build in authentication system to login in the user using Google account or their email id, password combination.

The users can chat in a private session or a public chatroom of their choice, they can create a chatroom of their own. Chats end whenever someone clicks on end chat button. HTML formatting/ DOM manipulation was done using Angular DOM manipulators like *ngFor and *ngIf. The screenshot tells the complete story in detail. The bootstrap font used is "https://bootswatch.com/simplex/" for this project.

Screenshots:


chatApp – Database – Firebase
https://console.firebase.google.com/u/0/project/chatapp-a215a/database/firestore/data~2Fchats~2Fnull
Firebase
Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics, Performance, Test L...
Analytics
Dashboard, Events, Conversions,...
Spark
Free $0/month
Upgrade
chatApp
Go to docs
Database
Cloud Firestore
Data
Rules
Indexes
Usage
chats > null
chatapp-a215a
Add collection
chats
users
chats
Add document
null
null
Add collection
messages
Add field
This document does not exist, it will not appear in queries or snapshots


ChatApp
localhost:4200/signup
Welcome
Signup now!
Email *
Password *
Sign up!



Verify your email for project-800432892547

Inbox

noreply@chatapp-... 8:22 AM

to me

Hello,

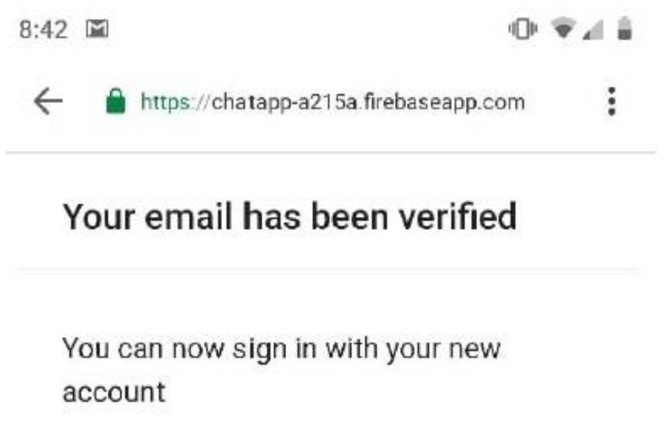
Follow this link to verify your email address.

https://chatapp-a215a.firebaseapp.com/__/auth/action?mode=verifyEmail&oobCode=C69g_-hk2ey6h8vY44X0MN6L0JyKMgHnpa8KaZ68s0UAAAFqyNqllg&apiKey=AIzaSyANi2YvOh6HvNV3NZJ1MdABd2t1z39CiIo&lang=en

If you didn't ask to verify this address, you can ignore this email.

