1. **INTRODUCTION**

To develop an android application: Android Application for freelancers and customers. Quick Jobs is a local freelancing application. The app is used mainly aims to provide quick service for a customer in any job need by finding the nearest available freelancer using GPS services. It can also help freelancers find all the local jobs available by category and customers to post them. The app requires Internet, Storage and Location services.

#### AIM

QUICK JOBS aims at providing best user interface and information regarding jobs available nearby or locally. The customer can create an account and view, manage and find the nearest freelancer available to assign a job to get it done.

Customer can also post jobs locally based on category. The freelancer can manage his/her account and get quick jobs or find local jobs available.

#### OBJECTIVE

The principle objective of Quick Jobs Application is,

* + - To offer ease and comfort to people anywhere to get their things done.
    - Allows customers and freelancers to create an account ,view, and manage it.
    - To help people earn money quickly thus providing quick employment.

#### PURPOSE

The purpose of these document is to give a detailed description of requirements for QUICK JOBS. A user of this app can create an account in freelancer or customer mode based on their requirement and can view, manage account and post or receive jobs. The software is mainly handled by both freelancers and customers based on their respective requirements.

#### SCOPE

Today we are going through an era where everything being managed using mobile devices. People are always in need of job or to get some job done at ease. So our app can easily help achieve peoples need and also provide better service for people even at need of emergencies. It can be accessed from anywhere at any time.

1. **REQUIREMENT ANALYSIS**

## SOFTWARE REQUIREMENT SPECIFICATION

The following subsection of the SRS document provides an overview of the entire SRS

#### PURPOSE

The purpose of the project is to help people to get their work done quickly or to get paid for doing quick work. Account registration is required to manage this app. User can login from anywhere at anytime. A customer can easily give jobs to be done quickly and freelancer can accept or decline based on their needs.

#### SCOPE

QUICK JOBS provide facilities for the customers to get any work as quickly needed to be assigned to a nearest freelancer available. User can create an account in the app and manage it. The user can login with the username and password. The app provides users to register and login using OTP Verification. Jobs once finished can be payed by the customer using Cash on Delivery or Google Pay . Customers and freelancers can also discuss about the job using a chatting screen and fix a cost to get the work done.

#### BENEFITS

QUICK JOBS reduces work, maintains accuracy , increases efficiency and save time. It helps people to get money or get quick works done especially in case of emergencies.

#### DEFINITIONS AND ABBRIVIATIONS

Freelancers

Freelancers can accept a job coming on their screen based on their needs. They can

chat with customers and decide whether to proceed with task.

Customers

Customers can post a job as per their need to be done quickly and can decline work

with freelancer before a freelancer accepts based on their chat and discussions.

#### REFERENCE

* + - 1. <https://developer.android.com/>
      2. ANDROID APP DEVELOPMENT FOR DUMMIES – DON FELKER
      3. WIKEPEDIA [www.wikipedia.com](http://www.wikipedia.com/)
      4. FIREBASE <https://firebase.google.com/>
    1. OVERVIEW

The rest of the SRS describes the various system requirement features and functionalities in details.

## OVERALL DESCRIPTION

This project aims to provide a means to help customers get quick work done by a

freelancer according to their need. In return freelancers can also make money in part

time.

The user for the system are:

Freelancers

Freelancers can accept a job coming on their screen based on their needs. They can

chat with customers and decide whether to proceed with task.

Customers

Customers can post a job as per their need to be done quickly and can decline work with

freelancer before a freelancer accepts based on their chat and discussions.

#### PRODUCT PERSPECTIVE

This is considered as new application that makes people work easy and help people earn quickly.

#### PRODUCT FUNCTIONS

It include-

Freelancers: Users who work and earn money.

Customers: The on who post jobs to find their work done quickly and pay freelancers.

#### USER CHARACTERISTICS

There are two types of users that interact with the system: freelancers and customers.

##### CONSDOCTORTS

**Regularity policies**: There is no regularity policies

**Interfaces to other applications**: There shall be no interfaces

**Higher order language functions**: Java and XML shall be used for developing the forms in the user interface with the help of Android Studio and for database information, Firebase shall be used.

**Safety and security considerations**: The passwords used are the security issues here.

**Hardware limitations:** App may not be at optimum conditions working if it runs on low memory devise and devices that doesn’t support high accuracy GPS service.

##### ASSUMPTIONS AND DEPENDECIES

* + - * The user must know how to use a mobile efficiently.
      * User can access both customer and freelancer mode at any time.

##### FUNCTIONAL REQUIREMENTS

**1.Content**: There are 3 categories.

##### Module Description

##### 1. Splash Screen

##### 2. Three Onboarding Screens

##### 3. User Selection Screen(Freelancer/Customer)

##### 4. Freelancer Login/Registration Screen

##### 5. Customer Login/Registration Screen

##### 6. OTP Registration/Login Screen

##### 7. Profile Registration Screen

##### 8. Freelancer and Customer Map Screen

##### 9. Job Post dialogue Box for Customer

##### 10. Freelancer and Customer Chat Screen

##### 11. Get route and directions for freelancers

##### 12. Verification on Job Complete for Customers

##### 13. Payment Gateway – Google Pay (Using UPI)

##### 15. History of Jobs

##### 16. Detailed History Screen

##### 17. Post Local Jobs for Customers based on Category.

##### 18. History of Local jobs for Customers

##### 19. Local Jobs availability for Freelancers

##### 20. Call Customers in Local Jobs for Freelancers

##### 21. Edit Profile Screen - Settings

##### 22. Invite to App Screen

##### Users of the system:

* 1. Freelancers

• Freelancers can create an account and login with email and password or with OTP.

• Freelancers can receive job request and have option to accept or decline it.

• Freelancers can chat with customer regarding job and start job.

• Freelancers can finish the job for verification.

• Freelancers can get paid for job using cash or Google pay

• Freelancers can access local jobs based on category and call customers.

• Freelancers can view history of their jobs

* 1. Customers

• Customers can create an account and login with email and password or with OTP.

• Customers can write a job and post them.

• Customers can chat with freelancer regarding job.

• Customers can verify once job completed.

• Customers can pay freelancers by cash or Google pay.

• Customers can post local jobs category wise.

• Customers can view history and delete local jobs posted.

• Customers can view history of their jobs.

##### NON-FUNCTIONAL REQUIREMENTS

They are the quality requirements that stipulate how well a software does what it has to do.

##### PERFORMANCE REQUIREMENTS

It is available whenever in need.

**•Performance**: The firebase server used should provide good performance and ability to manage performance with the increased number of users.

**•Reliability**: It means the extent to which the program performs with required

precision. The software created should be reliable that is it should perform its

required functions under stated conditions for a specified period of time.

**•Usability**: It should be user friendly and should require least effort to operate.

**•Portability**: The application is made using Java, XML etc. which are platform independent and can be transported to other servers with minimum effort.

**•Flexibility**: It is the effort required to modify operational program. The whole application should be made using independent modules so that any changes done in one module should not effect the other one and new modules can be added easily to increase functionality.

**•Availability:** Application will run 24\*7 if internet connection is available.

**•Security:** Security requirements places restrictions on the freelancers and customers by permitting access only by using password. The software developed is free from all security issues and the software details are secure.

**•Maintainability:** Maintenance is one form of change that typically is done after the software development has been completed. Application needs timely updation of data by the developer.

Software System Attributes:

**•Execution speed**: The speed with which a computational device can execute instructions and it is considerably good.

**•External requirements**: Requirements which arise from factors external to the system.

##### EXTERNAL INTERFACE REQUIREMENTS

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface

##### User Interfaces

First time when the user this system he will directed to Splash Screen. If the user is running the app for the first time then he/she will be directed to the onboarding page which carries instructions to the app. Then the user has an option to select freelancer or customer mode. If the user has not registered, he/she should be able to register login/registration page or with an OTP using phone number. Then the user is directed to their respective map activities. The user also have profile editing interface. The user can also access local jobs screen in their respective mode. The app also has an invite screen when users can invite their friends to download the application.

##### Hardware Interfaces

Ø mobile device or

Ø mobile simulators

##### Software Interfaces

XML shall be used for developing the website user interfaces with the help of Java and for data base information Firebase can be used.

##### Communication Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems and database in the service system.

##### CONSTRAINTS

* The user have to use password or user OTP to access their accounts.

##### CONCLUSION

The application executed successfully with access to all information the events at any time with no restrictions other than internet and gps.

1. **SOFTWARE FUNCTIONAL SPECIFICATION**

##### INTRODUCTION

Software functional requirements actually describe a set of high level requirements, where each high level requirements take some data input from the admin and provides some data to the user as output. Also each high level requirement might consist of some other functions.

Given below are the high level requirements of Event Management System .

##### HIGH LEVEL REQUIREMENTS

R1: Login

Description: Login function first determines whether the email entered by the user is a valid or

not. It checks the password and also whether the email and password are correctly or not.

Login is also possible using Phone number by OTP Verification.

R1.1:Enter email Input: Email

Output : Freelancer\Customer prompted to enter the password after the email is valid.

R1.2: Enter Password

Input: Password

Output : User permitted to login after the validation successful.

R1.3: Enter Phone Number and verify OTP

Input: Phone number and OTP

Output : User permitted to login after correct OTP.

