A Project Report

ON

CONNECT PES

BY

Sharath KP – 1PI13CS141

Shreyas G – 1PI13CS153

Nagashree AC – 1PI13CS201

GUIDE
Prof.Raghu B A Rao
PESIT-CSE
Bangalore

August 2016 – December 2016

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PES INSTITUTE OF TECHNOLOGY
(an autonomous institute under VTU)
100 FEET RING ROAD, BANASHANKARI III STATE
BANGALORE-5600

PES Institute of Technology,
(An Autonomous Institute under VTU)
100 Feet Ring Road, BSK 3rd Stage, Bangalore-560085

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Special Topics: Mini Project work entitled **CONNECT PES** is a bona-fide work carried out by **(Sharath KP – 1PI13CS141, Shreyas G – 1PI13CS153, Nagashree AC – 1PI13CS201)** in partial fulfilment for the award of degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the academic semester August 2016 to December 2016.

Signature of the Guide	Signature of the HOD
Prof. Raghu B A Rao	**Prof. Nitin .V. Pujari**

Sharath KP – 1PI13CS141

Shreyas G – 1PI13CS153

Nagashree AC– 1PI13CS201

ABSTRACT

Every interaction inside college happens though social media or through verbal communication between the students and teachers. The question papers, doubt clarifications, notes sharing all happens through the Google server or other means, which indirectly reaches out of college. The need to do all these within our college server exits. Also, generally due to the hectic schedule and pressure offered by an undergraduate course, a student will crumble under pressure and become stressed. This might result in the student losing track of important stuff and becoming disorganized.

So we aim to present a website which will help students to stay updated and help him to organize things better. Using our website, he can contact a faculty for doubt clarification or post a query. The chat room features enables a student easy access to a faculty or his friends. He can also maintain all his notes of various subjects in the website. There is also an in-built Twitter like social network where in the students can post about the latest happenings in college. Hence by our website, we want to provide and ensure that all the important things that a student needs will be at his fingertips.

We also have a todo list, which helps students to be more organised. Also makes him/her aware of every work and its deadline. Booking an appointment with the faculty makes project evaluation, doubt clarification go in a more organized manner. Students need not wait for hours to get faculty appointment. Hence aim to solve these with our Connect PES website.

<u>ACKNOWLEDGEMENT</u>

We would like to express our sincere gratitude to all the people who have been associated with this project directly or indirectly and have helped us with working on it.

We would like to extend our sincere thanks to **Prof. M. R. Doreswamy**, founder of PESIT, **Prof. D. Jawahar**, CEO of PESIT and **Dr. K.S. Sridhar**, Principal PESIT, for the encouragement, motivation and continuous support they have provided to us.

We would like to thank **Prof. Nitin V. Pujari,** Head of the Department CSE for his constant support and encouragement.

We would also like to thank **Prof. Raghu B A Rao**, an esteemed Faculty of CSE Department and also our guide, for providing us an opportunity to work on a novel concept, and also for his guidance throughout the duration of this project. We would also want to thank him for his continuous inputs and ideas for this project.
We would also like to thank our parents and other family members for their continuous support to us, without which, we would not have been able to achieve whatever we have.

We would also like to thank all our friends who have directly or indirectly helped us throughout the duration of the project.

Table of Contents

1. INTRODUCTION

College life is one of the most enjoyable and vibrant phases in a person's life. But along with all the joys, it also brings lot of stress and pressure to a student. Same goes to the professors as well. They would have a hectic schedule of seminars, project reviews and classes to balance. Due to this, there might develop a communication gap between the professors and the students which might result in the students not getting proper guidance .So we aim to bridge that gap between them and also aim at getting the students more organised.

Technology is changing the way various spheres of our society work including several traditional aspects of education. Some studies show that a better learning experience is created for the students if the professors engage to them on their preferred platform email, text messaging, social media. These are the preferred methods for students to engage nowadays. So what if we provided a common platform online for the students and professors to communicate.

Hence, we decided to design a website in which there would be special chat rooms where the students and professors can communicate easily. A student can join any chat room interact with any faculty related to his subject of interest. Also the student can maintain all his notes online, stay updated with all the latest happenings on the college though the mini social network embedded on our website.

2. PROBLEM DEFINITION

Design a website where in the students will be able to login to his account and access the features. If he doesn’t have an account, he should be able to sign up to the website and get an account. Each faculty also have an account and must login or register through that account.

Chat room features must be provided for the professors and students to communicate. A student must be able to leave and join his desired chat room, based on the subject he likes and also create his own room based on the interest. Also he is free to post any messages or queries he has in the chat room.

The faculty too has access to the chat room and can create his one room for the subject he teaches, reply, and answer to the query of students through the room itself. He can post important announcements such as assignment submission and quiz dates through the chat room.

The student will also be able to take down notes of different subjects he listens to classes. The faculty too can save notes and important points useful for the subject or other. These notes should be saved in his account and displayed even if he logs out and logs in again.

