**INTERCOLLEGIATE EVENT MANAGEMENT SYSTEM**

###### A Project Report

Submitted in partial fulfillment of the Requirement for the award of the Degree of

#### MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

###### By

Dimple. A. Badger Seat No.:

**Under the esteemed guidance of Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Designation: Course Co-ordinator**

**Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Designation: Assistant Professor**

###### DEPARTMENT OF INFORMATION TECHNOLOGY

**JMF’s VANDE MATARAM DEGREE COLLEGE OF SCIENCE & COMMERCE**

***(Affiliated to University of Mumbai)***

###### Dombivali,421201 Maharashtra 2020-2021

**JMF’s VANDE MATARAM DEGREE COLLEGE OF SCIENCE & COMMERCE**

***(Affiliated to University of Mumbai)***

**DOMBIVALI-MAHARASHTRA-PINCODE**

#### DEPARTMENT OF INFORMATION TECHNOLOGY

**CERTIFICATE**

This is to certify that the project entitled, **“INTERCOLLEGIATE EVENTMANAGEMENT SYSTEM”,** is bonafied work of **DIMPLE. A. BADGER**

Bearing Seat no.: **(7403)** submitted in partial fulfillment of the requirements for the award of degree of MASTER OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

###### Internal Guide Co-ordinator

**External Examiner**

**Date: College Seal**

**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

***(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)***

PNR No.: ……………………………….. Roll no.:

1. Name of the Student
2. Title of the Project
3. Name of the Guide
4. Teaching experience of the Guide
5. Is this your first submission? Yes No

Signature of the Student Signature of the Guide

Date ……………………… Date…………………

Signature of the Coordinator Date………………………

**DECLARATION**

I hereby declare that the project entitled the **“Intercollegiate Event Management System”** done at “**Vande Mataram Degree College of Science &Commerce”** has not been in any case duplicated to submit to any other university for the award of any degree. According to me, yet nobody have been submitted this project in university.

The project is done in partial fulfilment of the requirements for the award of degree of

**MASTER OF SCIENCE(INFORMATION TECHNOLOGY)** to be submitted as

Semester- 4 project as a part of curriculum.

###### Dimple A Badger

**ABSTRACT**

Management is useful for maintaining data of every events organized in the respective colleges, login forms are there through which an account will be created automatically and after that the user can upload their college events once they uploaded their events automatically other colleges will get the notification that the particular college have been held some events. Admins and other users are provided with user id and password for using this software

Then the students can register themselves for the events through this system itself but they will not allowed to do any payment for their events. After this entire process the user can log out and terminate the process. This process makes work efficient and easy with less errors and majorly provides an effective service to the users.

## ACKNOWLEDGEMENT

A project is a creative work for each and everyone. A proper synchronization between the team members is must for completing the project successfully. I would like to extend my gratitude to, our **Principal Dr. Rajkumar Kolhe Sir** and all the staffs of our **Vande Mataram Degree College of Science and Commerce** for providing us moral support, conductive work environment which was needed to complete this project.

I would also like to thanks our Course Co-ordinator **Prof.\_\_\_\_\_\_\_\_** and all the faculties of IT Department for giving us the most needed guidance and continuous encouragement throughout the duration of the project and without them it would not have been possible to accomplish this project.

I am also extremely thankful to the **University of Mumbai** for having prescribed this project work to me as a part of the academic requirement in the final year of **Master of Science in Information Technology .**

Finally, I would like to thanks my group members and other friends who have been helped us to reach our project in a good state.

## TABLE OF CONTENTS

[CHAPTER 1: INTRODUCTION 10](#_TOC_250030)

* 1. [BACKGROUND 12](#_TOC_250029)
  2. [OBJECTIVES… 13](#_TOC_250028)
  3. PURPOSE, SCOPE & APLLICABILITY 14
     1. [PURPOSE 14](#_TOC_250027)
     2. [SCOPE 14](#_TOC_250026)
     3. [APPLICABILITY 14](#_TOC_250025)
  4. [ACHIEVEMENTS… 15](#_TOC_250024)

CHAPTER 2: SURVEY OF TECHNOLOGIES 16

* 1. [EXISTING SYSTEM 17](#_TOC_250023)
  2. [PROPOSED SYSTEM… 17](#_TOC_250022)

CHAPTER 3: REQUIREMENTS AND ANALYSIS… 18

* 1. [PROBLEM DEFINITION 19](#_TOC_250021)
  2. [REQUIREMENT SPECIFICATION 20](#_TOC_250020)
  3. [PLANNING AND SCHEDULING 21](#_TOC_250019)

[GANTT CHART](#_TOC_250018)

* 1. [SOFTWARE AND HARDWARE REQUIREMENTS… 22](#_TOC_250017)

[CHAPTER 4: SYSTEM DESIGN 23](#_TOC_250016)

* 1. SYSTEM MODULE 24
  2. UML Diagrams… 27
     1. Entity Relationship Diagram 27
     2. Class Diagram 30
     3. Use Case Diagram 32
     4. Sequence Diagram 34
     5. Activity Diagram 36
     6. DFD Diagram 38
  3. DATA SCHEMA… 40
  4. USER INTERFACE DESIGN 42

CHAPTER 5: IMPLEMENTATION & TESTING 47

* 1. [IMPLEMENTATION APPROACH 48](#_TOC_250015)
  2. [CODING DETAILS & CODE EFFICIENCY 50](#_TOC_250014)
  3. [TESTING APPROACHES 82](#_TOC_250013)
     1. [Unit Testing 83](#_TOC_250012)
     2. [Integration Testing 83](#_TOC_250011)
     3. [System Testing 84](#_TOC_250010)
     4. [Acceptance Testing 84](#_TOC_250009)
  4. [MODIFICATION & IMPROVEMENTS 85](#_TOC_250008)
  5. [TEST CASES 87](#_TOC_250007)