R2:committee

Details: the details of the committee which the user belongs should be provided R2.1:Choose committee

Input: committee

Output: Displays the details of that particular committee , events, contests and workshops.

1. ER DIAGRAM
2. **SYSTEM DESIGN**

The object relationship pair can be graphically represented by a diagram called Entity Relationship Diagram. It is mainly used in database applications but now it is more commonly used in data design.

The primary purpose of ERD is to represent the relationship between data objects. Various components of ERD are

* + Entity
  + Relationship
  + Attribute.

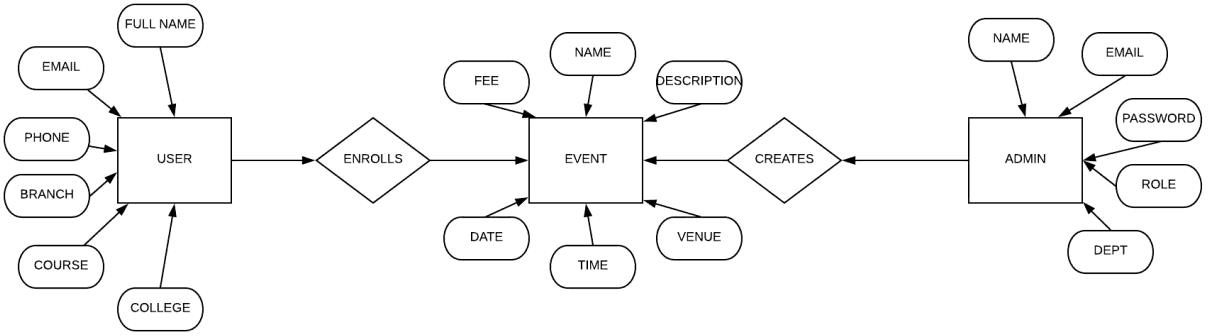
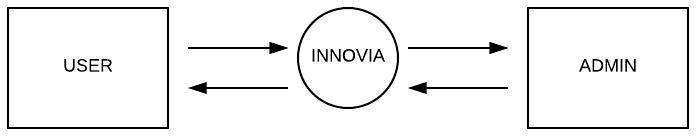


fig: E-R diagram of innovia2K18

#### . DATA FLOW DIAGRAM

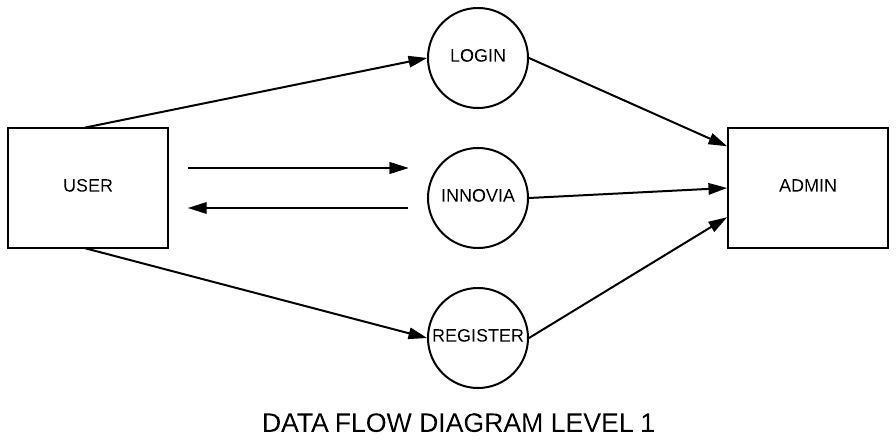
DFD LEVEL 0

##### CONTEXT DIAGRAM



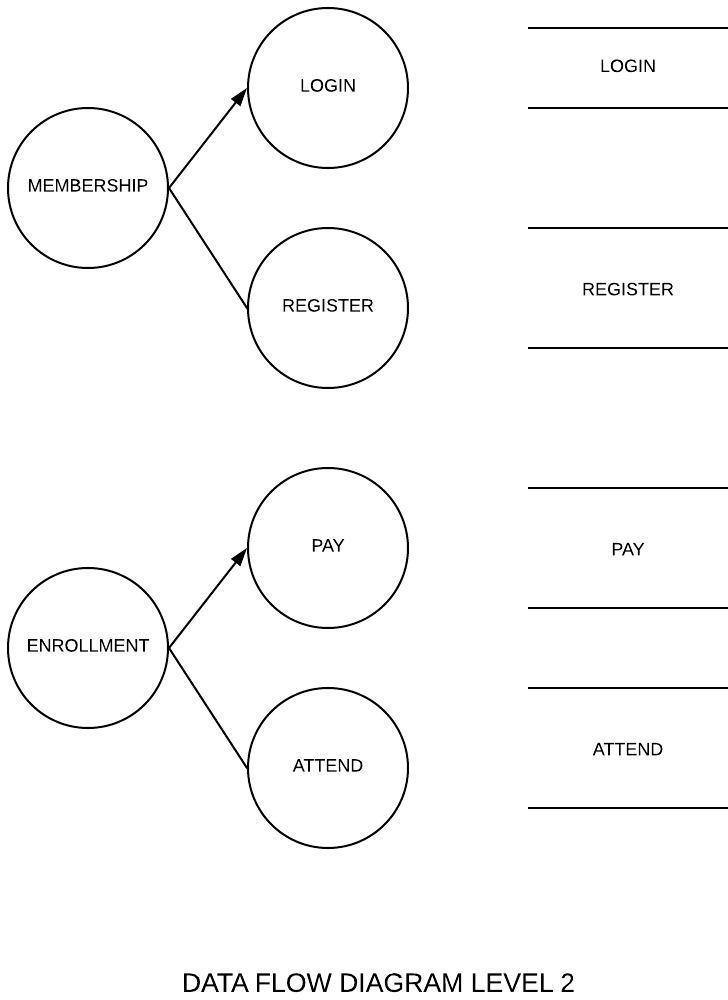
**Figure data flow diagram-level 0**

**DFD LEVEL 1**



**Figure data flow diagram-level 1**

DFD LEVEL 2



**Figure data flow diagram-level 2**

### INPUT AND OUTPUT DESIGN

##### Introduction

Design of a system can be defined as the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Thus system design is a solution, a ”how to” approach to the creation of a new system. This important phase provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Thedesign step produces a data design, an architectural design and a procedural design. The data design transforms theinformation domain model created during analysis into the data structures that will be required to implement the software. The architectural design defines the relationships among major structural components into a procedural description of the software. Source code is generated and testing is conducted to integrate and validate the software.

##### Input Design

Input design is the process of converting the user originated inputs to a computer based format. The goal of designing input data is to make the automation is easy and free from errors. Input design is a part of overall system design, which requires very careful attention. If data going into the system is incorrect, then the processing and output will magnify these errors. Thus the designer has a number of clear objectives in the different stages of input design:

* + To produce a cost effective method of input.
  + To achieve the highest possible level of accuracy.
  + To ensure that input is acceptable to and understood by the user

##### Output Design

At the beginning of output design various types of outputs (external, internal, operational, interactive and turnaround) are defined. Then the format, content, location, frequency, volume and sequence of the output are specified. The content of the output must be defined in detail. The system Analyst has two specific objectives at this stage .

To interpret and communicate the results of the computer part of a system to users in a form that they can understand and which meets their requirements. To communicate the output design specifications to programmers in a way, which is unambiguous, compressive and capable of being translated into a programming language.

### TABLE DESIGN

The database design is a logical development in the methods used by the computers to access and manipulate data stored in the various parts of the computer systems. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as an Organizational resource and as an integrated whole. The main objectives of database are data integration, data integrity and data independence.

Database management systems (DBMS) allow the data to be protected and organized separately from other resources like hardware, software and programs. DBMS is a software package which contains components that are not found in other data management packages

.The significance of DBMS is the separation of data as seen by the programs and data as stored on the direct access storage devices. ie; the difference between the logical and physical data.

Data design is the first design of the three design activities that are conducted during software engineering. The impact of data structure on program structure and procedural complexity causes data design to have a profound influence on software quality .The concepts of shared memory and semaphores provide the foundation for an approach to data design.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. The database serves as the repository of data, so a well designed database can lead to a better program structure and reduce procedural complexity. In a database environment, common data are available and used by several users.

Data design consists of the design of the database and various data structures used. The design of data begins during the creation of the analysis model. The movement of the data between the process environment and the database is completely invisible to a programmer and user.

Objective in database design is:

Controlled Redundancy: A unique aspect of database is storing data only once, which controls redundancy and improves system performance.

Ease of learning and use: Database should be modified without interfacing with established ways of using the data.

Data Independence: Adding new data without rewriting the application programs. The database should be ”tunable” to improve performance without rewriting programs.

More information at low cost: Using storing and modifying data at low cost are important

Accuracy and integrity: The accuracy of the database ensures the data of good quality and the content remains constant. Integrity controls detect data inaccuracies where they occur.

Recovery from failure: With multi-user access to a database, the system must recover quickly after it is down with no loss of transaction.

Privacy and security and security: Database should be prevented from unauthorized access. Users must be positively identified and their actions monitored.

* Key

A key is a column used to identify rows, it is not the same as an index.