Similar to this, there should be a To-do list where the student and faculty can maintain list of important things he or she has to attend to and tends to forget due to busy schedule.

Also, the web application has a booking system to schedule meeting with faculty and a social network where in students and faculty can post updates about the events on college.

3. LITERATURE SURVEY

At the beginning of our project, we carried out a search on the existing ways different colleges use technology to communicate within the college. We found that there are many colleges which use their own platform for purposes like chatting, doubt clarification, note etc. Hence for helping the faculty and students, we needed a tool for better communication between them and getting them organized is very much needed.

Also all the communications in college happens through the google server, Gmail accounts, whatsapp group etc. All these other technologies makes the college intellectual property to go out into others server. Which makes the data highly unreliable. Hence we need to use PES sever for all these communications.

We went through our present college website and found out it is just a static website without any extra features for the students. The same website is used for pubic people too. Students need portal which makes their life easier, simple and organised. Currently there is no such portal for PES students where they can easily interact with teachers, makes their life simpler with notes and todo lists.

Faculty are generally busy with too many works which makes them available for only limited time and only for few students.
Hence we need an appointment system to book time with the faculty so that all students need not wait and easily contact the faculty.

We went through various websites to learn the fundamentals of MEAN stack and MVC Structure such as quora.com, mongodb.com, expressjs.com, angularjs.org. We also saw various YouTube tutorials to get a grasp over MEAN technology.

4. <u>PROJECT REQUIREMENT DEFINITION</u>

The usage of this project is mainly for the communication between students and teacher of PES College without using any other social platforms.

A cloud platform such as Microsoft Azure or Google is required to host the database and node Js server, so that all users get and update data synchronously, which can easily be scalable to many users.

Any person who knows to use social platforms can easily our application. Any student of PES University with an account in our website and an internet connection can access the features of the website easily.

The student or teacher must have an account to login and access the features. If he/she doesn’t have an account, then he can sign up and create an account.

The user must have the knowledge to use features such as saving notes, todo list and chat which will help both student as well the teacher.

5. SYSTEM REQUIREMENT SPECIFICATION

The system hosting the web application must have all the components of MEAN stack installed, which will be mongoDB, express.js, angularJS, and node.js.

The system must have the required memory to store details of all users. Upon scaling the application and increase of number of users, the memory to store their details must also be increased.

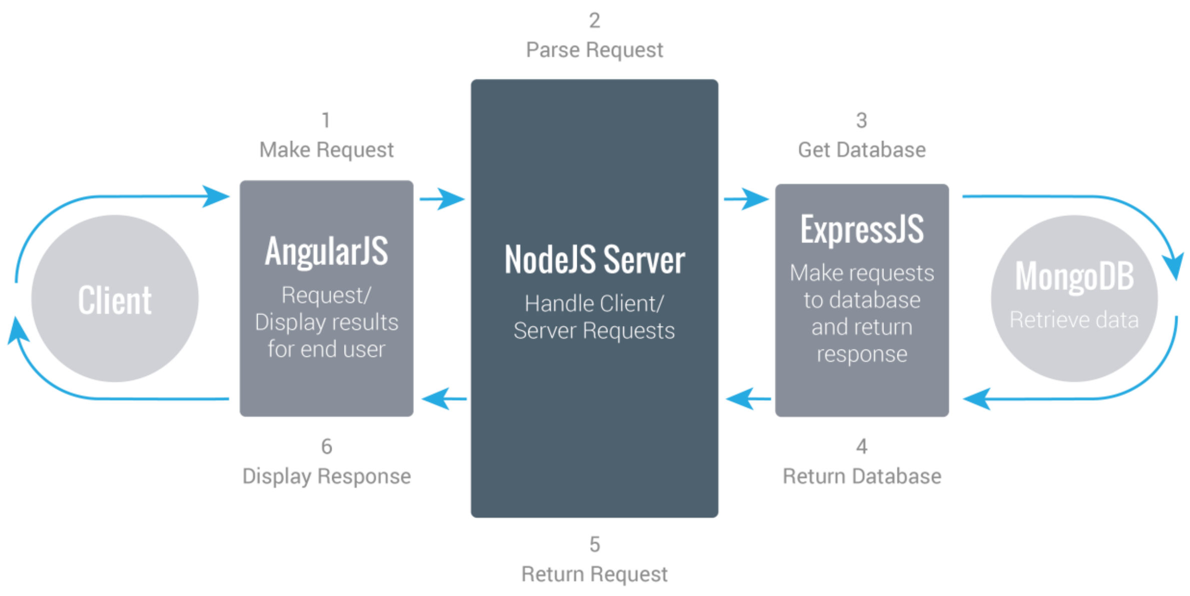
The system required noSql database mongoDB to store the user’s data. The minimum version of 3.4.0 is needed to run the application. Express is used for routing and requires 4.9.0 version.

Node Js is needed with the latest stable release and Angular JS with the latest version or its CDN is needed.

The system also required other major dependencies like body-parser, cookie-parser, ejs, express-session, mongoose and passport to run the web application fluently.