CHAPTER 6: RESULTS & DISCUSSION 91

* 1. [TEST REPORTS 92](#_TOC_250006)
  2. [USER DOCUMENTATION 94](#_TOC_250005)

CHAPTER 7: CONCLUSION 98

* 1. CONCLUSION 99
     1. [SIGNIFICANCE OF THE SYSTEM 100](#_TOC_250004)
  2. [LIMITATIONS OF THE SYSTEM 101](#_TOC_250003)
  3. [FUTURE SCOPE OF THE SYSTEM 102](#_TOC_250002)

[REFERENCES 103](#_TOC_250001)

[GLOSSARY 104](#_TOC_250000)

**CHAPTER 1**

#### INTRODUCTION

* 1. **BACKGR OUND 1.2OBJECTIVES**
  2. **PURPOSE, SCOPE & APLLICABILITY**
     1. **PURPOSE SCOPE**
     2. **1.3.3APPLICABILITY**
  3. **ACHIEVEMENTS**

## CHAPTER 1 INTRODUCTION

Intercollegiate Event Management System is application of project management for creation and development events, workshops etc. Our project brings out the manual process

i.e. the paper work of intercollegiate event management system which has been built using Asp.Net as a front end and SQL server as a backend.

The main purpose of this project is to make simpler the process of managing each event by providing a web interface for admin, teachers and students.Every college have to register to the system before they upload their events. If the college will not register they will not able to upload the events.Once they registered to the system they will be allowed to upload their respective events.

Here, the respective colleges has to upload their college events for e.g. Seminars, cultural fest, etc. through this different colleges will get the updates that the particular college have been organised an event. This events will be easily visible to those colleges who have been registered before. Other colleges will get the notification that in which college the events have been organised. After this other colleges can easily be notified. The students of other colleges can register for the events but they will not be allowed to pay their event money their itself the students have to pay the money for the events only when they visit to the respective college who are organising the events.

In our project we have also included an chat application i.e. the feedback zone. Here this part gives a very easy communication between the students and the co-ordinator of the event. Here the students can easily ask their queries without wasting their time in visiting different colleges even the students can give the feedback regarding the events in which they have been participated. The colleges can also views this reviews in the feedback zone.For easy and convenient processing we need to develop software, which works as per our requirements. While preparing this software we got to remember few things i.e. this software

must have a friendly environment, in other words it should not be much complicated to handle, it should have options for any modifications.

In all this system, makes the work of the students as well as colleges easy. The student will not be needed to visit different colleges for promoting their college events. In this manner the students time get save and they can easily promote their college events through this system in the college itself.Also it saves the paperwork of the college co-ordinators.

## Background:

As the title of our project itself “Intercollegiate Event Management System” says that our system has to manage the events for different colleges. So to keep the information correct, it will manage the status/information about the events. Different colleges can upload their events in our system, once they uploaded the events other colleges will be notified about the events and students can participate through the system itself but they will be not allowed to pay their events fees because it may happen that students who register themselves will not be able to come to the colleges and attend the events. So, its better that the registered students to pay their event amount on the same day of event.

In the existing system they required to maintain all records on paper, thus when we have to search any information related to events or the registered students it takes a lot of time and even human errors possibilities is more.Security was not maintained in the existing one.

## Objectives:

Our project is basically aimed in providing latest technologies and tendencies to the college sector services, so that all colleges can update their events to each other’s colleges through this system. To provide all the services required for organizing an event. It allows the user to add ,modify and delete their college details.

Our main aim of this project is to make and provide some features to register for the events, to cancel events, to protect the data of the user etc. To fulfil the requirements of user and manage the information of event and registration. To create auser-friendly which will be more efficient, saves time and be easy to use. This is an Internet based application that can be accessed by authenticated users. This system can be useful to search the previous year events , participants students name with their college name.There are features like notifications which will notify to other colleges regarding the events.

## Purpose, Scope and Applicability:

### Purpose:

The purpose of our project is to shorten the procedure of managing each event by providing a web interface of admin,teachers and students.The another main purpose is that its gonna save time of students and teachers both in terms like the students are not needed to visit to different colleges and in this way they will not miss their lectures and teachers who are the main co-ordinators of the events will not waste their time in making the lists of the colleges in which students have to visit for promoting their events.

It can lead to error-free, secure, reliable,fast management system. The college can maintain computerized records without redundant entries.It also maintains the relationship between the different colleges.

### Scope:

This web application includes the IT fields. Normal users are the participants. They have to register for the respective events. Students could easily register the events of their choice. Easy to access the system anywhere and anytime. User will have the easy communication with the manager and gets instant message about any change updates easily.

This application leads to provide the user especially the students the reliability in finding the schedule of events at one place rather than to go for each of the college websites.

### 1.3.3.Applicability:

Our project is going to be useful to teachers and students both. As teachers will not needed to do any paper work like making the list of the students who want to participate in the events and even they will not need to send students to different colleges.

As students have to visit to different colleges for promoting their college events they have to miss their lectures which affects their studies. This system saves the time and energy of student so that the students will not able to rome different colleges.As well as our system

will notify about the events going to be organized in different colleges.This is user friendly and it also provides security to data.

### Achievements

While we were performing this project we had got a lot of knowledge about the software’s- Microsoft Visual Studio 2010 and MS-SQL Server ;which we have been used for the project and also about its versions and features. We had a good knowledge about the coding too.