Thanks,

Your project-800432892547 team

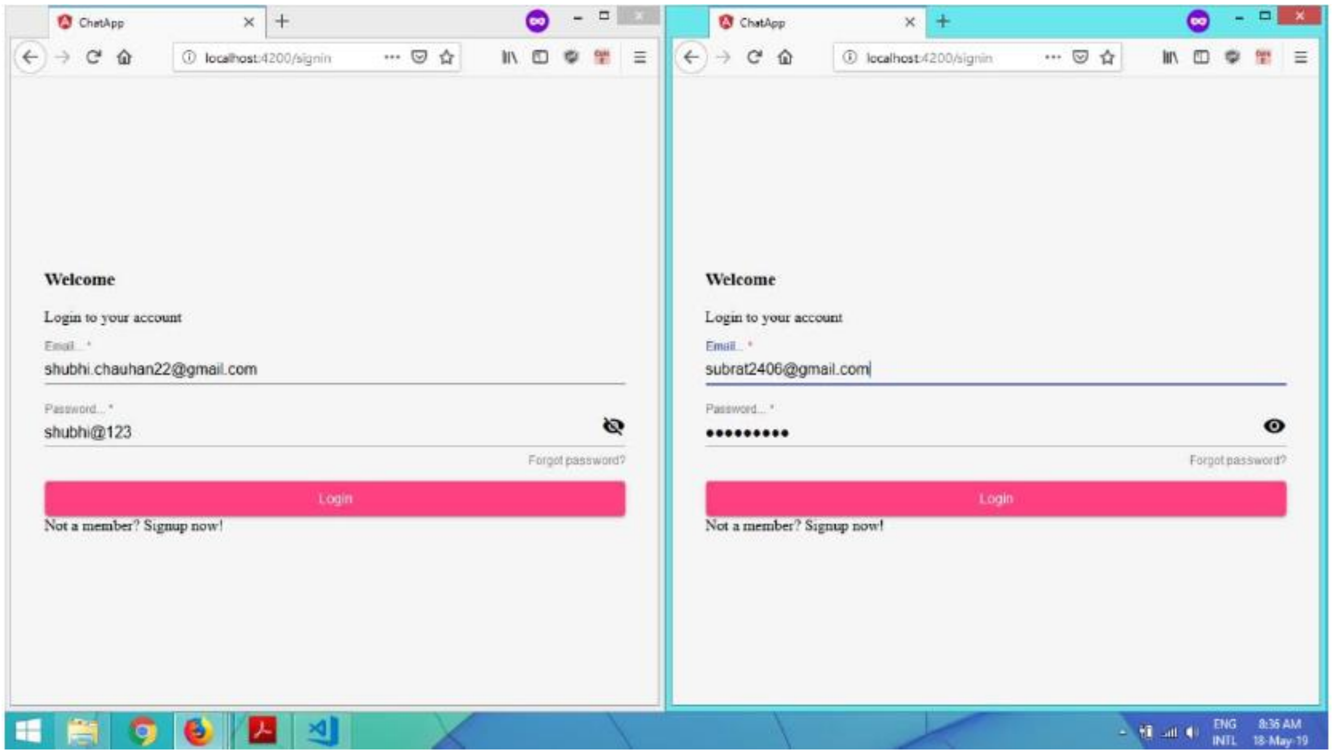


8:42

https://chatapp-a215a.firebaseapp.com

Your email has been verified

You can now sign in with your new account



ChatApp
localhost:4200/signin

Welcome

Login to your account

Email *: shubhi.chauhan22@gmail.com

Password *: shubhi@123

Forgot password?
Login
Not a member? Signup now!