6. DESIGN

Our website follows the MVC software design pattern. The main tools we have used to build our website is MEAN Stack.



The above diagram depicts the flow of the website using MEAN.

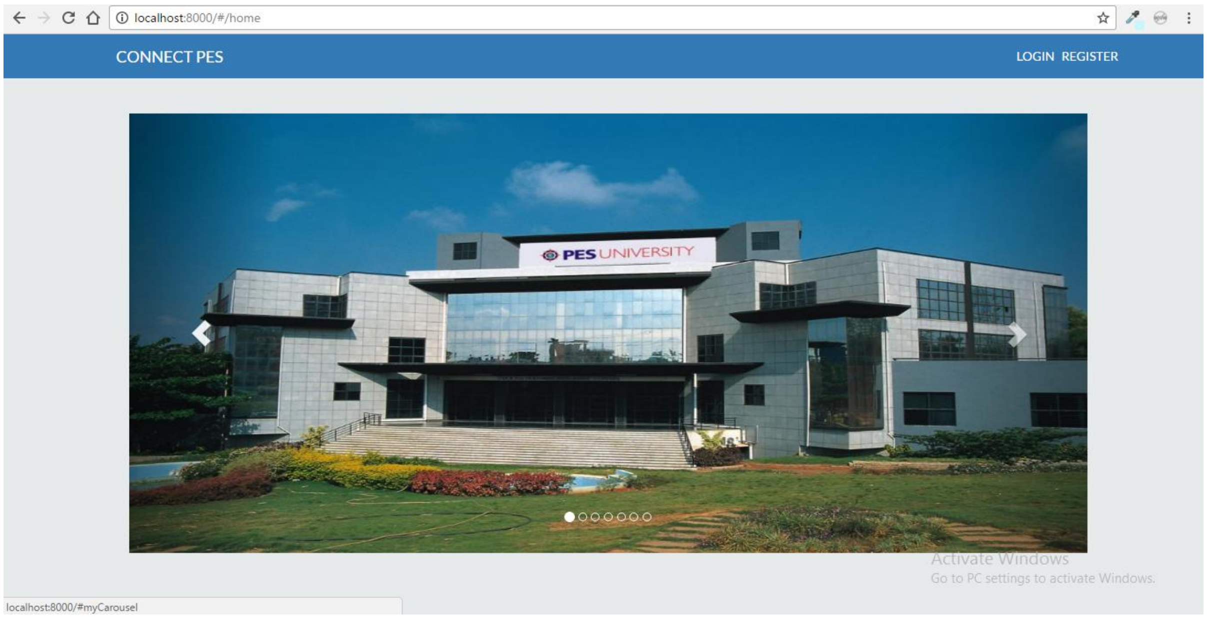
MEAN is a free and open-source JavaScript software stack for building dynamic web sites and web applications.

The MEAN stack is MongoDB, Express.js, Angular, and Node.js. Because all components of the MEAN stack support programs are written in JavaScript, MEAN applications can be written in one language for both server-side and client-side execution environments. It has emerged as the most superior software stack to develop web apps. As it allows us to incorporate various features and provide a better user experience and user interaction. Its role in connecting the modern world and providing solution to real world problems is immense. One of the driving factors for the popularity of MEAN is the ability to use JavaScript both in frontend and backend.

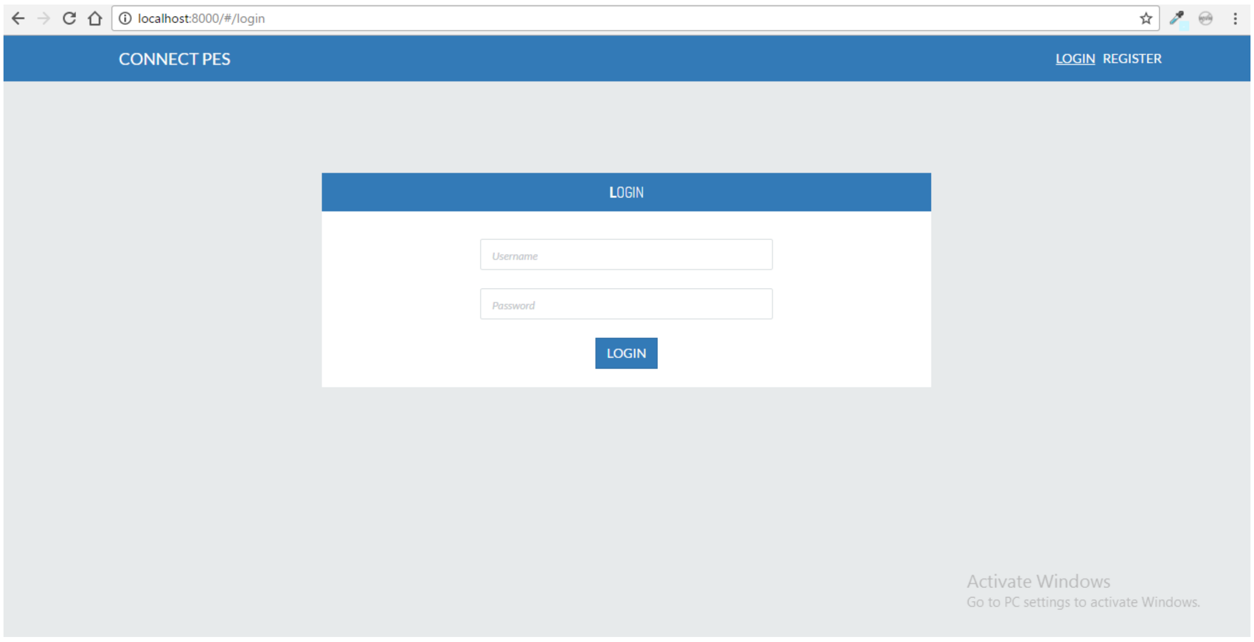
So by using this technology we aim to build an interactive and effective website which will revolutionize the various traditional methods of the education field.