Hence, this was the knowledge we had gained while performing this project.

###### Goals Achieved:

- The goals have been partially achieved.

# CHAPTER 2

#### SURVEY OF TECHNOLOGIES

* 1. **EXISTING SYSTEM**
  2. **PROPOSED SYSTEM**

**CHAPTER 2**

**SURVEY OF TECHNOLOGIES**

* 1. **Existing system**

Existing process is manual process.Manual process requires more man power.Manual process requires many records to maintain. College authorities needs to take care to store each and every records of the students and details of the respective colleges in which the events are organized.Here, Data Security is not provided and even integrating a data is also a problem in the existing system.

* 1. **Proposed system**

Our Proposed system is a web application.In this application student details are maintained efficiently admin has a facilityto view the student detailslike students name, from which college students belong from, their contact number, and the details about the events in which the students are going to participate. This has an enhanced facility.It is a fast, reliable system.

# CHAPTER 3

#### REQUIREMENTS AND ANALYSIS

* 1. **PROBLEM DEFINITION**
  2. **REQUIREMENT SPECIFICATION**
  3. **PLANNING AND SCHEDULING**
  4. **GANTT CHART**
  5. **SOFTWARE AND HARDWARE REQUIREMENTS**

**CHAPTER 3 REQUIREMENT AND ANALYSIS**

* 1. **Problem Definition**

Before coming new technology we used to face many problems like wastage of time and energy and many other things. So now here finally we have a new technologies with us, the work have been became easier.Previously, Intercollegiate Event Management System were very difficult and complicated to handle and also to maintain the data was very difficult and it was used to consume a lot time and man power.

So the invented technologies provides us ease for completing any project.As if we know that the existing system was very complicated and time consuming because it is done by manual process we aretrying to build this project in such a way that it should reduce the man power and time.

The problems are:

* Data security is not provided in existing system whereas here this project is providing a data security as if only authenticated users can only access the system not others.
* Difficult to insert, modify and delete the records because there can happen some human errors
* The major problem was faced by students and teachers because students used to visit different colleges for promoting their events which consumes a lot time and even students used to miss their lectures and teachers have to take their lectures which even waste the time of professors etc.
* Paper work is more and it becomes tedious to handle it.
  1. **Requirements Specification**

A software requirements specification is a descriptionof a software system to be developed and also it describes the nature of project ,software, or application. It rests out functional and non-functional requirements, and may include a set of use cases that describe user interaction that the software must provide. In addition to this, it also covers the information aboutsafety and security requirements, software quality attributes of the project.

This system is capable of storing the data of events and all the students (users). Admin can add the event and can add the students. Admin will give the access to the students to register for the events. This system is portable i.e. it can run on any operating system whether it is 7/8/10 and even it is reliable and performance of the speed is high. It provides the Security so that our data will not leak.

### Planning and scheduling

This phase describes the tasks and steps necessary to plan and schedule the activities that we have should perform to achieve a successful completion of a project. A Project plan is prepared in such a manner which defines it assummary document that describes everything about project in terms of its objectives and how the objectives are to be achieved.

We have choosen this topic to overcome the problems like – previous system does not run efficiently, it was having a need of man power and difficult to response every user within short period of time ,thus we got to know that there is aa important need to make it automated.

**Step 1:** The task of IEMS is to reduce the man power and to do work paperless.In this we create data about events and users in these system. We have also created a login id and password for the users.

**Step 2:**The major task of our project is to complete the work easily and paperless.

**Step 3:**Topic selection: 4 days have been taken to select the topic

**Discuss :**4 days have been taken to discuss about the topic

**Chapter 1:** 6 days have been taken for chapter 1 in which we have shown in introduction,Objectives,Purpose,Scope,Applicability,Achievements.

**Chapter 2:**2 days have been taken for survey of technologies.

**Chapter 3:** Appearing

###### Gantt Chart

A Gantt Chart is a horizontal bar chart developed as a production tool in 1917 by Henry L. Gantt, an American engineer and social scientist.Frequently, used in project management, a Gantt chart provides a graphical illustration of a schedule that helps to complete a specific tasks in a project.

### Software and Hardware Requirements

###### Software Requirement:

* + - Asp.net with C# by using Microsoft visual studio 2010.
    - MS SQL server 2008 for database connection.

###### Hardware Requirements:

* + - * Operating system : windows 7/8/10
      * RAM : 500 MB
      * Processor:Intel® Pentium®

###### Features:

* Protect the database of the firm by requiring a correct and registered username and password.
* Provide an option for users to update information.
* Create new events.
* Provide an easy function where you can navigate the forms whenever necessary.
* User friendly.

# CHAPTER 4

#### SYSTEM MODULE

* 1. **UML Diagrams**
     1. **Entity Relationship Diagram**
     2. **Class Diagram**
     3. **Use Case Diagram**
     4. **Sequence Diagram**
     5. **Activity Diagram**
     6. **DFD Diagram**
  2. **DATA SCHEMA**
  3. **USER INTERFACE DESIGN**

## CHAPTER 4 SYSTEM DESIGN

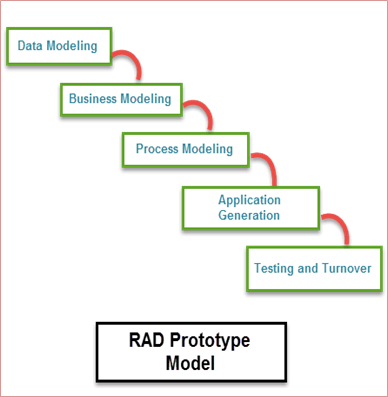
* 1. **System Model**

**RAPID APPLICATION DEVELOPMENT MODEL**

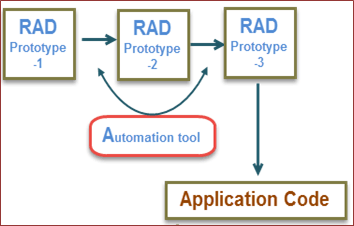
RAD or Rapid Application Development process is an adoption of the waterfall model; it targets at developing software in a short span of time. RAD follow the iterative

SDLC RAD model has following phases

* Business Modeling
* Data Modeling
* Process Modeling
* Application Generation
* Testing and Turnover



It focuses on input-output source and destination of the information. It emphasizes on delivering projects in small pieces; the larger projects are divided into a series of smaller projects. The main features of RAD model are that it focuses on the reuse of templates, tools, processes, and code.