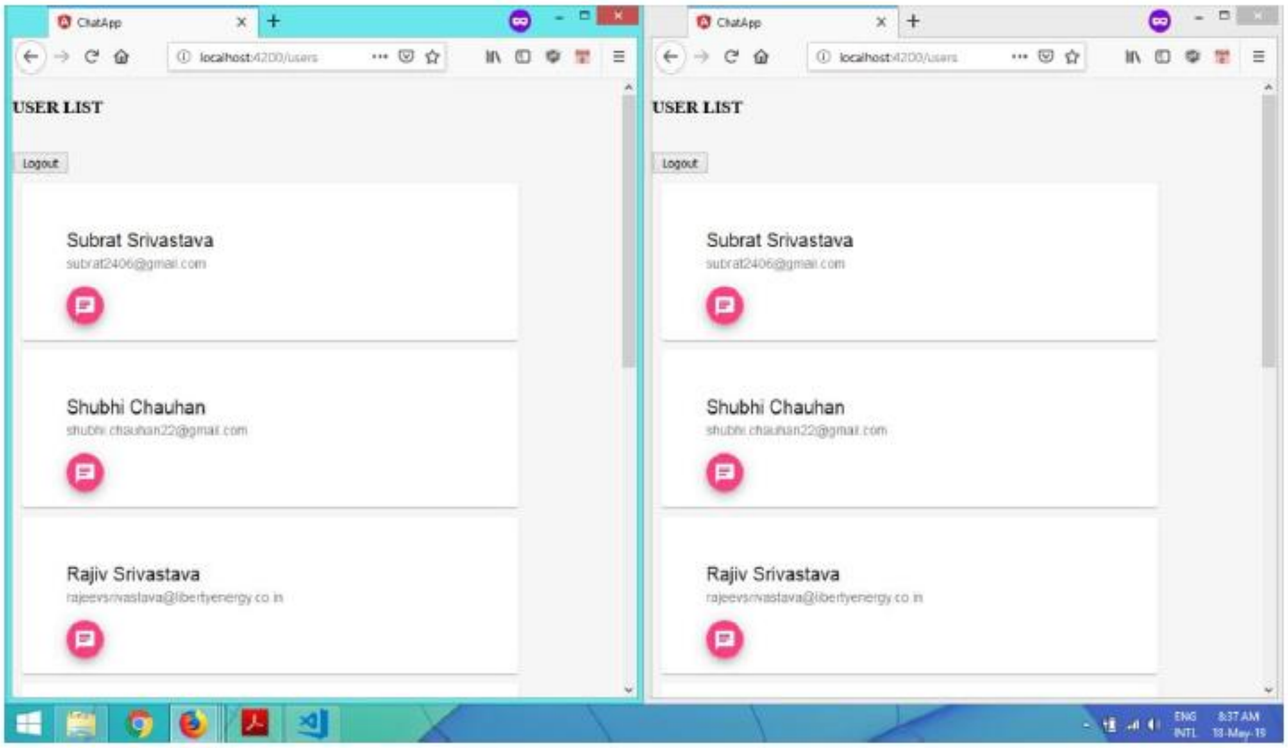
Welcome

Login to your account

Email *: subrat2406@gmail.com

Password *: •••••••••

Forgot password?
Login
Not a member? Signup now!
ENG INTL 8:36 AM 18-May-19



USER LIST

Logout

Subrat Srivastava
subrat2406@gmail.com

Shubhi Chauhan
shubhi.chauhan22@gmail.com

Rajiv Srivastava
rajeevsrivastava@libertyenergy.co.in

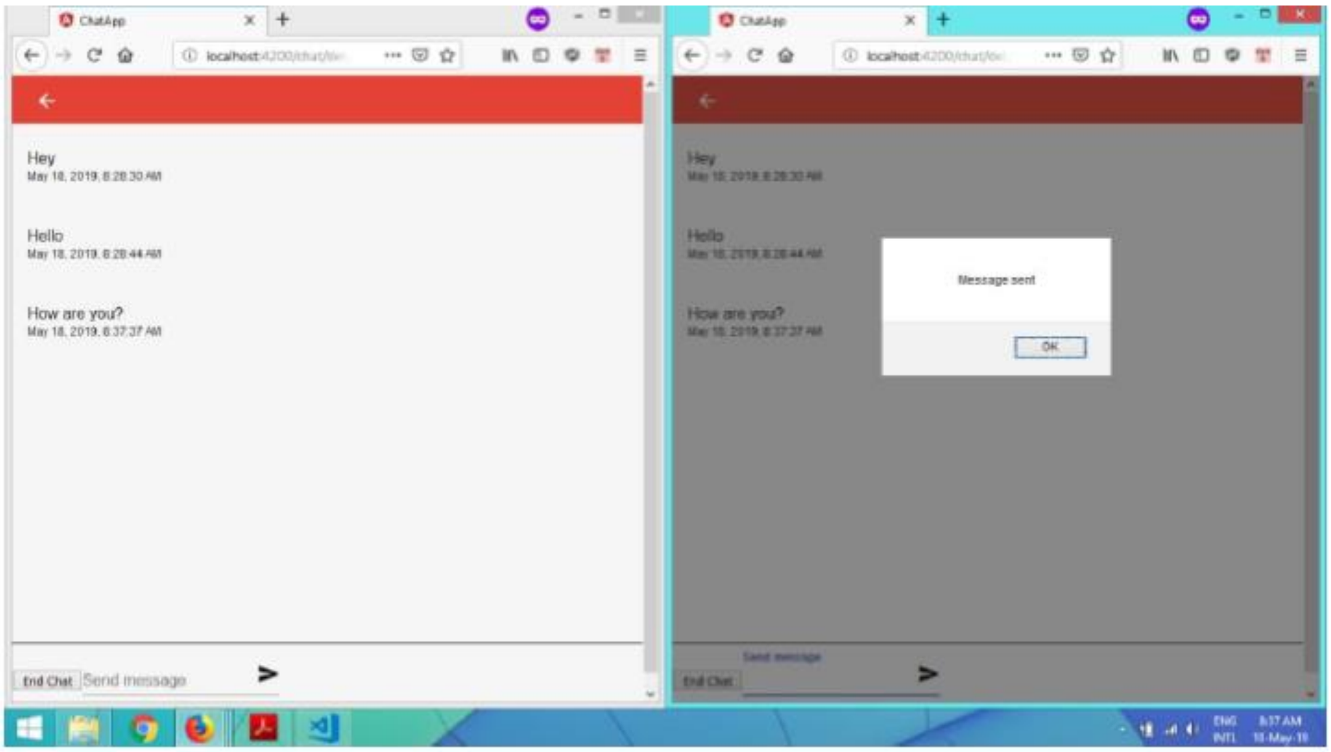
USER LIST

Logout

Subrat Srivastava
subrat2406@gmail.com

Shubhi Chauhan
shubhi.chauhan22@gmail.com

Rajiv Srivastava
rajeevsrivastava@libertyenergy.co.in



Hey
May 18, 2019, 8:28:30 AM

Hello
May 18, 2019, 8:28:44 AM

How are you?
May 18, 2019, 8:37:37 AM

End Chat Send message

Hey
May 18, 2019, 8:28:30 AM

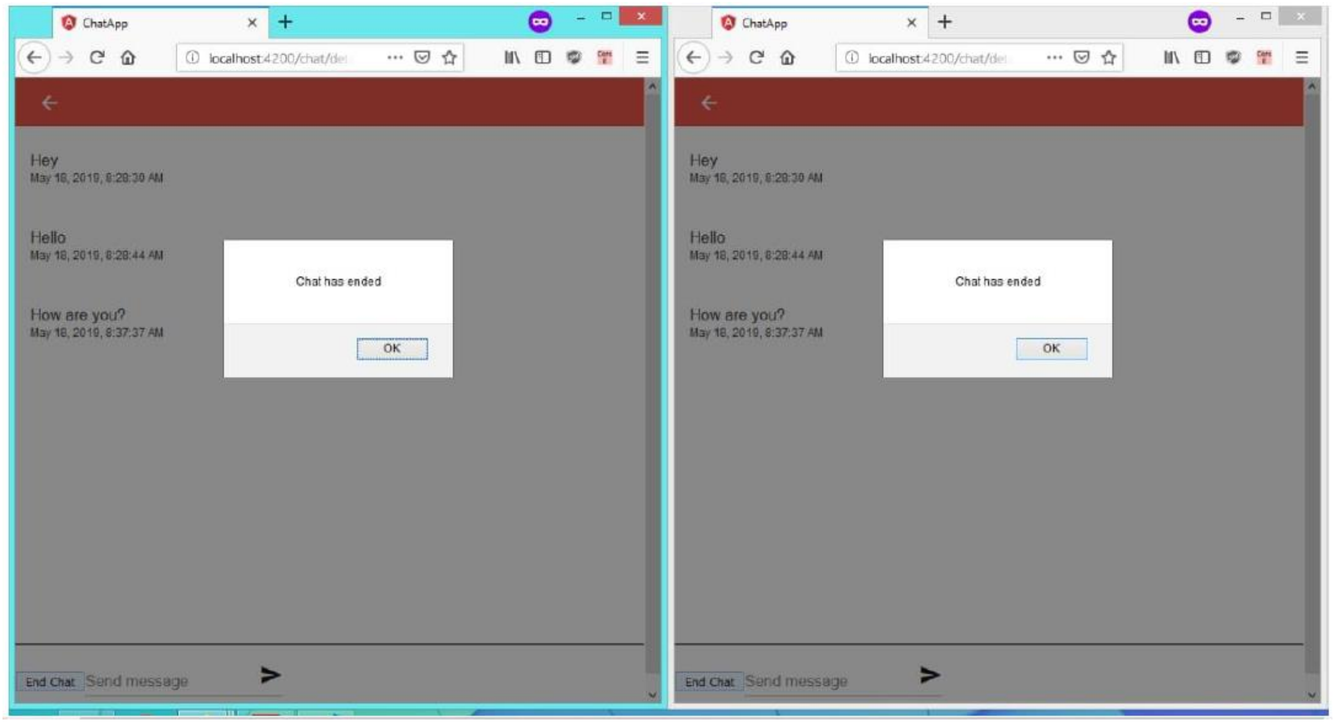
Hello
May 18, 2019, 8:28:44 AM

How are you?
May 18, 2019, 8:37:37 AM

Message sent

OK

End Chat Send message

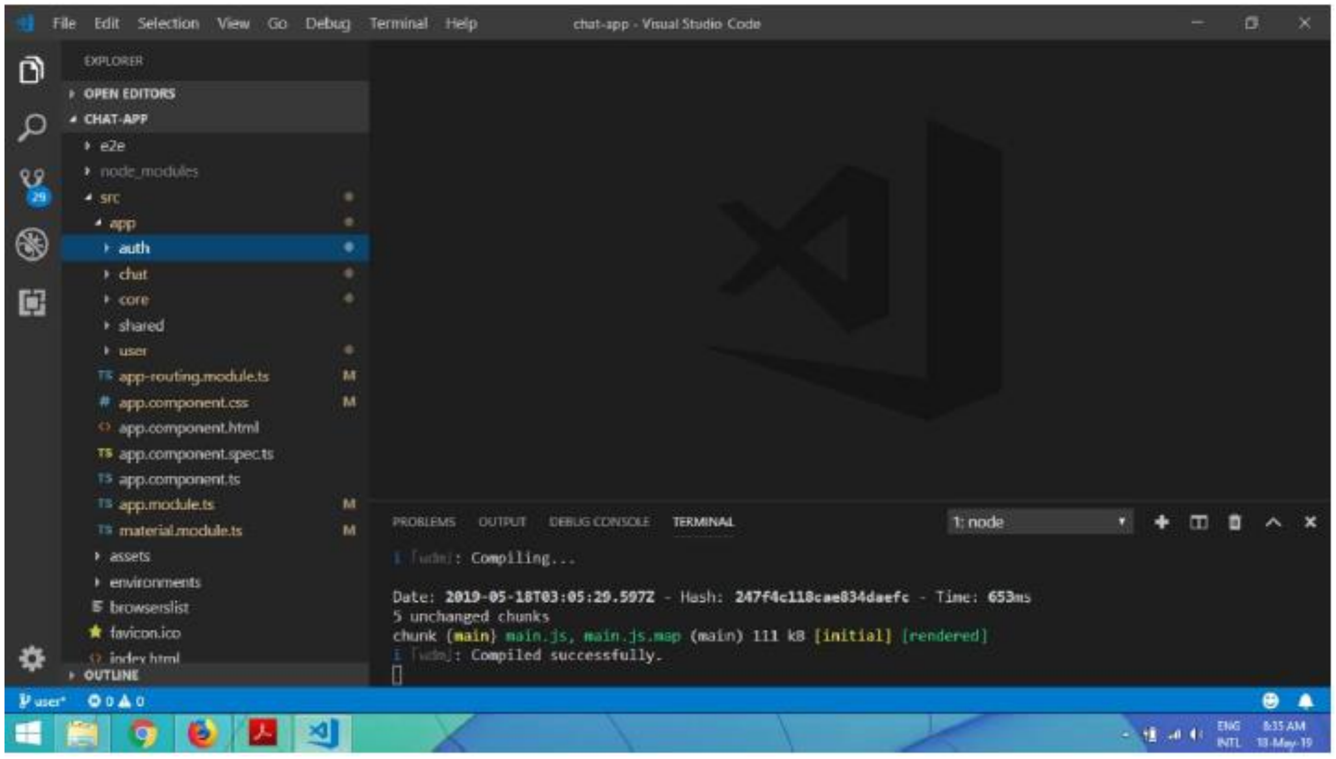


Hey