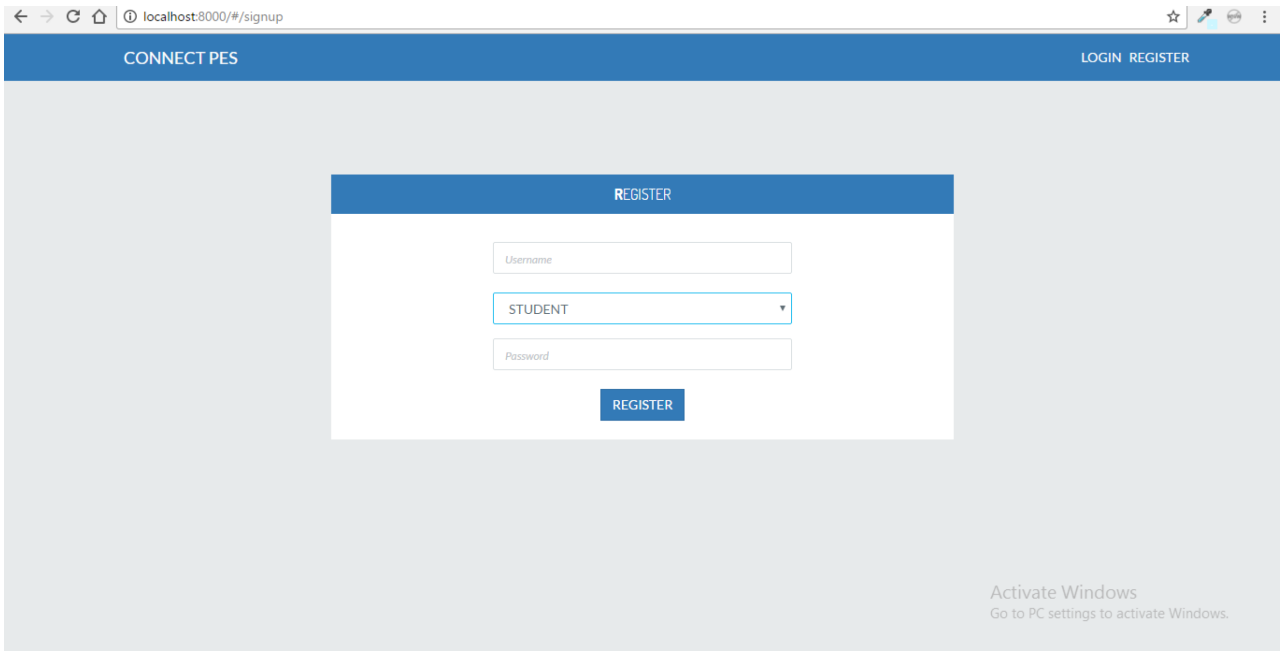
The flow of our website starts with login/register, which follows with the features Tweet, Notes, Chat, Todo list and Appointment.

Initially the home page loads up.



Following this will be the login/ register pages.