###### RAD Model in Software Engineering

**Different phases of RAD model includes:**

**Business Modeling:** On basis of the flow of information and distribution between various business channels, the product is designed.

**Data Modeling:** The information collected from business modeling is refined into a set of data objects that are significant for the business.

**Process Modeling:** The data object that is declared in the data modeling phase is transformed to achieve the information flow necessary to implement a business function.

**Application Generation:** Automated tools are used for the construction of the software, to convert process and data models into prototypes.

**Testing and Turnover:** As prototypes are individually verified during every iteration, the overall testing time is reduced in RAD.

**When to use RAD Methodology:**

* When a system needs to be produced in a short span of time (2-3 months)
* When the requirements are known
* When the user will be involved all through the life cycle
* When technical risk is less
* When there is a necessity to create a system that can be modularized in 2-3 months of time
* When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation

#### Advantages-

Flexible and adaptable to changes

It is useful when you have to reduce the overall project risk It is adaptable and flexible to changes

It is easier to transfer deliverables as scripts, high-level abstractions and intermediate codes are used

Due to code generators and code reuse, there is a reduction of manual coding. Due to prototyping in nature, there is a possibility of lesser defects.

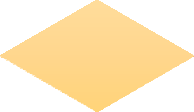
Each phase in RAD delivers highest priority functionality to client With less people, productivity can be increased in short time **Disadvantages-**

It can't be used for smaller projects

When technical risk is high, it is not suitable Requires highly skilled designers or developers

Reduced scalability occurs because a RAD developed application begins as a prototype and evolves into a finished application.

* 1. **UML DIAGRAMS ER-DIAGRAM**



An Entity Relationship diagram shows the relationships of entity sets stored in a database. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

###### Symbols:

* + 1. **Entity -**Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.



* + 1. **Attributes** -Attibutes are the properties of entities. Attributes are represented by means of ellipses. Attributes are of 2 types:
       1. **Composite Attributes:** They are further divided in tree like structure.Every node is then connected to its attribute.



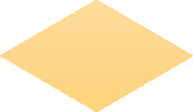
* + - 1. **Derived Attributes:** It is depicted By double ellipse.



* + 1. **Relationships-** Relationships are represented by diamond shapes. All the entities Participating in a relationship, are connected to it by a line.

###### --Binary Relationship and Cardinality:

1. **One-to-One :**When only one instance of a entity is associated with a relationship.

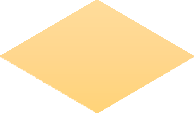


~~1~~   ~~1~~

1. **One -to-many :** When more than one instance of an entity is associated with a relationship.

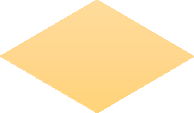
1 N

1. **Many -to-one :** When more than one instance of an entity is associated with a relationship.



N 1

1. **Many- to -Many :**When more than one instance of an entity on the left and more than one instance of an entity on the right can be associated.



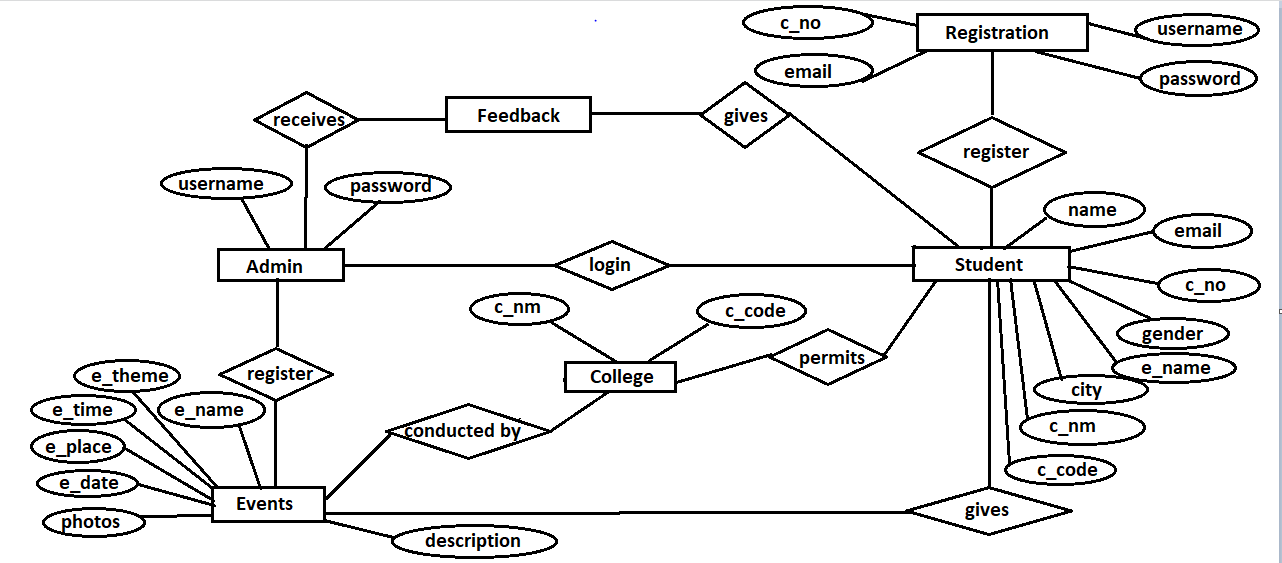
N

N

###### --Participation Constraints:

1. **Total Participation:** Each entity is involved in a relationship. It is represented by double lines.
2. **Partial Participation:** Not all entities are involved in a relationship. It is

represented by single line.

