## **Summer Training Report**

## **On**

**Online Notice Board Application**

## **Submitted**

## **In The Partial Fulfillment Of**

## **Bachelor of Technology**

## **(Computer Science Engineering)**

## **Submitted by:**

## **Uzma Ahmed**

## **Roll No.-00520907216**

# C:\Users\Shehzad\Downloads\download.jpg

DEPT. OF COMPUTER SCIENCE

GOVIND BALLABH PANT ENGINEERING COLLEGE

NEW DELHI-110020

2015-2019

**ACKNOWLEDGEMENT**

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior, and acts during the course of study.

I am thankful to **Mrs. Sunita Kumari** for his support, cooperation, and motivation provided to me during the training for constant inspiration, presence and blessings.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of our college faculty members of CSE Department for their encouragement & guidance.

Lastly, I would like to thank the almighty and my parents for their moral support and my friends with whom I shared my day-to-day experience and received lots of suggestions that improved my quality of work.

**CANDIDATE DECLARATION**

I, **UZMA AHMED**, Roll No. 00520907216, B.Tech(Semester-5th) of Govind Ballabh Pant Engineering College, New Delhi hereby declare that the Training Report entitled “**Online Notice Board Application**” is an original work and data provided in the study is authentic to the best of my knowledge.This report has not been submitted to any other institute for the award of any other degree.

Place: New Delhi **Uzma Ahmed**

Date: 12.11.2017 (Roll No: 00520907216)

CONTENTS

1. **INTRODUCTION 6**
   1. Introduction to Organisation
   2. Introduction to Project
   3. Technology and Framework

1.3.1 Technology

1.3.1.1 Android Platform

1.3.1.2 Java Script

1.3.1.3 Node.js

1.3.2 Framework

1.3.2.1 Angular JS

1.3.2.2 Cordova

1.3.2.3 Express

* + - 1. Ionic

1. **REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION 16** Software Requirement Specification (SRS)
   1. Data Requirement
   2. Functional Requirement

2.2.1 User Registration

2.2.2 User Login

2.2.3 Dashboard

2.2.4 Search Notice

2.2.5 Post Notice

2.2.6 Delete Notice

2.3 Performance Requirement

1. **PROJECT DESIGN 18**
   1. Use Case Diagram
   2. ER Diagram
2. **RESULT 19** 
   1. Snapshots of Application
3. **REFRENCE 26**

**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction to Organisation**

I had my two months industrial training at Vitarka Solutions which is the world’s leading progressive IT solutions company with a unique focus on innovation in technology, leadership and design, started by an IIT Kanpur alumni in 2010.

Today, Vitarka Solutions is one of the pioneers in providing Smartphone Applications. Their ideology is to be a consultant to the customer rather being a mere seller. They have developed many applications in Education sector.   
In collaboration with Delhi Integrated Multi-Modal Transit System Ltd (DIMTS) they have developed applications in transportation sector also.

Good thing about Vitarka Solutions is that it is driven by ethical values, energy to build great apps and determination to deliver the best. They are keen to provide out of box solutions to the client keeping the cost to company aside. Vitarka Solutions teams are trained to work under various software models such as Agile, Scrum etc. They make sure that they adhere to these standards so that they are able to deliver quality product. Vitarka Solutions innovate and build applications using latest technologies, which makes them fast, robust and scalable. This company contributes to open source software by organizing/participating in OSS Camps.

Vitarka Solutions believes that “communication” is the foremost ingredient to make a project successful. It communicate with the clients on regular basis and using tehe dedicated VOIP lines and various other ways of communication. This company understand different needs of different customers and hence have custom solutions based on the requirements. Their onsite team facilitation is one such example.

**1.2 Introduction to Project**

The purpose of the online notice board application is to display schools and colleges notices through an application anywhere.

It’s a platform for posting notices that can be viewed by the people who are using that application for example, announcing events , sharing syllabus , date sheet , information related to sports or any other classroom activity etc.