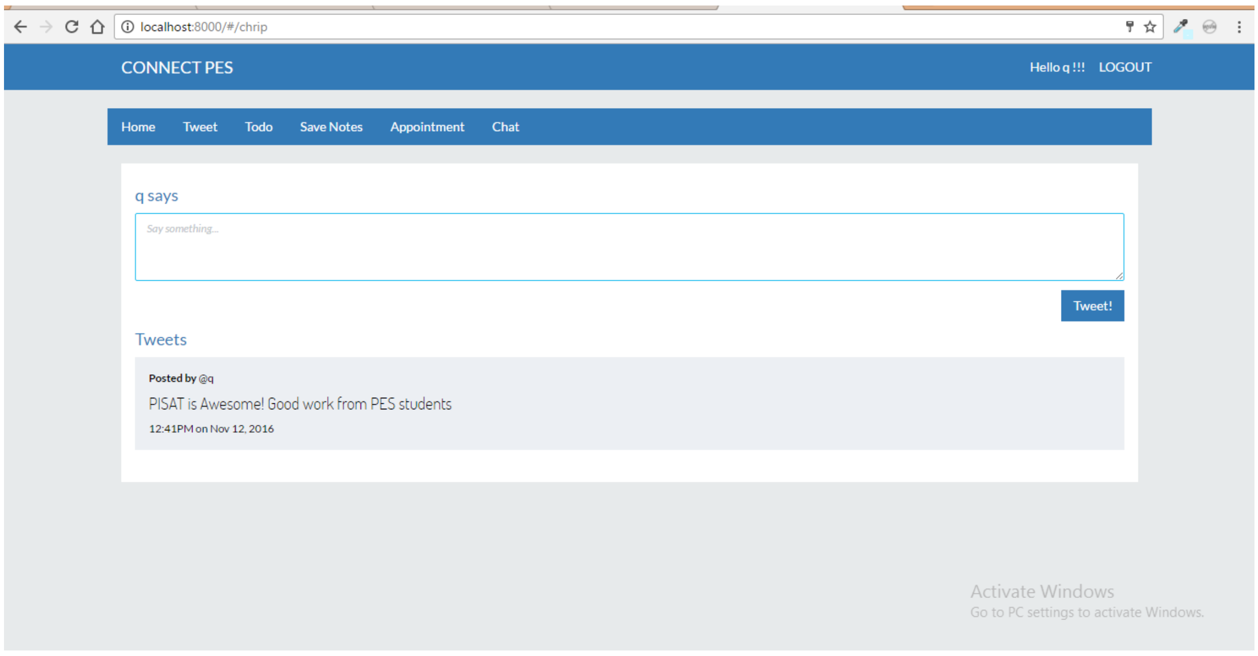




Consider a scenario where in a student with email student@gmail.com uses the website. On opening the website, a login page appears. On login, the page is redirected to home page of the website from which he can access the features of the website

Different navigation tabs are provided in the home page. On clicking on these tabs the page navigates to the activity mentioned on it.

The PES tweet feature is as shown below, where any student or teacher can tweet about latest updates in college as well as express their opinion about it.

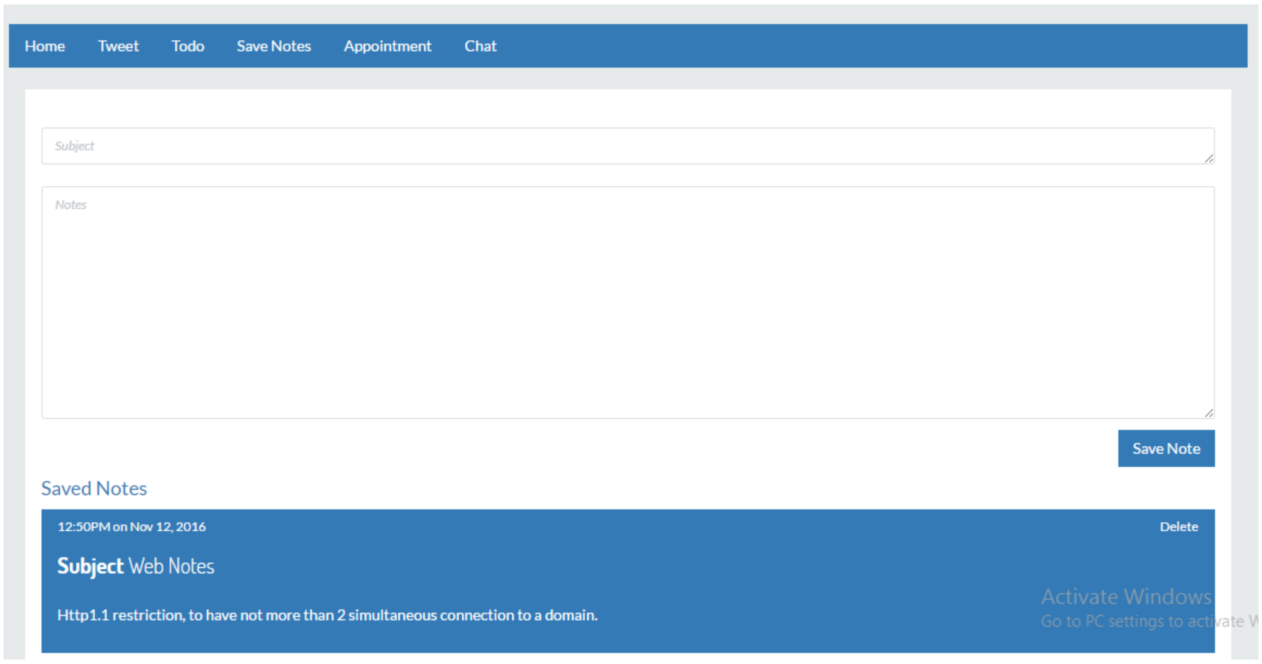


Posted by @q

PISAT is Awesome! Good work from PES students

12:41PM on Nov 12, 2016

The Notes page, to take down notes for any user is as show below.