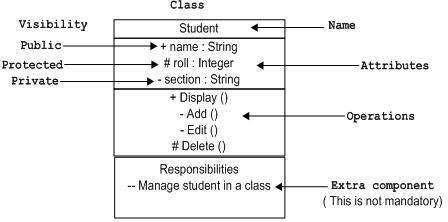


### CLASS DIAGRAM

###### Class Notation:

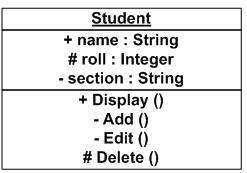
A class represented by the following figure. This diagram is divided into 4 parts:

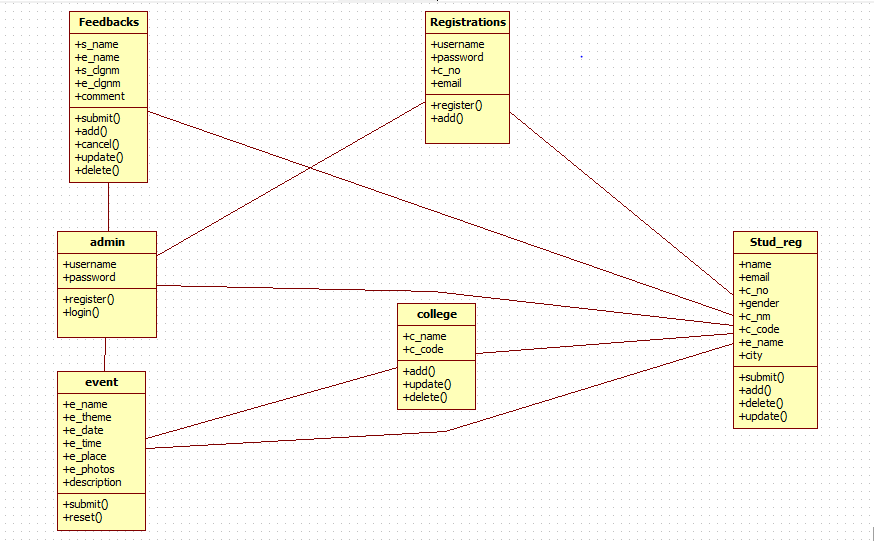
* The top section is used to name the class.
* The second one is used to show the attributes of the class.
* The third section is used to describe the operations performed by the class.
* The fourth section is optional to show any additional components.



###### Object Notation:

The *object* is represented in the same way as class. The only difference is the *name* which is underlined as shown in the figure. As the object is an actual implementation of a class, which is known as the instance of the class. Hence, it has the same usage as class.





### USE CASE DIAGRAM

A use case diagram as its simplest is a representation of a user’s interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. It is a requirement analysis concept. Describes the actions from a point of view of user . A sequence of events is involving. Interactions of a user with the system specifies one aspect of the behaviour of a system, without specifying the structure of the system.

A use case diagram contains 4 components:

* **Actor:** Actor represents roles that take on when they use the IT system. Actors can be person, another system or organization.
* **Use case:** Use case describes the interactions that take place between actors and Systems during the execution.
* **Association:** An association is a connection between an actor and a use case. An association indicates that an actor can carry out a use case. Several actors at one use case mean that each actor can carry out the use case.
* **Include Relationships:** An include relationship is a relationship between two cases:

<<include>>

It indicates that the use case to which the arrow points is included in the use case on the other side of the arrow. This makes it possible to reuse a use case in another use case.



### SEQUENCE DIAGRAM

Sequence diagram is an “interaction diagram” that models a single scenario executing in a system.

###### Key Parts:

* **Participant –** An object or an entity , the sequence diagram actor.
* **Sequence –** Diagram starts with an unattached “found message” arrow.
* **Message –** Communication between objects.

###### Syntax and Semantics:

**An object :** a box with an underlined label that specifies the object type and optionally the object name.

* + Write the object’s name if it clarifies the diagram.
* An object’s “life line” is represented by a dashed vertical line.
* Represents the life span of the object during the scenario being modelled.

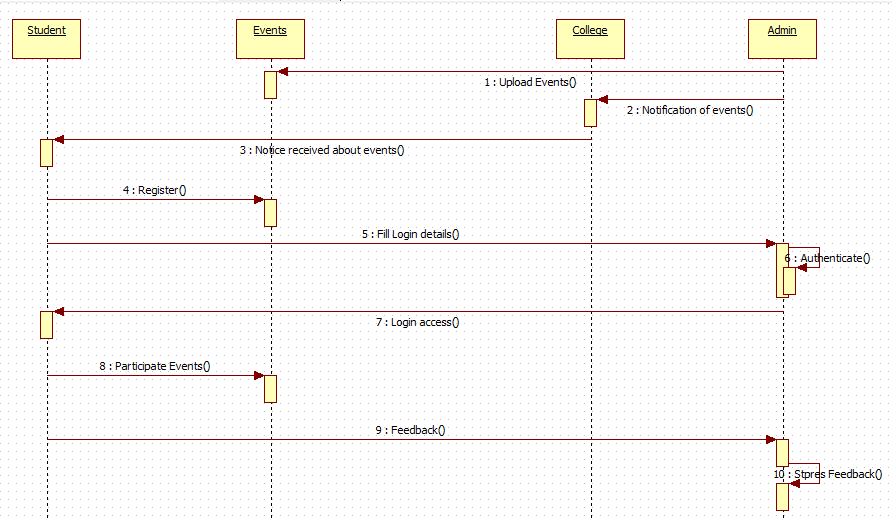
###### Representing messages between objects:

A message: an horizontal arrow to the receiving object. Write message name and arguments above the arrow.