Various types of keys are:

Primary Key: A primary key is a unique key with the extra constraint that all key columns must be defined as NOT NULL.

Candidate Key: A candidate key is the combination to one or more columns, the values of which uniquely identify each row of a table.

Foreign Key: One or more columns whose values are based on the primary key or the candidate key values from another.

Unique key: One or more column that must be unique for each of the table. Unique key is allowed to be null.

The following tables used in this project

Table name: Login

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Type | Collation | Attributes | Null | Defaults | Comments |
| Id | Int |  |  | No | None |  |

# SYSTEM IMPLEMENTATION

Implementation is the stage of project when the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on existing practices shifts to the user department. If the implementation stage is not carefully planned and controlled, it can cause chaos. The implementation stage is a systems project in its own right. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, training of staff in the changeover procedure and evaluation of changeover methods

##### Implementation Planning

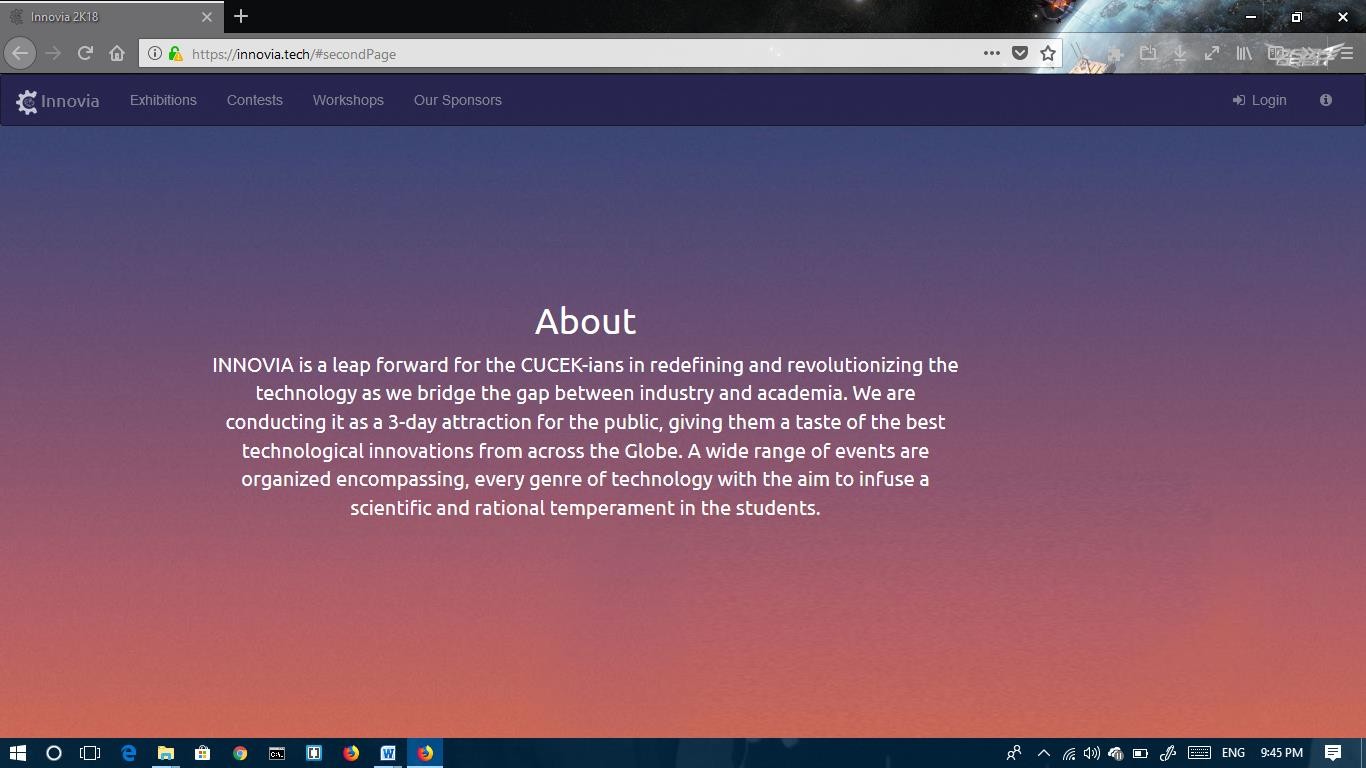
The implementation of a system involves people from different departments and system analysts are confronted with the practical problem of controlling the activities of people outside their own data processing department. Prior to this point in the project, system analyst has interviewed department staff with the permission of their respective managers. The implementation coordination committee should be responsible for a successful implementation. The composition of committee is important. There should be at least one representative of each department affected by the changes and other members should be co- opted for discussions of specific topics.