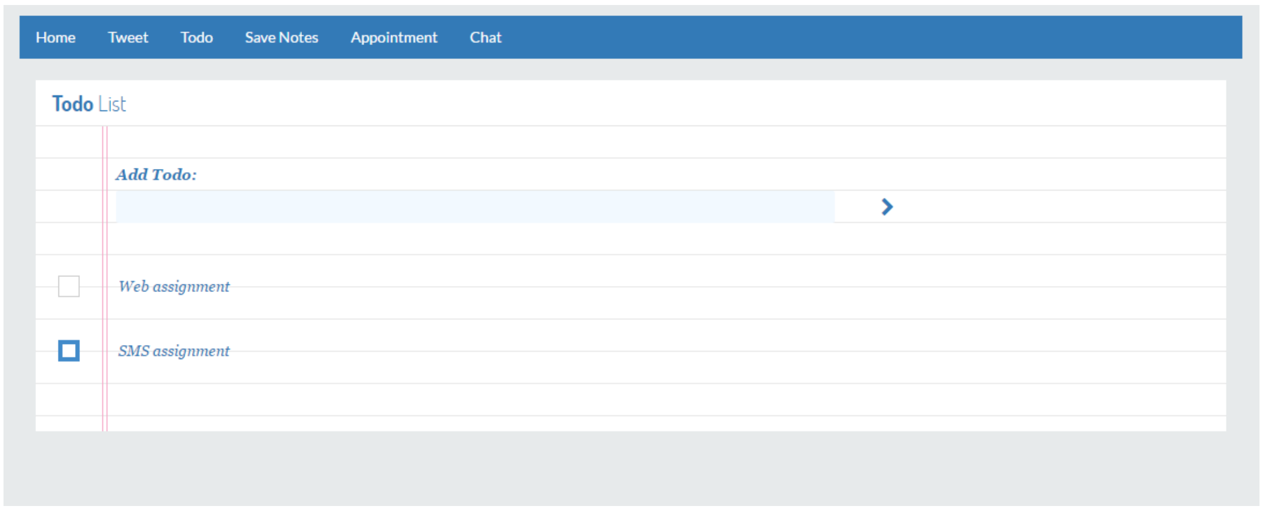
12:50PM on Nov 12, 2016

Subject Web Notes

Http1.1 restriction, to have not more than 2 simultaneous connection to a domain.

The user can store his necessary notes giving the main subject and the contents of the notes. These notes are stored on server and can be accessed by the user until he is a member of the college or he chooses to delete it.

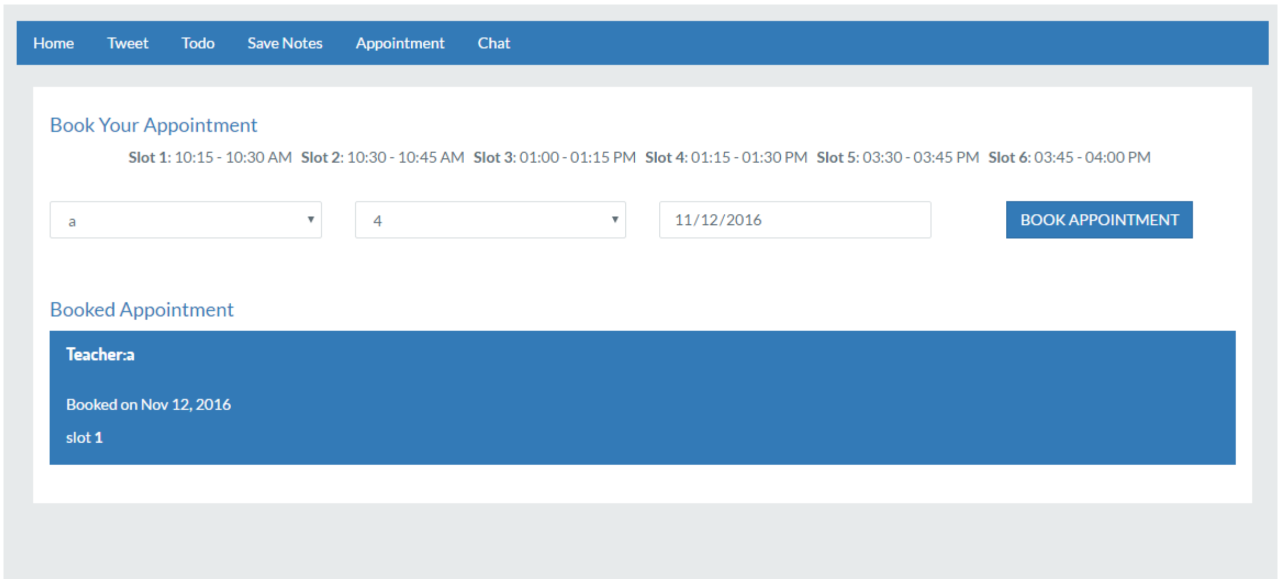
The next feature is a todo list where both students and teachers can store in a list of activities they should complete. These activities can be checked out once done. They can be unchecked again for further use or can be deleted. The todo list page is as shown below where any user can take down his/her list.