###### Different types of messages:

Type of arrow indicates types of messages:

* Synchronous message: Solid arrow with solid head
* Asynchronous message: Solid arrow with stick head.
* Return message: Dashed arrow with stick head.



### ACTIVITY DIAGRAM

The activity diagram is used to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as the operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagram deals with all type of flow controls by using different elements such as fork, join etc.

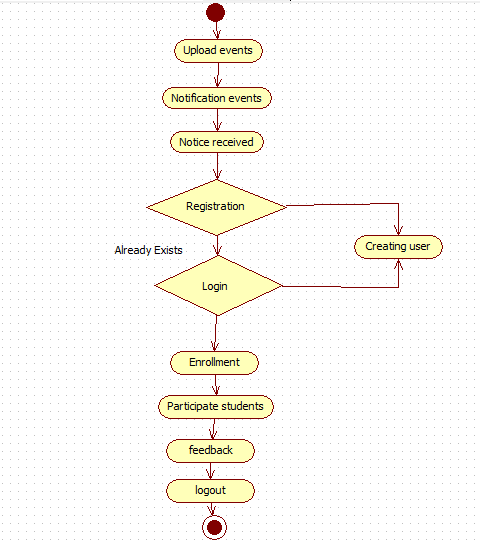
###### Purpose:

* Draw the activity flow of a system
* Describe the sequence from one activity to another.
* Describe the parallel, branched, concurrent flow of the system.

Before drawing an activity diagram, we need to identify the following elements:

* Activities
* Association
* Conditions
* Constraints

Once the above mentioned parameters are identified, we need to make a mental layout of the entire flow.



### DFD DIAGRAM

A DFD diagram illustrates how data is processed by a system in terms of inputs and outputs.As it name indicates its focus is on the flow of information, where data comes from, where data goes, and how it gets stored.

###### Symbols:

There are essentially 2 different types

###### Process Notations:

A process transforms incoming data flow into outgoing data flow.

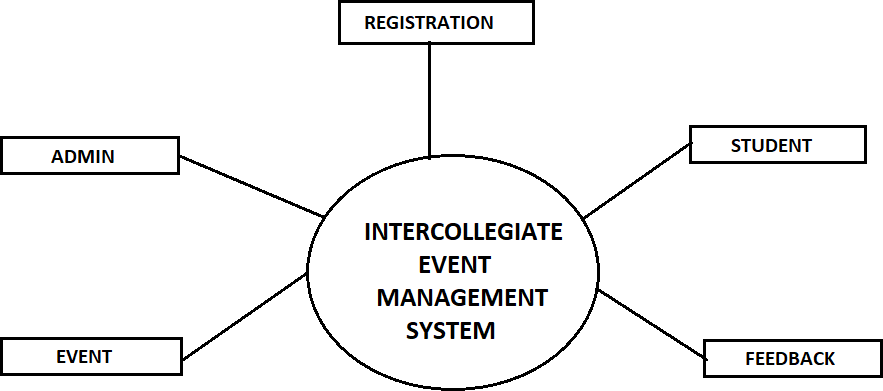


###### Datastores Notations:

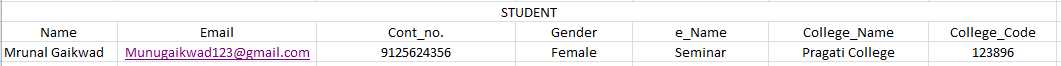
Datastores are repositories of data in the system. They are sometimes referred to as

files.