The app would be focusing on students who face problems regarding sharing of any relevant message. This application is made only for notice purpose, as it would always be cumbersome sharing information or important notice by e-mails, or through whatsapp messages. The user can perform functions such as postings and  view notices. He/She might edit, delete and update their own notices.

It will save time, keep students informed & can serve for variety of purposes.

The  application will help students to share notice or any other important stuff which they want to share. The user should first signed in, after verification of users, he/she would be able to view the notices shared by other members of the app.

Users would be provided with a dashboard where they can edit their profile. The notices posted by the user would be displayed on their dashboard too so that they would be able to delete or update it. If user logout, he won’t  be able to use the app.

The online Notice Board Application is an exciting new innovative way to send information to people like staffs, teachers, students etc. Replace paper Notice board with an online notice board to speed up the process of communication to an instant. This Application helps the students to retrieve all the notices and articles directly through their cell phones.

The purpose of online Notice board is to eliminate the paper work from the university environment. Due to this new idea everybody in the university will be timely updated from important announcement and notices.

**1.3 Technology and Framework**

**1.3.1 Technology**

**1.3.1.1 Android Platform**

**Android**

Android is basically an operating system for smartphones. But we find now integrated into PDAs, touch pads or televisions, even cars (trip computer) or netbooks. The OS was created by the start-up of the same name, which is owned by Google since 2005 .

Android consists of a kernel based on the Linux kernel, with middleware, libraries and APIs written in C and application software running on an application framework which includes Java-compatible libraries based on Apache Harmony. Android uses the Dalvik virtual machine with just-in-time compilation to run compiled Java code. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices.

Developers write primarily in a customized version of Java. There are currently more than 520,000 apps available for android apps can be downloaded from third-party sites or through online stores such as Android Market, the app store run by Google.

**1.3.1.2 Java Script**

**JavaScript**

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers. The ECMA-262 Specification defined a standard version of the core JavaScript language.

• JavaScript is a lightweight, interpreted programming language.

• Designed for creating network-centric applications.

• Complementary to and integrated with Java.

• Complementary to and integrated with HTML.

• Open and cross-platform.

**Client-Side JavaScript**

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

The merits of using JavaScript are:

• Less server interaction: You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.

• Immediate feedback to the visitors: They don't have to wait for a page reload to see if they have forgotten to enter something.

• Increased interactivity: You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.

**1.3.1.3 Node.js**

**Node.js**

Node.js is an open-source server side runtime environment built on Chrome's V8 JavaScript engine. It provides an event driven, non-blocking (asynchronous) I/O and cross-platform runtime environment for building highly scalable server-side application using JavaScript.

Node.js can be used to build different types of applications such as command line application, web application, real-time chat application, REST API server etc. However, it is mainly used to build network programs like web servers, similar to PHP, Java, or ASP.NET.

Node.js was written and introduced by Ryan Dahl in 2009.

**Advantages of Node.js**

1. Node.js is an open-source framework under MIT license. (MIT license is a free software license originating at the Massachusetts Institute of Technology (MIT).)
2. Uses JavaScript to build entire server side application.
3. Lightweight framework that includes bare minimum modules. Other modules can be included as per the need of an application.
4. Asynchronous by default. So it performs faster than other frameworks.
5. Cross-platform framework that runs on Windows, MAC or Linux

## **NPM: The Node Package Manager**

Node.js have built-in support for package management using the [NPM](http://docs.nodejitsu.com/articles/getting-started/npm/what-is-npm) tool that comes by default with every Node.js installation. The idea of NPM modules is quite similar to that of Ruby Gems*:* a set of publicly available, reusable components, available through easy installation via an online repository, with version and dependency management.

A full list of packaged modules can be found on the [npm website](https://npmjs.com/" \t "_blank), or accessed using the npm CLI tool that automatically gets installed with Node.js. The module ecosystem is open to all, and anyone can publish their own module that will be listed in the npm repository.