☐ *Web assignment*

☐ *SMS assignment*

The appointment scheduling with teachers is divided into 6 slots as shown. A student can choose to book any of the day and slot number based on availability. If slot is already booked it is shown as booked to the user. The teacher however can only see the appointments the students have taken with them.



Book Your Appointment

Slot 1: 10:15 - 10:30 AM **Slot 2:** 10:30 - 10:45 AM **Slot 3:** 01:00 - 01:15 PM **Slot 4:** 01:15 - 01:30 PM **Slot 5:** 03:30 - 03:45 PM **Slot 6:** 03:45 - 04:00 PM

a | 4 | 11/12/2016

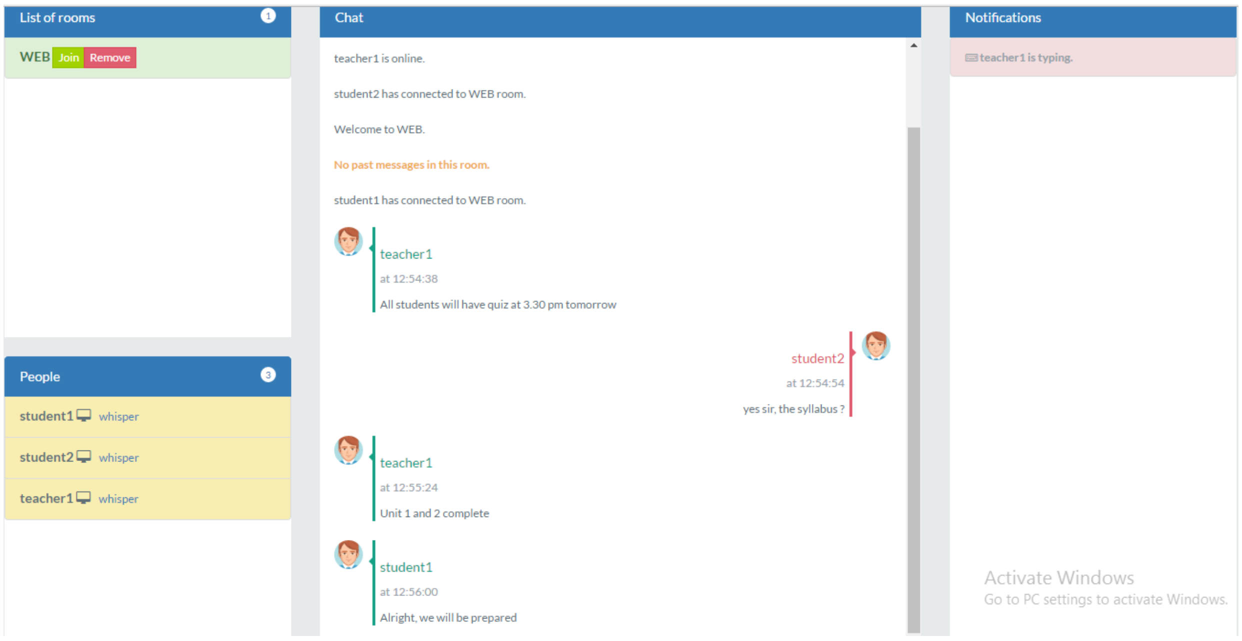
Booked Appointment

Teacher:a

Booked on Nov 12, 2016

slot **1**

The Chat page is such that it's in a new window and the other features of the website can be used with the chat application.



student1 whisper
student2 whisper
teacher1 whisper

teacher1 is online.

student2 has connected to WEB room.

Welcome to WEB.

No past messages in this room.

student1 has connected to WEB room.

teacher1
at 12:54:38
All students will have quiz at 3.30 pm tomorrow

student2
at 12:54:54
yes sir, the syllabus ?

teacher1
at 12:55:24
Unit 1 and 2 complete

student1
at 12:56:00
Alright, we will be prepared

teacher1 is typing.

The student can join a particular subject's group and chat with other students and teacher related to the subject. On joining a room the previous messages are shown to the user. A mini notification of who is typing is also shown. A provision to send a private message to any particular user is also available.