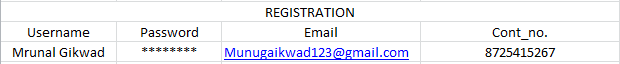




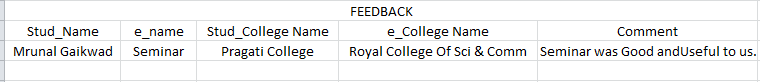
* 1. **DATA SCHEMA: Student table:-**



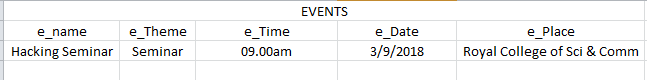
**Registration table:-**



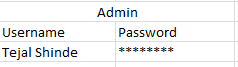
**SMS table:-**



**Event table:-**

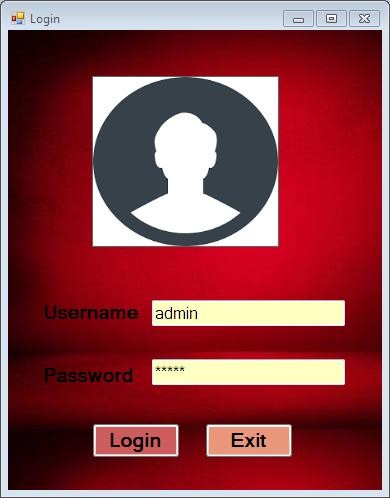


**Login(Admin) table:-**



* 1. **USER INTERFACE**

#### Login:



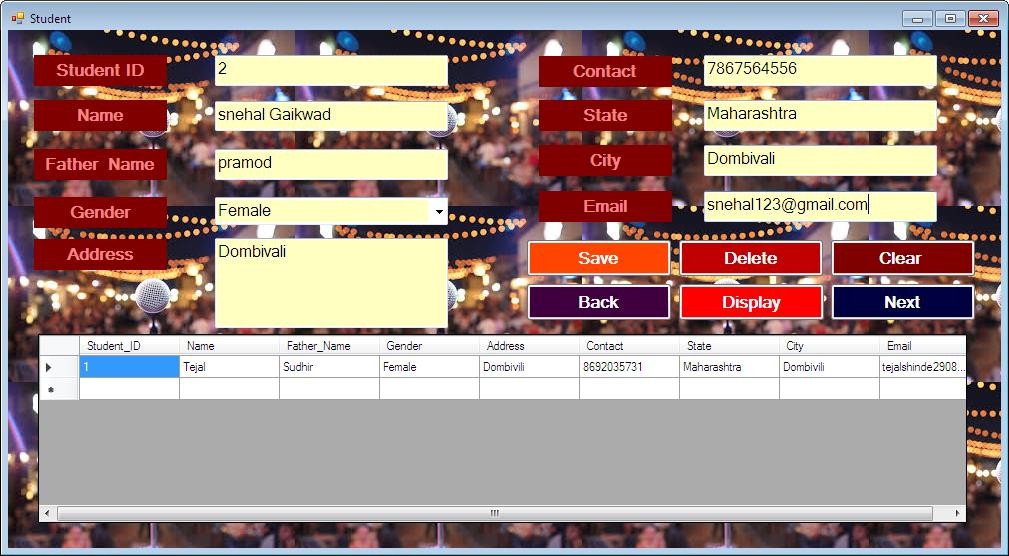
**Dashboard:**



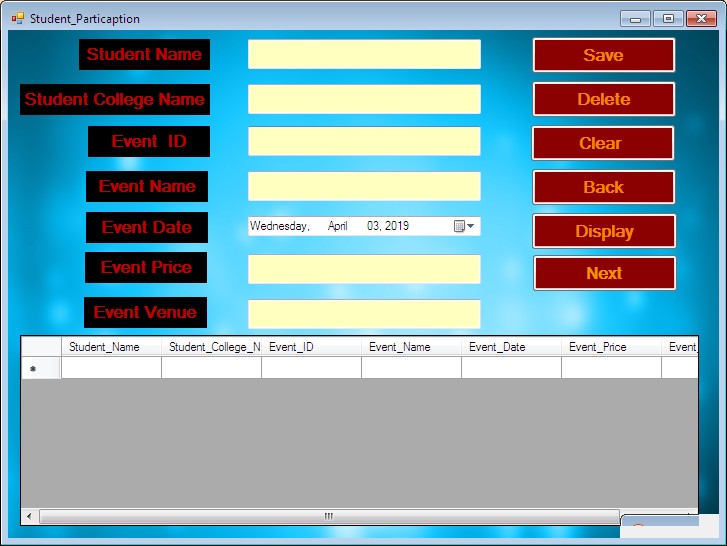
**About\_Us:**



**Student Registration:**



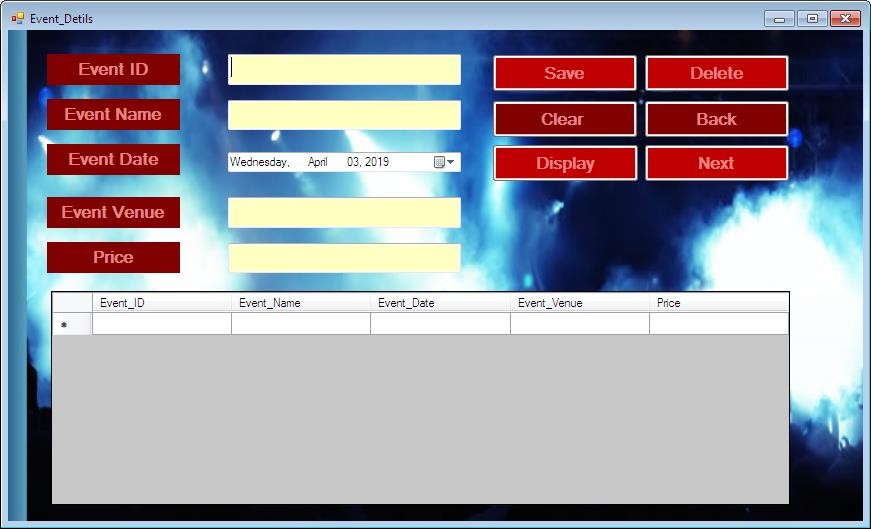
**Student Participation:**



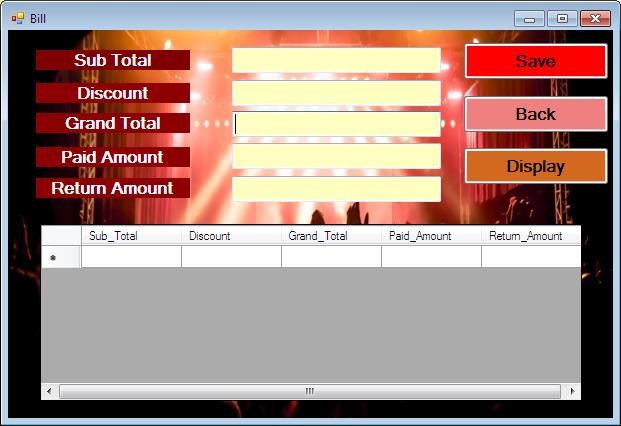
**Events:**



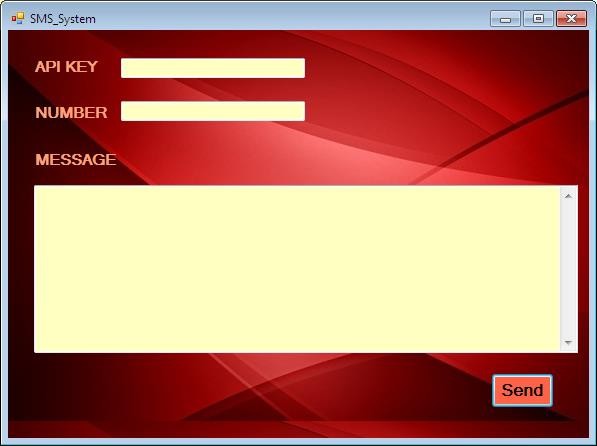
**Events\_Details:**



**Bill:**



**SMS\_System:**



**CHAPTER 5**

#### IMPLEMENTATION APPROACH

#### CODING DETAILS & CODE EFFICIENCY

#### TESTING APPROACHES

#### Unit Testing

#### Integration Testing

#### System Testing

#### Acceptance Testing

#### MODIFICATION & IMPROVEMENTS

#### TEST CASES

**CHAPTER 5**

**IMPLEMENTATION & TESTING**

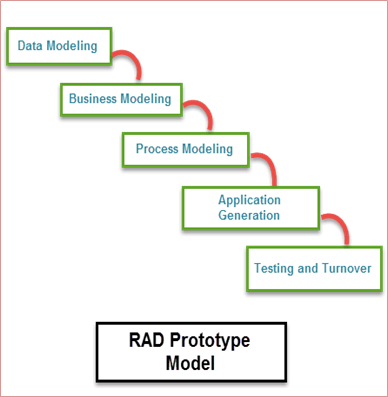
* 1. **IMPLEMENTATION APPROACH**

Method which will be used for “INTERCOLLEGIATE EVENT MANAGEMENT SYSTEM” is“RAPID APPLICATION DEVELOPMENT”.

Rapid application development is an software development model/methodology that uses minimum planning in favour of rapid prototyping. In these, the functional modules ae developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since, there is no pre-planning, it makes it easier to incorporate the changes within the development process.

We have implemented this model in our project because our project is broken down into small modules

Our project though it follows RAD model, the most important aspect for this model to be successful is to make sure that the protypes developed are reusable.



The phases in the rapid application development (RAD) model are:

**Business modeling:** The information flow is identified between various business functions.

**Data modeling:** Information gathered from business modeling is used to define data objects thatare needed for the business.

**Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.

**Application generation:** Automated tools are used to convert process models into code and the actual system.

**Testing and turnover:** Test new components and all the interfaces.

**When to use RAD model:**

* + - RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.