**1.3.2 Framework**

**1.3.2.1 AngularJS**

**AngularJs**

AngularJS is an open source web application framework. It was originally developed in 2009 by Misko Hevery and Adam Abrons. It is now maintained by Google. Its latest version is 1.4.3.

## **Features**

* AngularJS is a powerful JavaScript based development framework to create RICH Internet Application(RIA).
* AngularJS provides developers options to write client side application (using JavaScript) in a clean MVC(Model View Controller) way.
* Application written in AngularJS is cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.
* AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache License version 2.0.

Overall, AngularJS is a framework to build large scale and high performance web application while keeping them as easy-to-maintain.

## **Core Features**

Following are most important core features of AngularJS −

* **Data-binding** − It is the automatic synchronization of data between model and view components.
* **Scope** − These are objects that refer to the model. They act as a glue between controller and view.
* **Controller** − These are JavaScript functions that are bound to a particular scope.
* **Services** − AngularJS come with several built-in services for example $https: to make a XMLHttpRequests. These are singleton objects which are instantiated only once in app.
* **Filters** − These select a subset of items from an array and returns a new array.
* **Directives** − Directives are markers on DOM elements (such as elements, attributes, css, and more). These can be used to create custom HTML tags that serve as new, custom widgets. AngularJS has built-in directives (ngBind, ngModel...)
* **Templates** − These are the rendered view with information from the controller and model. These can be a single file (like index.html) or multiple views in one page using "partials".
* **Routing** − It is concept of switching views.
* **Model View Whatever** − MVC is a design pattern for dividing an application into different parts (called Model, View and Controller), each with distinct responsibilities. AngularJS does not implement MVC in the traditional sense, but rather something closer to MVVM (Model-View-ViewModel). The Angular JS team refers it humorously as Model View Whatever.

## **Concepts**

Following diagram depicts some important parts of AngularJS :

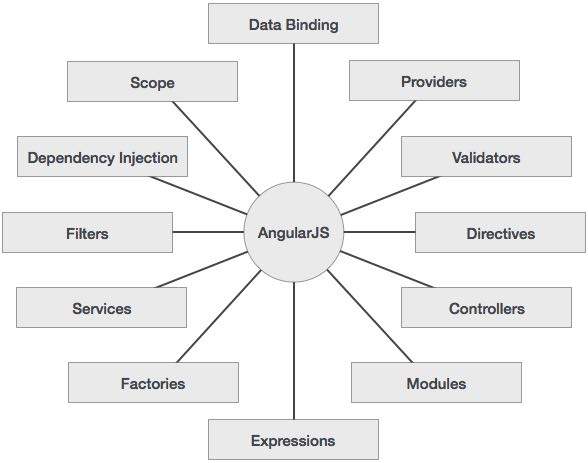


Figure 1.1: Features of AngularJS

## **AngularJS Components**

The AngularJS framework can be divided into following three major parts −

* **ng-app** − This directive defines and links an AngularJS application to HTML.
* **ng-model** − This directive binds the values of AngularJS application data to HTML input controls.
* **ng-bind** − This directive binds the AngularJS Application data to HTML tags.

**1.3.2.2 Cordova**

**Cordova**

Cordova is a platform that is used for building mobile apps using HTML, CSS and JS. We can think of Cordova as a container for connecting our web app with native mobile functionalities. Web applications cannot use native mobile functionalities by default. This is where Cordova comes into picture. It offers a bridge for connection between web app and mobile device. By using Cordova, we can make hybrid mobile apps that can use camera, geolocation, file system and other native mobile functions.

## **Cordova Features**

The features of Cordova in brief:

* **Command Line Interface (Cordova CLI)**

This tool can be used for starting projects, building processes for different platforms, installing plugins and lot of other useful things that make the development process easier. You will learn how to use the Command Line Interface in the subsequent chapters.

### **Cordova Core Components**

Cordova offers a set of core components that every mobile application needs. These components will be used for creating base of the app so we can spend more time to implement our own logic.