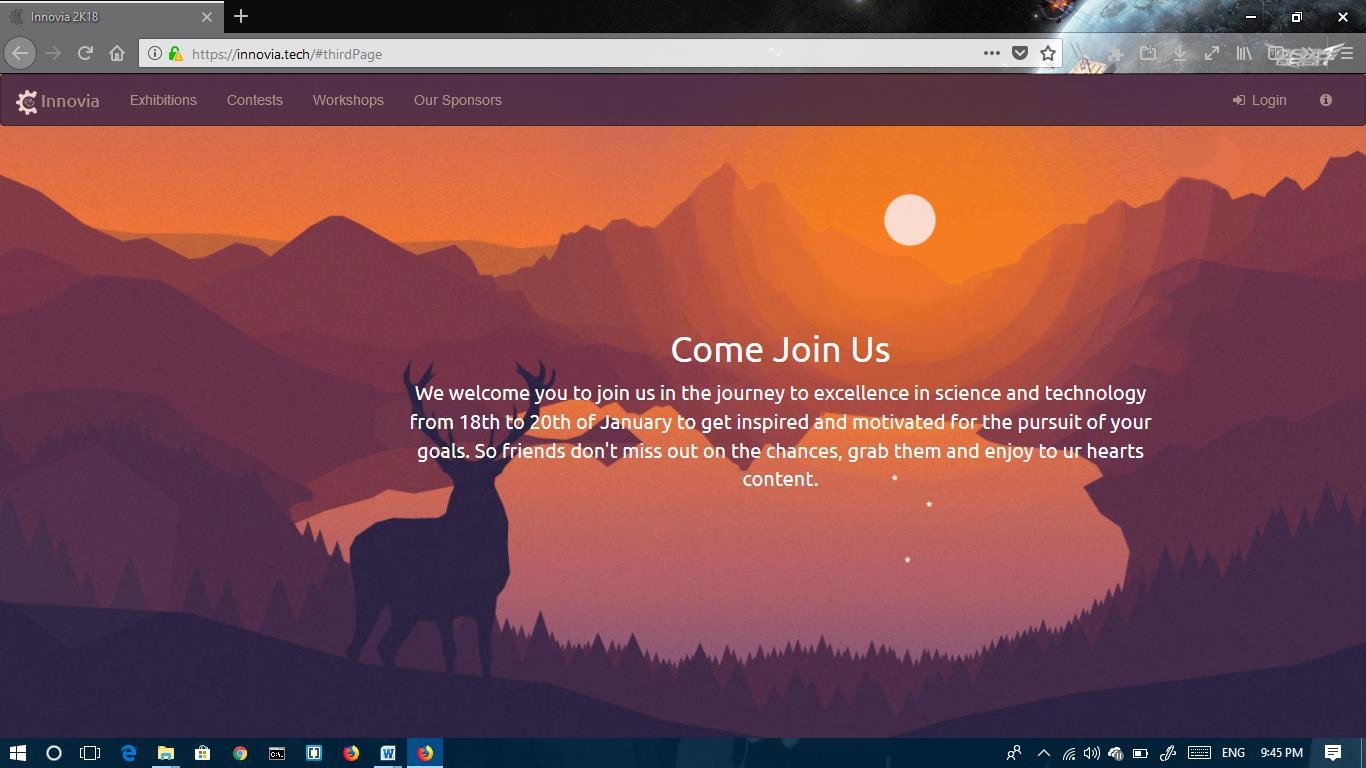
**Forms**

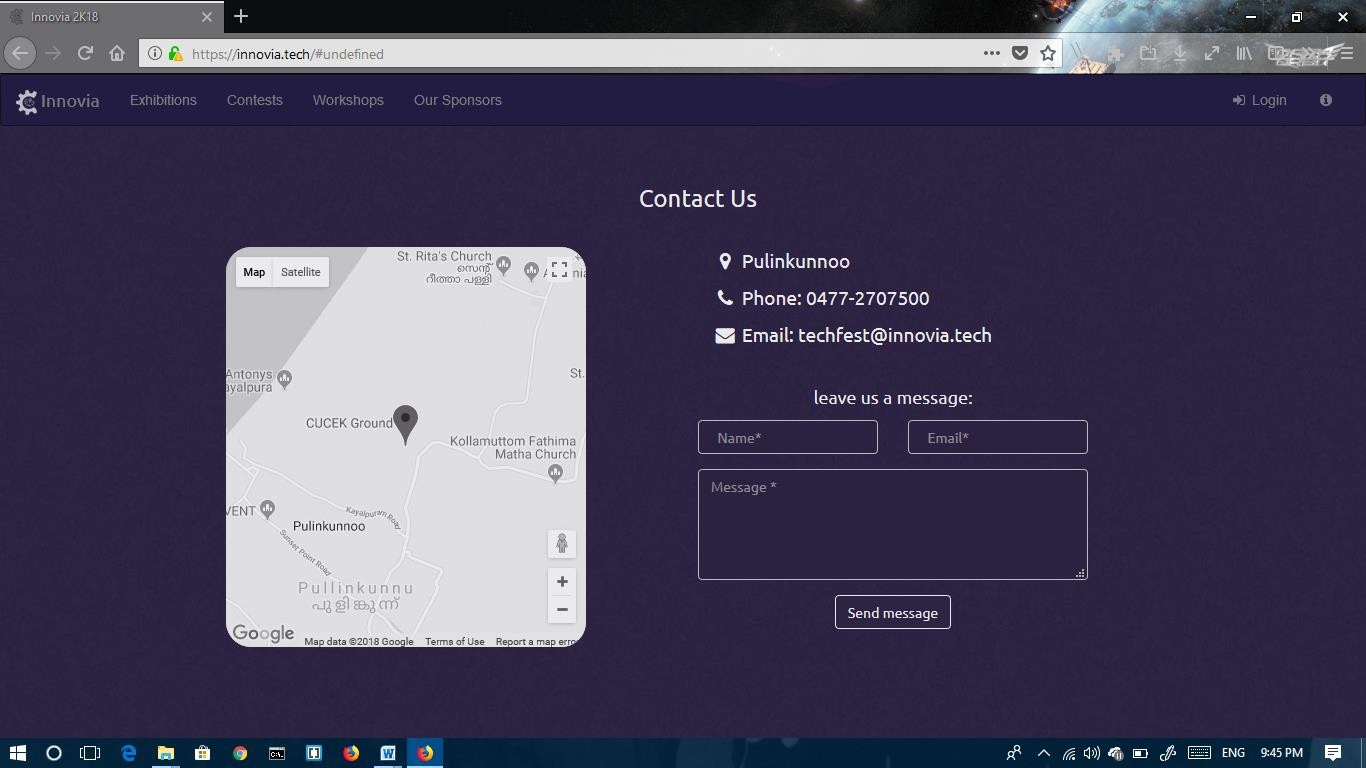
**Home page**



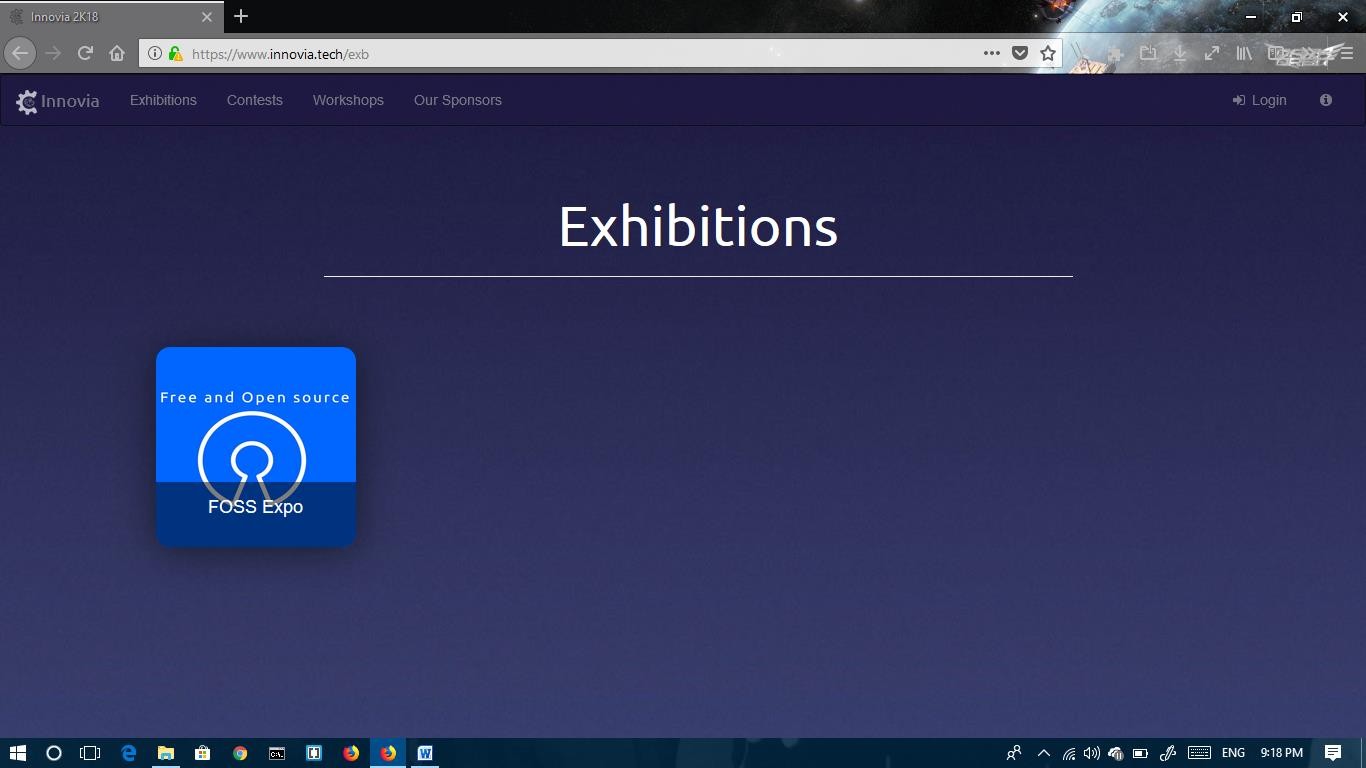
This is the homepage which will appear when we first enter the site