May 18, 2019, 8:28:30 AM

Hello
May 18, 2019, 8:28:44 AM

How are you?
May 18, 2019, 8:37:37 AM

Chat has ended



```
i ｢wdm｣: Compiling...

Date: 2019-05-18T03:05:29.597Z - Hash: 247f4c118cae834daefc - Time: 653ms
5 unchanged chunks
chunk {main} main.js, main.js.map (main) 111 kB [initial] [rendered]
i ｢wdm｣: Compiled successfully.
```

Challenges

1. Digital Assistant
    - There was a problem with the Google Chrome not allowing this application to make multiple Http request from a single page, so I had to enable Cross-origin resource sharing.

    - Making a Http request then collecting the result and then again making a request to the collected result from the previous request. I solved it by introducing a wait after the first request until the second request is made.

2. Attendance Monitor System
    - The main challenge with this project was making the project dynamic as the data is changing and according to that the DOM needs to be rendered properly. Careful placing of loop control and render control was the solution to it.

    - The home page rendering. It was a single page displaying different buttons, so they needed to get the rendering data from the backend. So, I made changes to the backend API to fit my need and return an array of String that is associated to my actor.

3. Web based chat application
    - Here the challenge was to implement the front-end without a back-end to which I can flexibly interact to. Obviously, Firebase had its own features like I don't have to handle the login and signup which was plus. But it was like I can't change it according to what I like.

    - The FireBase is a tool that is in development and reliable support is not available for it. So, a challenge was to bring all the elements together and hope they work with each other. Version matching of different parts were a big hassle.