### **Cordova Plugins**

Cordova offers API that will be used for implementing native mobile functions to our JavaScript app.

### **License**

Cordova is licensed under the Apache License, Version 2.0. Apache and the Apache feather logos are trademarks of The Apache Software Foundation.

**1.3.2.3 Ionic**

**Ionic** is a front-end HTML framework built on top of **AngularJS** and **Cordova**. As per their official document, the definition of this Ionic Open Source Framework is as follows −

Ionic is an HTML5 Mobile App Development Framework targeted at building hybrid mobile apps. Think of Ionic as the front-end UI framework that handles all the look and feel and UI interactions your app needs to be compelling. Kind of like "Bootstrap for Native", but with the support for a broad range of common native mobile components, slick animations and a beautiful design.

## **Ionic Framework Features**

Following are the most important features of Ionic −

* **AngularJS** − Ionic is using AngularJS MVC architecture for building rich single page applications optimized for mobile devices.
* **CSS components** − With the native look and feel, these components offer almost all elements that a mobile application needs. The components’ default styling can be easily overridden to accommodate your own designs.
* **JavaScript components** − These components are extending CSS components with JavaScript functionalities to cover all mobile elements that cannot be done only with HTML and CSS.
* **Cordova Plugins** − Apache Cordova plugins offer API needed for using native device functions with JavaScript code.
* **Ionic CLI** − This is NodeJS utility powered with commands for starting, building, running and emulating Ionic applications.
* **Ionic View** − Very useful platform for uploading, sharing and testing your application on native devices.

**1.3.2.4 ExpressJS**

**Express**

ExpressJS is a web application framework that provides you with a simple API to build websites, web apps and back ends. With ExpressJS, we need not worry about low level protocols, processes, etc.

It provides various features that make web application development fast and easy which otherwise takes more time using only Node.js.

Express.js is based on the Node.js middleware module called connect which in turn uses **http** module. So, any middleware which is based on connect will also work with Express.js.

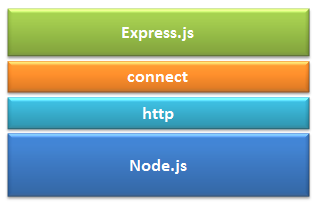
[](http://www.tutorialsteacher.com/Content/images/nodejs/expressjs.png)

Figure 1.2 Express.js

**1.3.3 Database**

**1.3.2.1 MongoDB**

# **MongoDB**

MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling.

MongoDB is a document database. Each database contains collections which in turn contains documents. Each document can be different with varying number of fields. The size and content of each document can be different from each other.

The document structure is more in line with how developers construct their classes and objects in their respective programming languages. Developers will often say that their classes are not rows and columns but have a clear structure with key-value pairs.As seen in the introduction with NoSQL databases, the rows (or documents as called in MongoDB) doesn't need to have a schema defined beforehand. Instead, the fields can be created on the fly.The data model available within MongoDB allows us to represent hierarchical relationships, to store arrays, and other more complex structures more easily.

The MongoDB environments are very scalable. Companies across the world have defined clusters with some of them running 100+ nodes with around millions of documents within the database

**CHAPTER 2**

**REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION**

Software Requirement Specification (SRS)

**2.1 Data Requirement**

Data Requirements is meant to be the data that will be used in our application. Data required in their project in all notices that need to be conveyed to the user. This application also require the username, email and passwords of persons in order to register them and sending notification about updates. So two main requirements are-

* + - * 1. Notice Details
        2. User Details
  1. **Functional Requirement**

In order to make the application functional, we require the following-

* + 1. **Download mobile Application**

A user should be able to download the mobile application through either an application store or similar service on the mobile phone.

* + 1. **User Registration**

Given the user has downloaded, then the upper should be able to register through the mobile app. The user must provide user name, passwords and e-mail address.