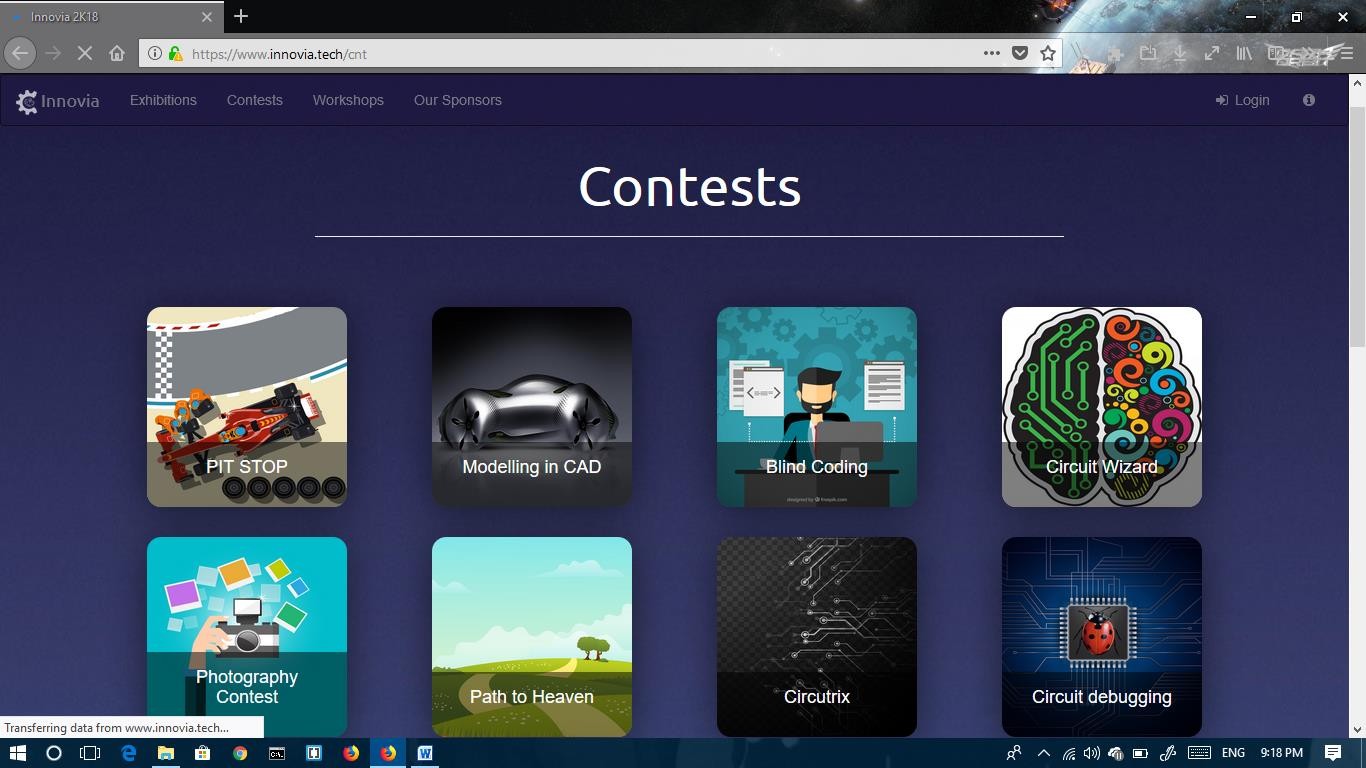




Exhibition



The is the page which appear when we click Exhibition Contests



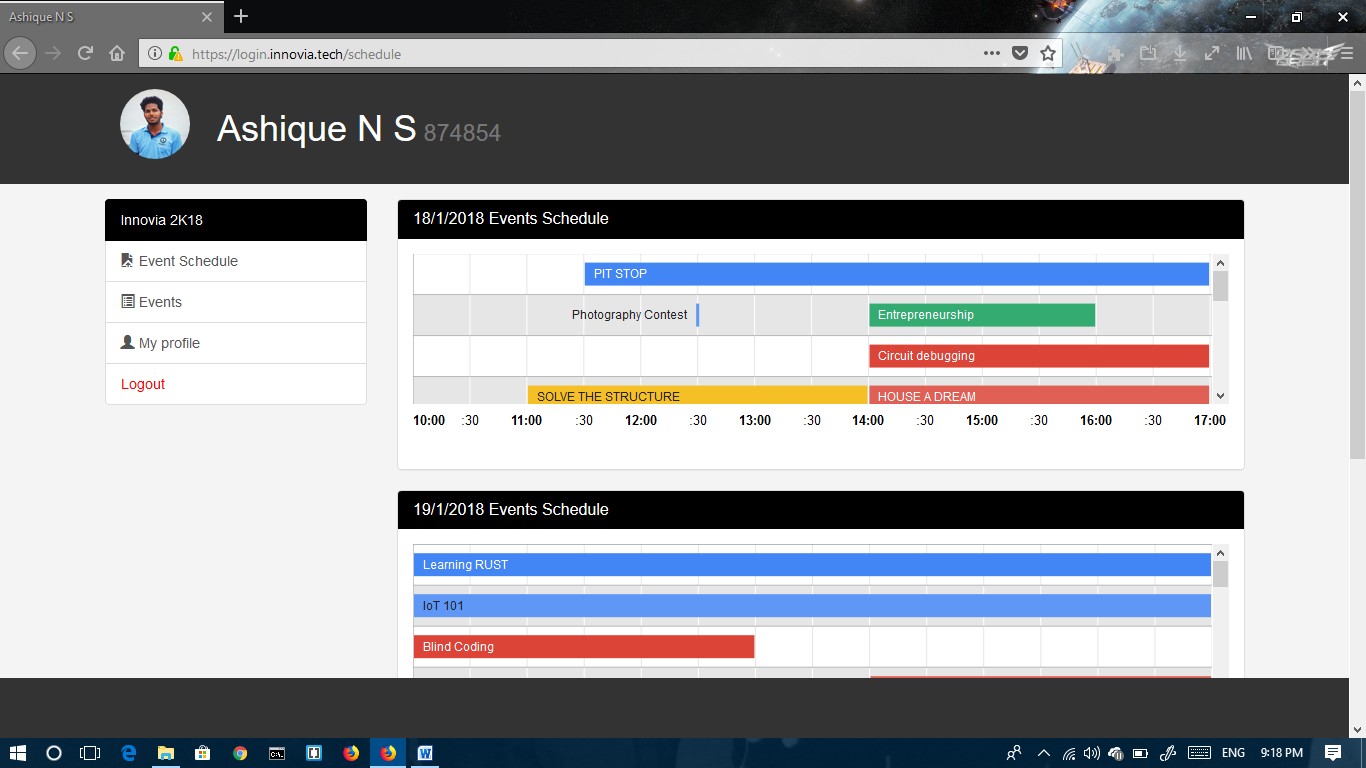
This page will appear when we click on the Contest

WorkShops



This is the workshop page

Logged in page

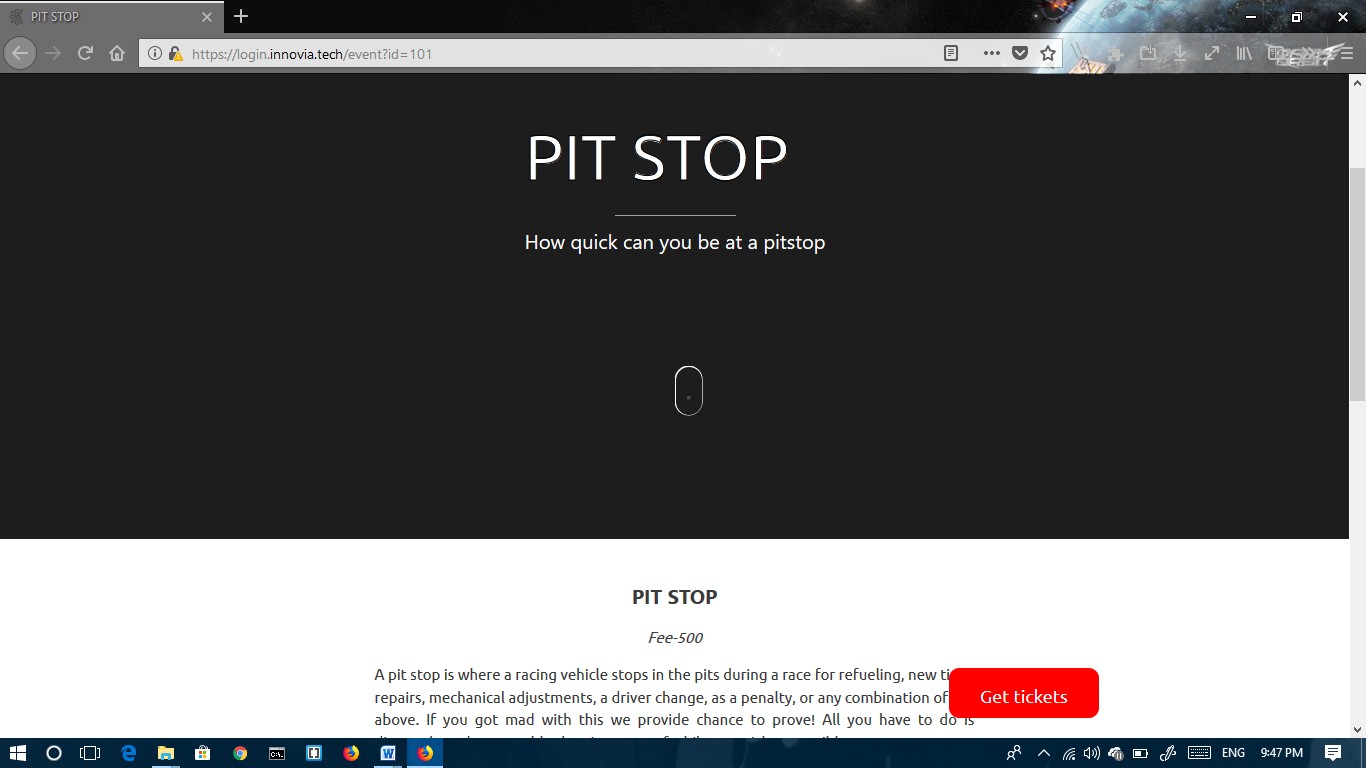


This is the page which appear when we log into the site. Where we can manage the events and all