7. PSEUDO CODE

```
// app.js
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
varcookieParser = require('cookie-parser');
varbodyParser = require('body-parser');
var session = require('express-session');
var passport = require('passport');
varmethodOverride = require('method-override');

//initialize mongoose schemas
require('./models/models');

var index = require('./routes/index');
varapi = require('./routes/api');
var authenticate = require('./routes/authenticate')(passport);
var mongoose = require('mongoose');                    //add
for Mongo support
mongoose.connect('mongodb://localhost/chatapp');
//connect to Mongo
var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

// uncomment after placing your favicon in /public
//app.use(favicon(__dirname + '/public/favicon.ico'));
app.use(logger('dev'));
app.use(session({
secret: 'keyboard cat'
}));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
app.use(passport.initialize());
app.use(passport.session());
app.use(methodOverride());

app.use('/', index);
app.use('/auth', authenticate);
app.use('/api', api);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
var err = new Error('Not Found');
err.status = 404;
next(err);
});

//// Initialize Passport
varinitPassport = require('./passport-init');
initPassport(passport);

// development error handler
// will print stacktrace
if (app.get('env') === 'development') {
app.use(function(err, req, res, next) {
res.status(err.status || 500);
res.render('error', {
message: err.message,
error: err
        });
    });
}

// production error handler
// no stacktraces leaked to user
app.use(function(err, req, res, next) {
```

```
res.status(err.status || 500);
res.render('error', {
message: err.message,
error: {}
   });
});


module.exports = app;

//authenticate.js
var express = require('express');
var router = express.Router();

module.exports = function(passport){

      //sends successful login state back to angular
      router.get('/success', function(req, res){
            res.send({state: 'success', user: req.user ? req.user
: null});
      });

      //sends failure login state back to angular
      router.get('/failure', function(req, res){
            res.send({state: 'failure', user: null, message:
"Invalid username or password"});
      });

      //log in
      router.post('/login', passport.authenticate('login', {
            successRedirect: '/auth/success',
            failureRedirect: '/auth/failure'
      }));

      //sign up
      router.post('/signup', passport.authenticate('signup', {
            successRedirect: '/auth/success',
            failureRedirect: '/auth/failure'
      }));

      //log out
      router.get('/signout', function(req, res) {
            req.logout();
            res.redirect('/');
      });

      return router;

}

//modals.js
var mongoose = require('mongoose');

varpostSchema = new mongoose.Schema({
      created_by: String,         //should be changed to ObjectId,
ref "User"
      created_at: {type: Date, default: Date.now},
      text: String
});

varuserSchema = new mongoose.Schema({
      username: String,
      usertype: String,
      password: String, //hash created from password
      created_at: {type: Date, default: Date.now}
})

varnoteSchema = new mongoose.Schema({
      username: String,
      created_at: {type: Date, default: Date.now},
      note_header: String,
      note_body: String
})
```

```
vartodoSchema = new mongoose.Schema({
	username: String,
	text: String
})

varappointmentSchema = new mongoose.Schema({
	username: String,
	teacher: String,
	slot: String,
	booked: Date
})


mongoose.model('Post', postSchema);
mongoose.model('User', userSchema);
mongoose.model('Note', noteSchema);
mongoose.model('Todo', todoSchema);
mongoose.model('Appointment', appointmentSchema);
```

8. RESULTS

The web application works as needed and gives the right actions and functionalities required by the user. The login and register work perfectly accessing the database and allowing only authenticated users into the website.

The PES tweet shows in the tweet from every user and also stores it in the database and retrieves it for all users without errors. The students and teachers access to notes also works properly storing the notes subject and contents into the server inside the DB. The unwanted notes are also deleted and removed from the server.

Chat between the users works efficiently and shows every message given by user. The chat history also shows previous messages correctly to tell the user about what discussions is going on in that subject room. The whisper feature sends the message correctly to the given user and not others.

Appointments are booked using student account and shown in the teacher's account. The appointment slots already booked cannot be booked again and it is notified to the user.

9. CONCLUSION

After good four months that we spent on the project studying about MEAN stack, its features and applications, we are very happy with the web application we have developed.

The website is able to handle a good amount of load and run its features without any errors. We have achieved, to some extent, what we had hoped to achieve.

We have also started taking initiative to add more features into the website and expand the accessibility and utility of the application.

MEAN stack usage for website is very advantageous and is a vital step which could be takes to develop fast responsive and high performance websites.

The website can be hosted and given access to students and teachers which could be the next improvement in college. With having great features it will be a very useful application in the hands of students and teachers.

10. FUTURE ENHANCEMENTS

Making the website scalable to more number of users, hosting it and making it as a part of college is the next step in the project. Extending the features available already and upgrading them for better usage is our next aim.

The notes and todo list feature now available will be upgraded to have a share option where students and teachers can share them to other users. An option to not only view in website but also to download can be provided.

The mini tweet will be extended to even support images and videos monitored by admin to even handle irrelative contents can be a future upgrade.

The chat can be also extended to have better UI with extra features to search, share and upload files as well. Bringing in new features such as club pages, placement schedule, experiences, helper tutorials and access to video tutorials of college are also part of future enhancements of this project.

11. BIBLIOGRAPHY

1. Mean stack Tutorial by Microsoft Virtual Academy https://www.youtube.com/watch?v=Jh0er2pRcq8
2. Full Stack Javascript Development With Mean by Adam Bretz, Colin J Ihrig
3. Node.js in Action by Marc Harter, Mike Cantelon, Nathan Rajlich, and T. J. Holowaychuk
4. The Definitive Guide to MongoDB by David Hows , Eelco Plugge , Peter Membrey , Tim Hawkins.
5. Mean Web Development by Amos Q Haviv
6. Express.js: Web App Development with Node.js Framework by Ralph Archer.