**2.2.3 User Login** Given user has registered, then the user should be able to login to the mobile app. The login information will be stored on the phone and in the future the user should be logged in automatically.

**2.2.4 Dashboard**

Given that a user is logged in, then first page that is shown should be the dashboard page.

* + 1. **Search Notice**

The user should be able to search for a notice by is title. For ex, if a user type fee, all the notices having fee in their content get displayed.

* + 1. **Deleting Notices**

The user should have the option to delete the unnecessary notices from his phone and that would get deleted from database too. The student who posted the notice would only be able to edit or delete the notice.

* + 1. **Posting Notices**

The user should be able to post the notices. He should be able to add a picture within notices that picture can be taken either from gallery or by using camera of mobile phones.

* 1. **Performance Requirement**

The requirements in this section provide a detailed specification of the user interaction with the software :

* Prominent search feature
* Usage of the notice information
* Response time should be less
* Fault tolerance should be good

**CHAPTER 3**

**PROJECT DESIGN**

**3.1 Use Case Diagram**

**Use case diagram** are usually referred to as behavior diagrams used to describe a set of actors(use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system(actors).

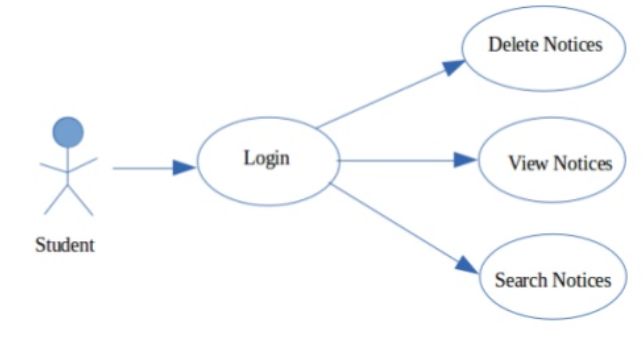


Figure 3.1 Use Case Diagram

**3.2 ER Diagram**

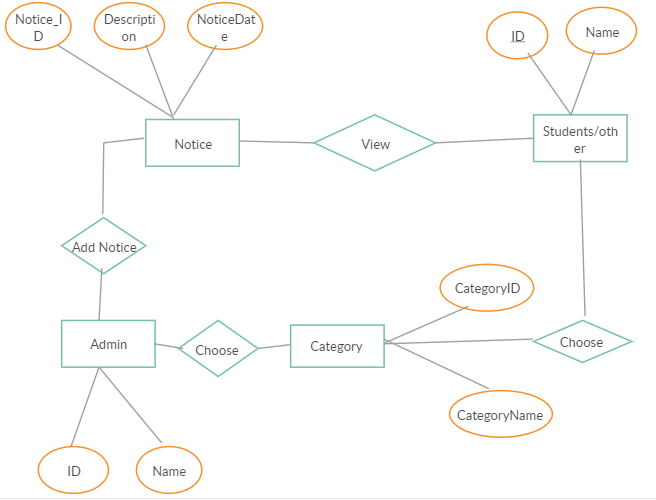
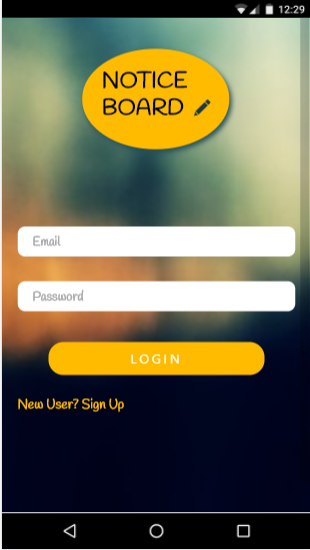
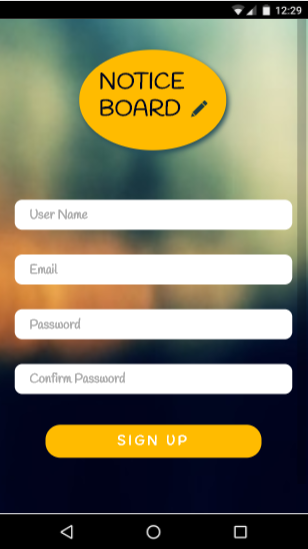
****