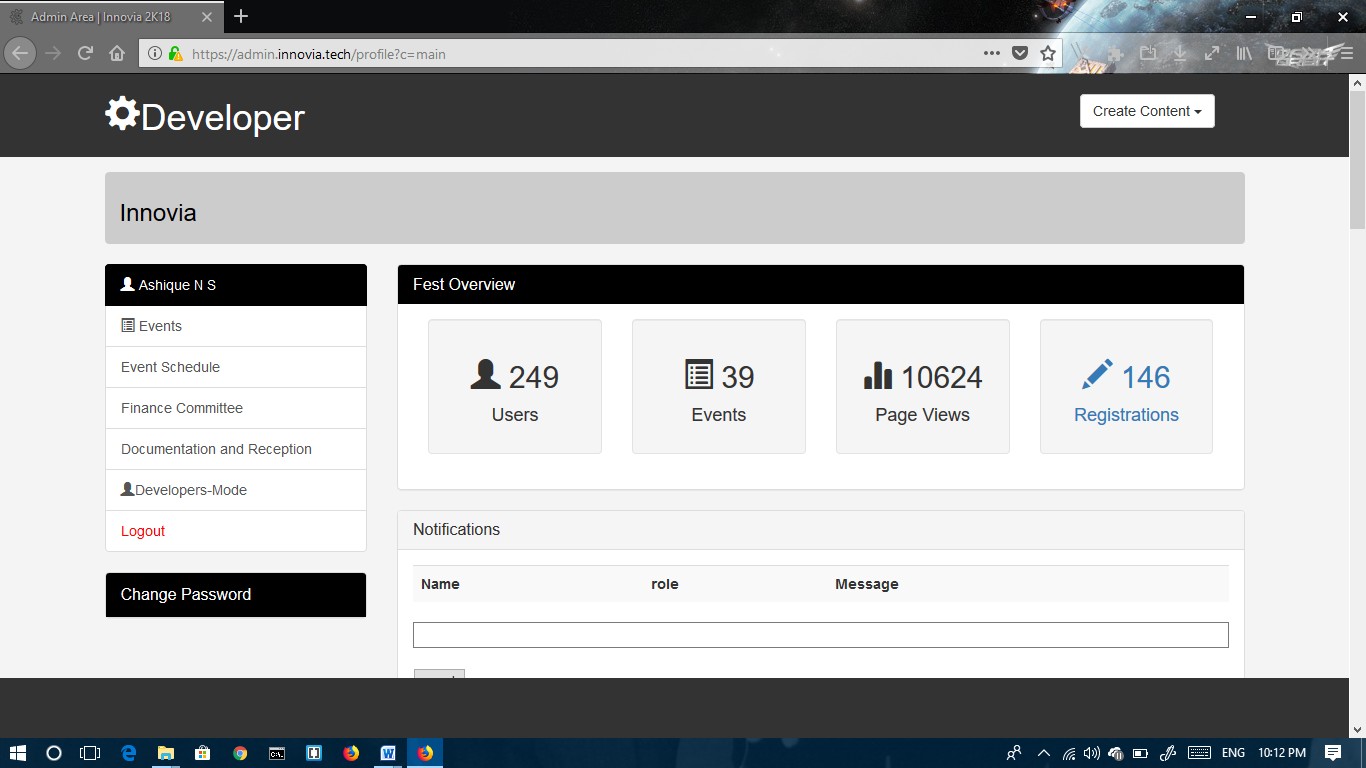
Event chart is well displayed

Event Registration

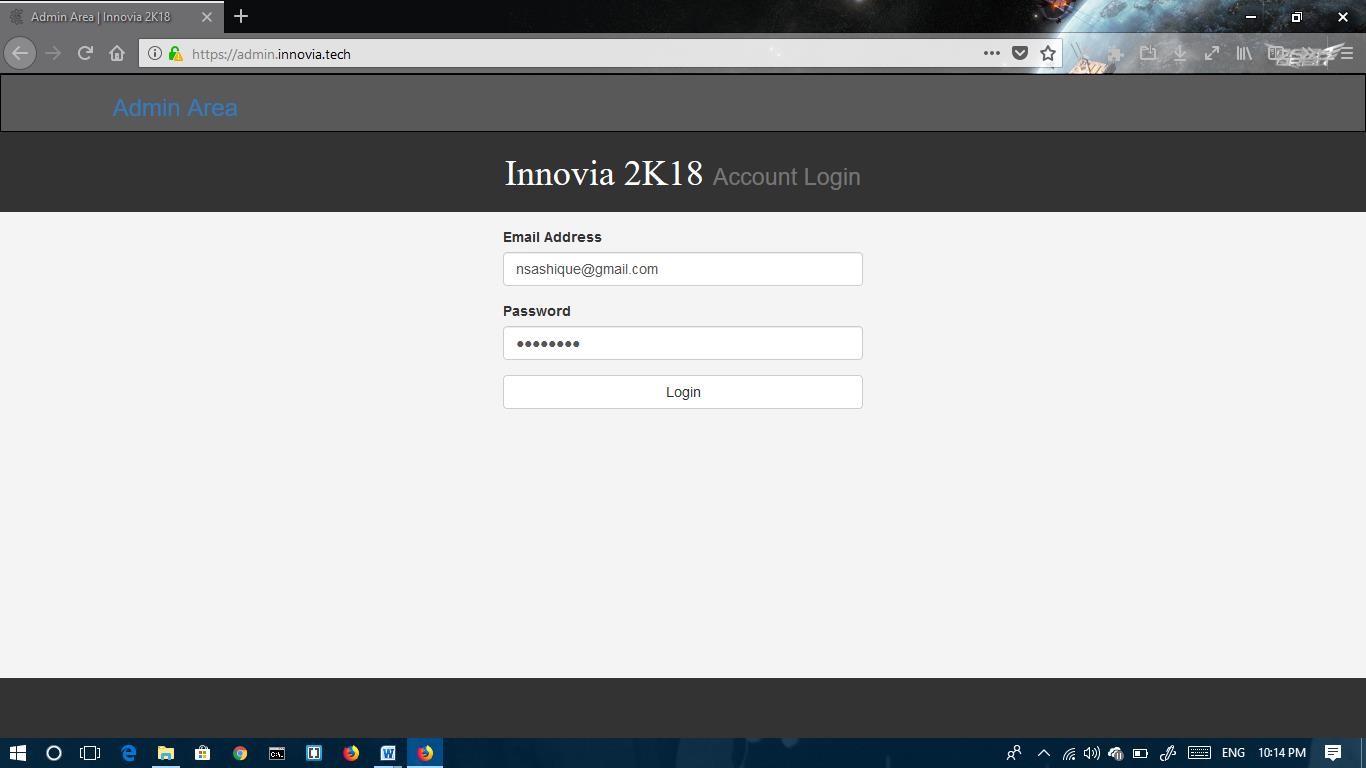


For registering the events a „GET TICKET‟ button is displayed

Developer page



The developer can manage the site through this page.



Developer login page

# TESTING

Software testing is a critical phase of software quality assurance. It indicates the ultimate review of specification ,design and code generation. Once the source code has been generated

,software must be tested to uncover and correct maximum possible errors before being delivered. Testing emphasizes on a set of methods for the creation test cases that fulfill overall testing objectives.

The primary objectives of software testing are as follows:

* + Testing is a process of executing program to find an error in it.
  + A good test case should have high probability of finding an as-yet undiscovered error.
  + A test case will be considered successful if it uncovers an as-yet undiscovered error

Test cases

* + Description : This test case would check whether the details of the admin is taken correctly into the database.

Steps : Enter Username , Password and other necessary details shown in the signup form

Expected output : Check all the required fields are Filled then display the message

* + Description: This test case would check whether all the fields in the form are filled

Step : Keep the some field blank

Expected output : Display “Enter the fields ” .

* + Description : To check user name and password are correct.

Steps :Enter Wrong user name and password for login.

Expected output : Display Error Message .

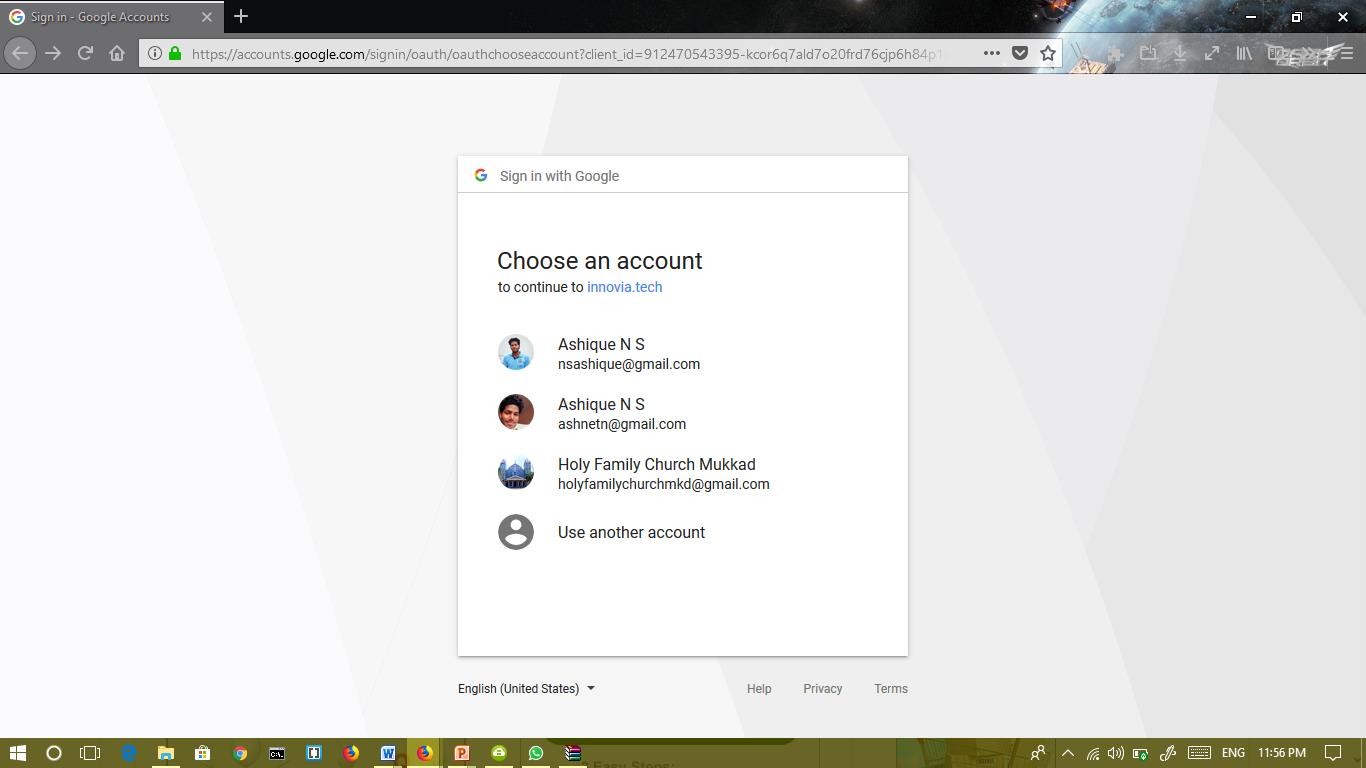
Description: This test case would check for a valid username and password in the login form.

Steps : open login form, enter username and password then press the Login button.

Expected output: the corresponding user should be allowed to login and he is directed to the home page.

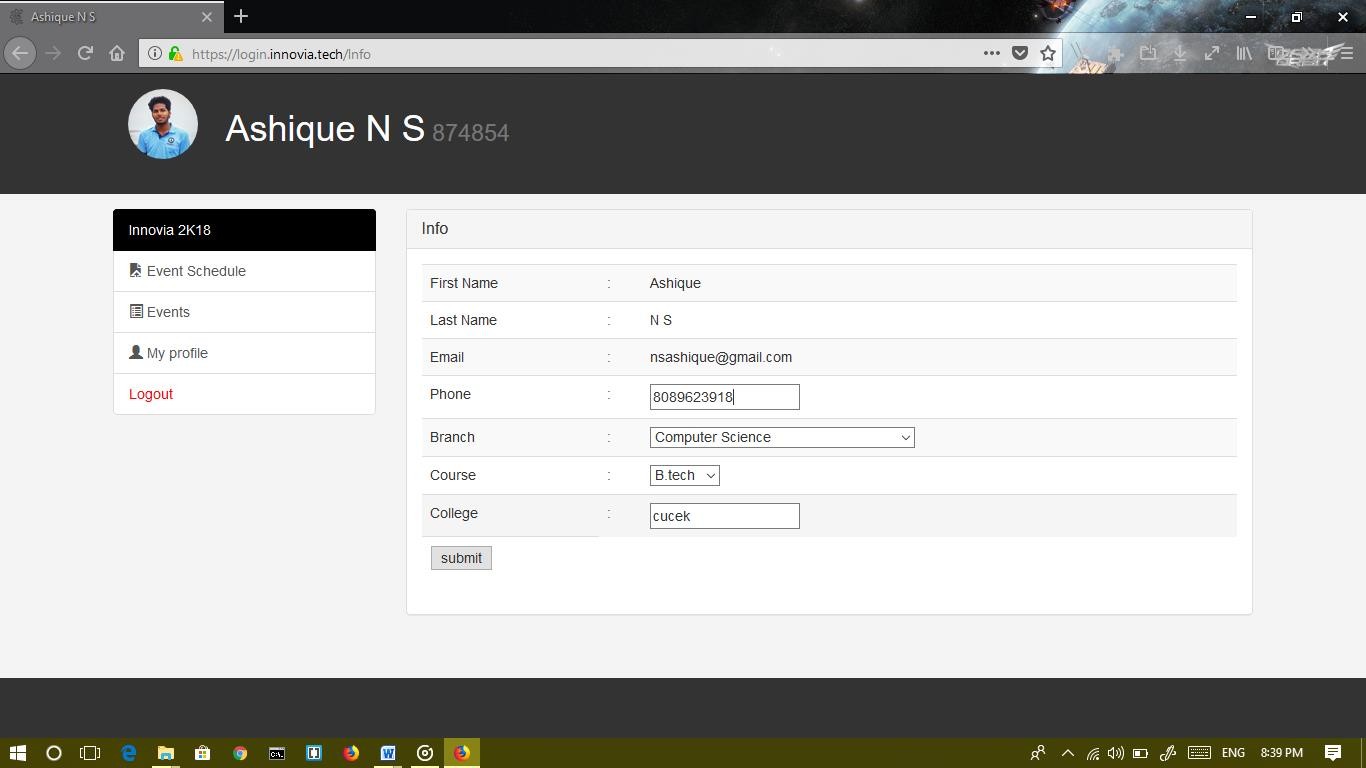
USER LOGIN

Innovia 2K18 account login using Login.