Figure 3.2 ER Diagram

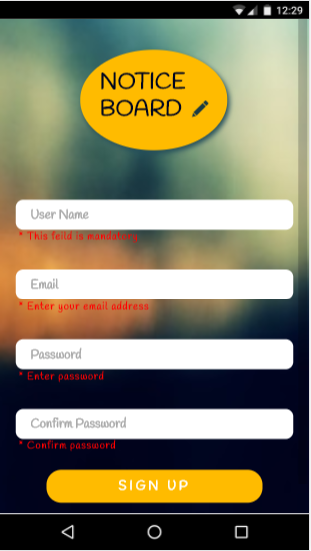
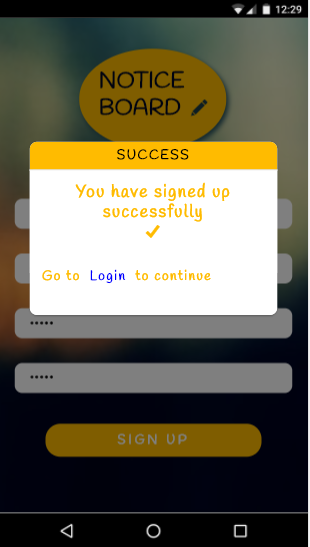
**CHAPTER 4**

**RESULT**

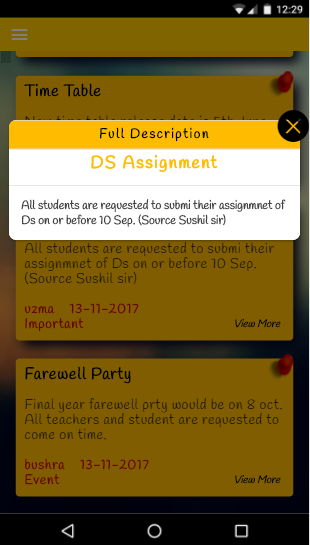
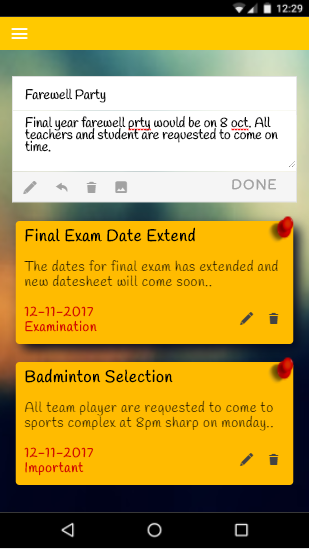
**4.1 Snapshots**

** **

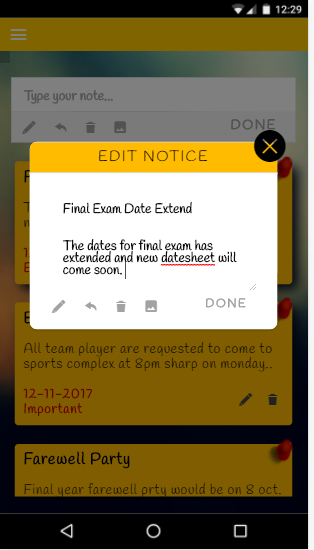
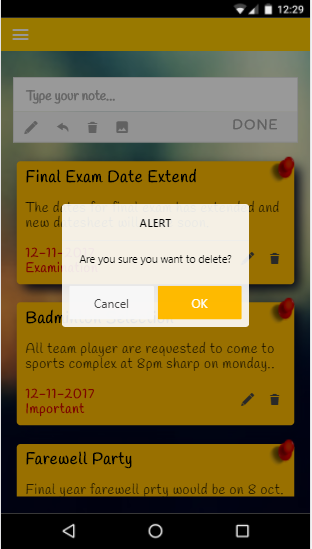
**Figure 4.1 : Login Page Figure 4.2: SignUp Page**