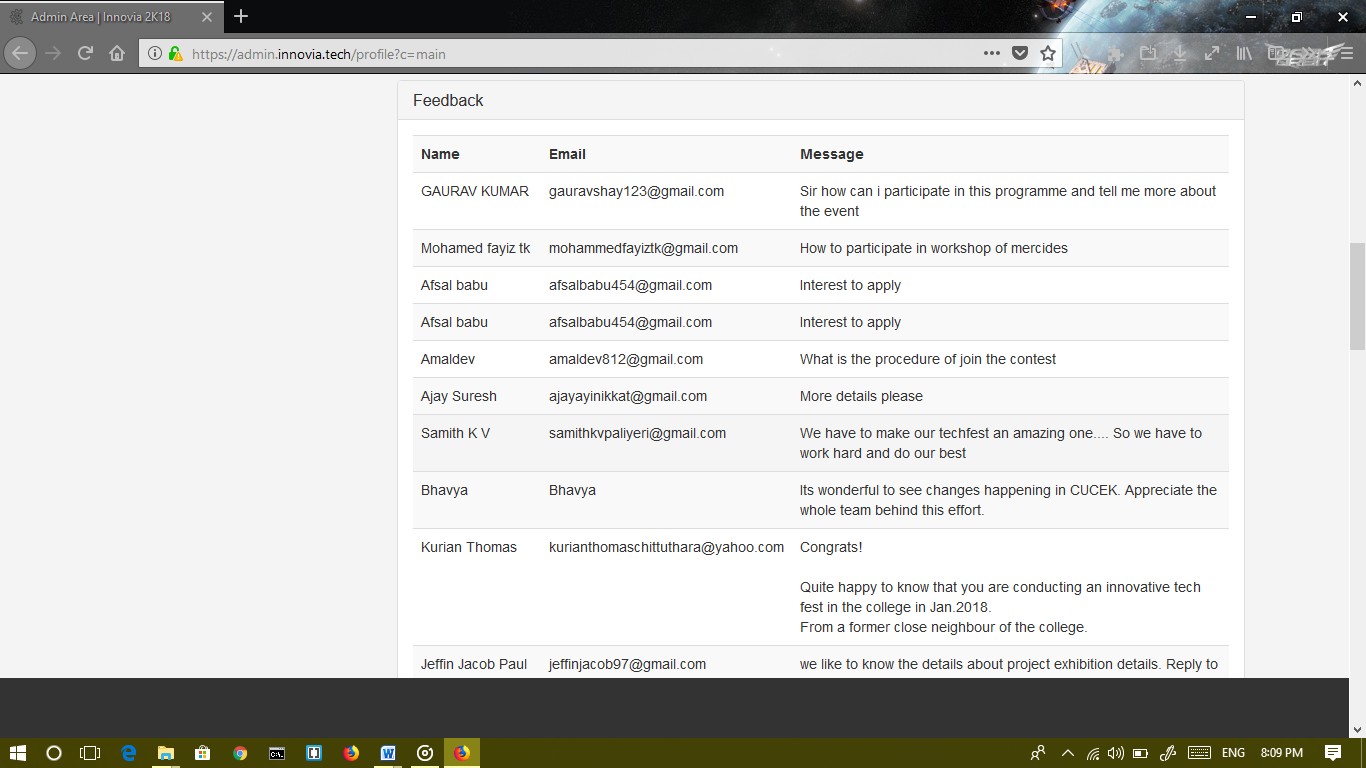
Logged in page

Can edit the user info after signing with google account to innovia 2K18

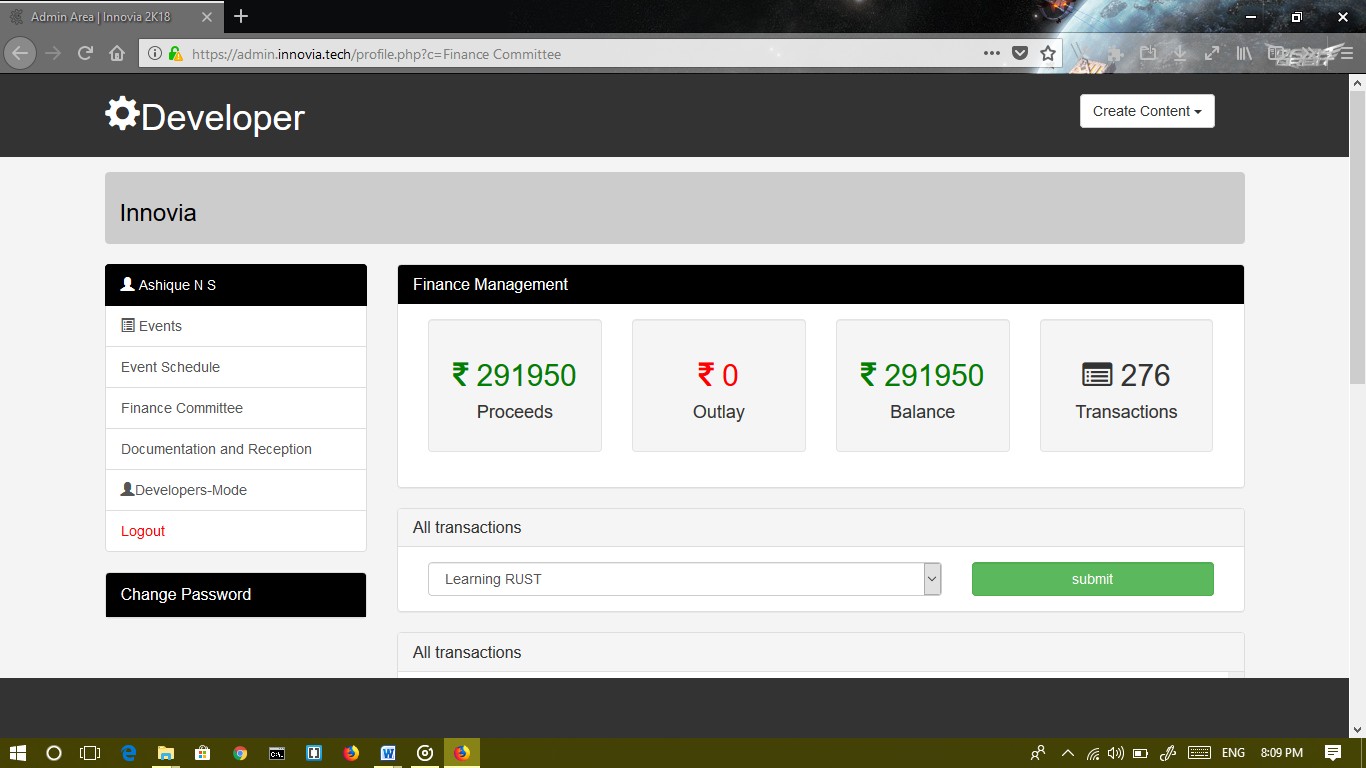


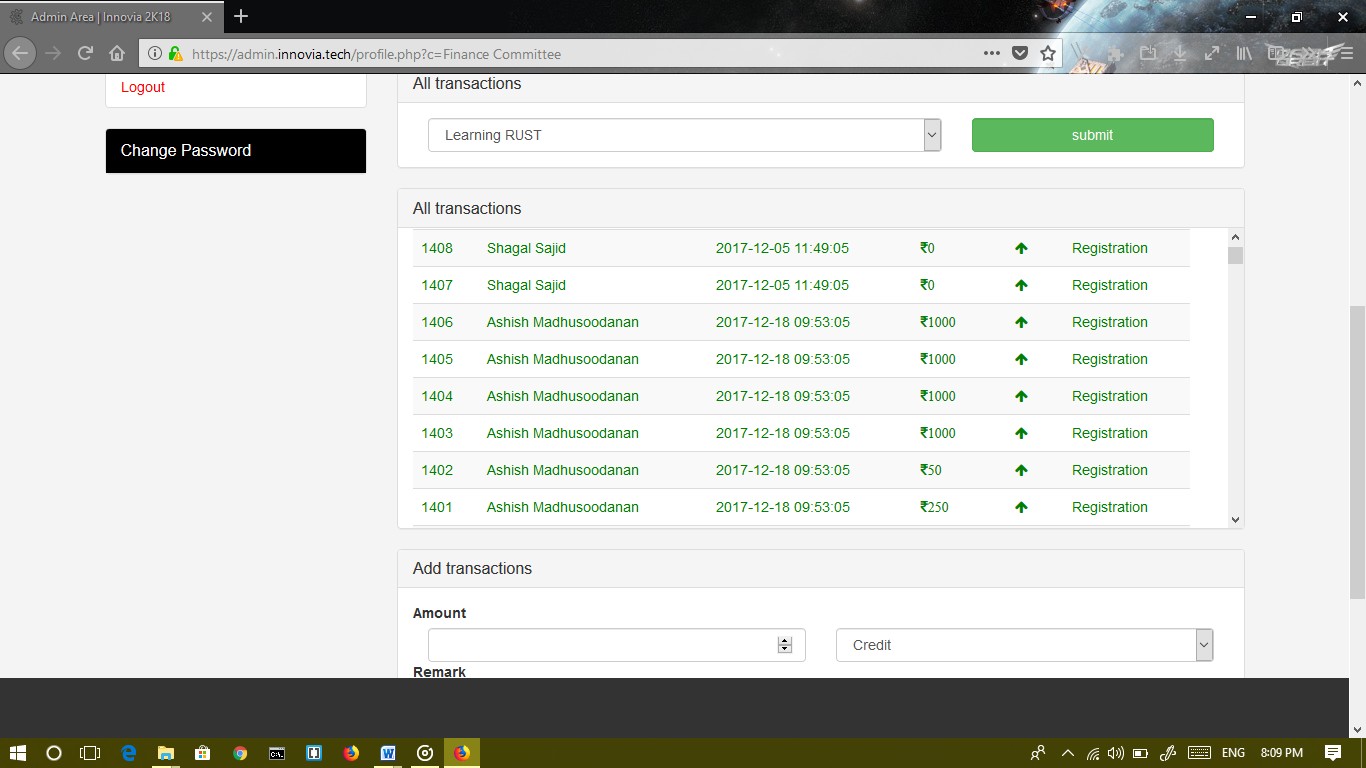
1. RESULT

List of students enrolled for events

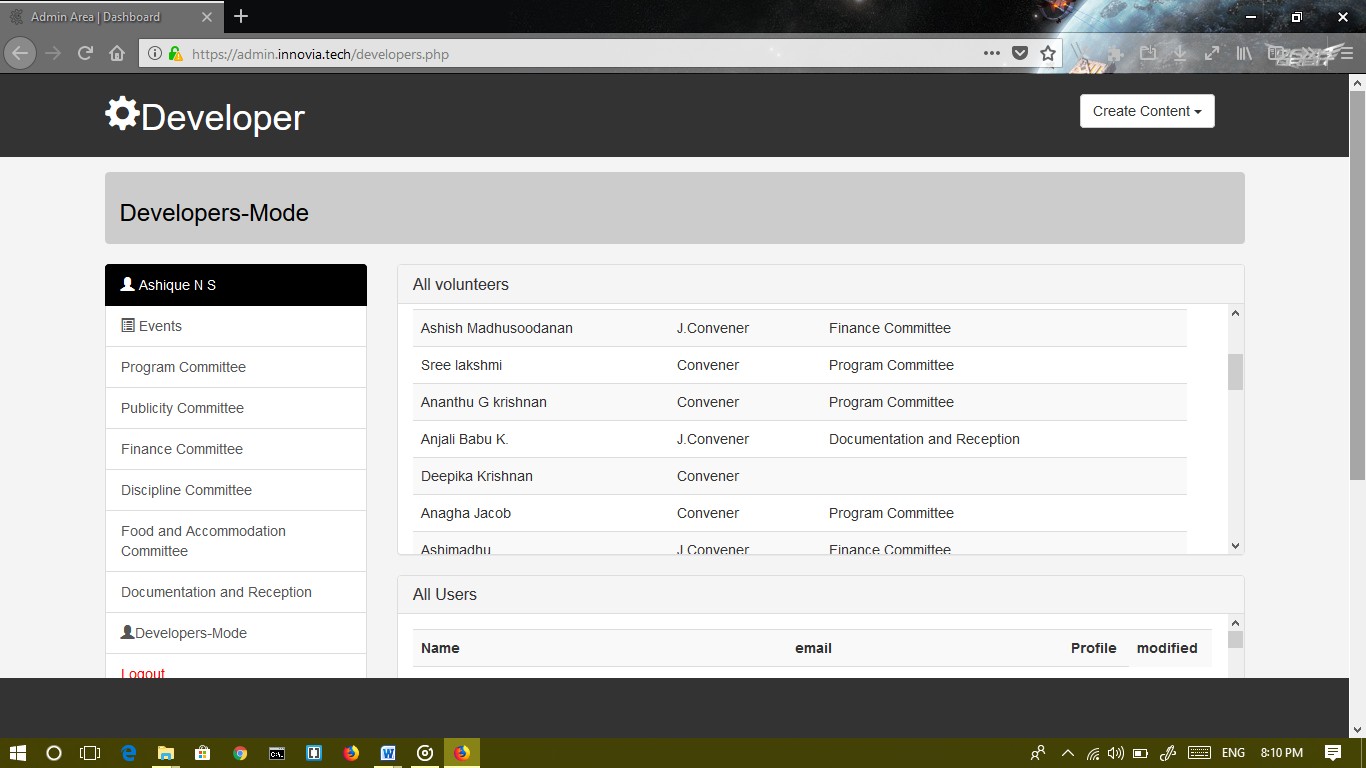


Finace management and invoice





List of Volunteers of innovia 2K18



## FUTURE SCOPE

##### Expanding the databases.

**System to add photos and videos of the previous event.**

**Website integrated payment system more updated user interface**

**Creating mobile application for easy access.**

1. **CONCLUSION**

The main system objectives specified in the requirement documents are believed to have been satisfactory met. The system gives good result in the testing. The processing of the proposed system is simple and is in regular order.

The various interfaces employed in the project are user friendly and easy to understand. The system has been designed in manner so as to provide valuable, timely and accurate information. The system is effective and efficient. Any native user can understand the working of the system and can efficiently use the system.

The system also has extensibility feature that allows the users to fulfill their additional requirement in the current system. Therefore updating of the system will make the system live.

# BIBLIOGRAPHY

1. Programming the world wide web,4th edition-Robert w.Sebesta
2. Fundamental of Software engineering,4th edition-Rajib mall
3. UML2 and Unified process,2nd edition-Jim Arlow and IlaNeustadt
4. Wikiepedia-[www.wikiepedia.com](http://www.wikiepedia.com/)
5. Wikiepedia-[www.wikiepedia.com](http://www.wikiepedia.com/)
6. Database and management system-Navathe.