** **

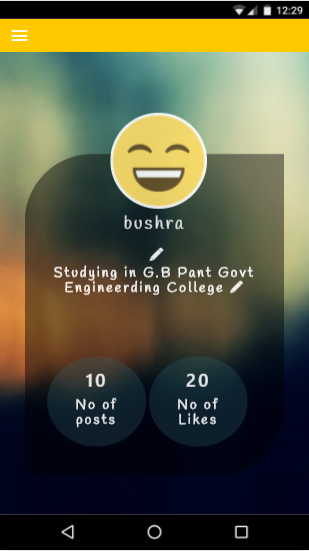
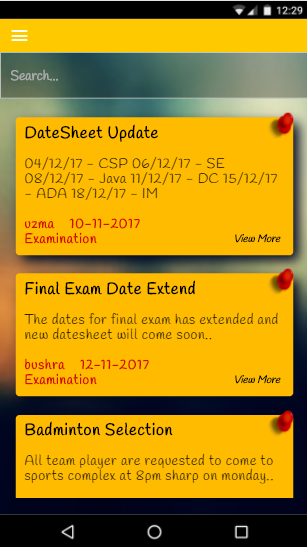
**Figure 4.3 : Authenticaing User Data Figure 4.4: Succesfully SignUp**

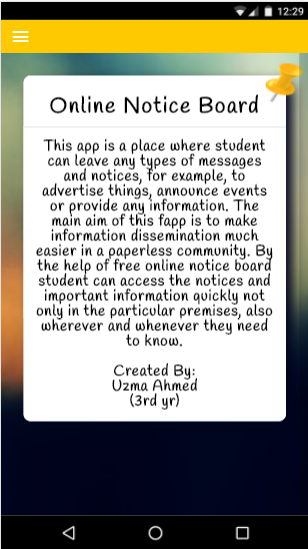
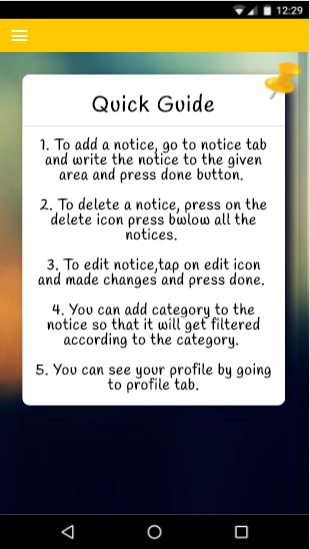
**Figure 4.5 : Showing Full Notice Figure 4.6: Post Notice**

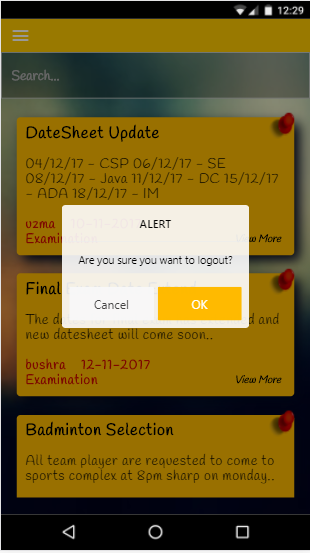
**Figure 4.7 : Edit Notice Figure 4.8: Delete Notice**

**Figure 4.9 : User Profile Figure 4.10: Notice Board**

**Figure 4.11 : About Page Figure 4.12: Help Page**



**Figure 4.13 : Logout User**

**CHAPTER 5**

**REFERENCE**

* [**https://ionicframework.com/docs**](https://ionicframework.com/docs)
* [**https://expressjs.com**](https://expressjs.com)
* [**https://www.mongodb.com**](https://www.mongodb.com)
* [**http://www.tutorialspoint.com/cordova**](http://www.tutorialspoint.com/cordova)