    - It should be used if there’s high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools.
    - RAD [SDLC model](http://tryqa.com/what-are-the-software-development-models/) should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

#### CODING DETAILS & CODE EFFICIENCY Login.cs

using System; usingSystem.Collections.Generic; usingSystem.ComponentModel; usingSystem.Data; usingSystem.Drawing; usingSystem.Linq; usingSystem.Text; usingSystem.Windows.Forms; usingSystem.Data.SqlClient; using System.IO;

namespaceintercollege\_management\_system

{

publicpartialclassLogin :Form

{

publicLogin()

{

InitializeComponent();

}

publicstringconnstring = "Data Source=DESKTOP- CN9M6DO;InitialCatalog=intercollege;Integrated Security=True";

SqlCommandcmd;

//login btn

privatevoid button1\_Click(object sender, EventArgs e)

{

SqlConnection conn = newSqlConnection(connstring); conn.Open();

SqlDataAdaptersda = newSqlDataAdapter("select count(\*) from Login where Username='" + textBox1.Text + "'and Password='" + textBox2.Text + "'", conn);

DataTable dt = newDataTable(); sda.Fill(dt);

if (dt.Rows[0][0].ToString() == "1")

{

this.Hide();

Dashboard ds = newDashboard(); ds.Show();

}

else

{

MessageBox.Show("please check username and password");

}

}

//exit btn

privatevoid button2\_Click(object sender, EventArgs e)

{

Application.Exit(); } }}

**Dashboard.cs**

using System; usingSystem.Collections.Generic; usingSystem.ComponentModel; usingSystem.Data; usingSystem.Drawing; usingSystem.Linq; usingSystem.Text; usingSystem.Windows.Forms;

namespaceintercollege\_management\_system

{

publicpartialclassDashboard :Form

{

publicDashboard()

{

InitializeComponent();

}

privatevoid toolStripButton1\_Click(object sender, EventArgs e)

{

Student s = newStudent(); s.Show();

}

privatevoid toolStripButton2\_Click(object sender, EventArgs e)

{

Event\_Detils ed = newEvent\_Detils(); ed.Show();

}

privatevoid toolStripButton3\_Click(object sender, EventArgs e)

{

Student\_Particaptionsp = newStudent\_Particaption(); sp.Show();

}

privatevoid toolStripButton4\_Click(object sender, EventArgs e)

{

Events EE = newEvents(); EE.Show();

}

privatevoid toolStripButton5\_Click(object sender, EventArgs e)

{

Bill B = newBill(); B.Show();

}

privatevoid toolStripButton6\_Click(object sender, EventArgs e)

{

SMS\_System ss = newSMS\_System(); ss.Show();

}

privatevoid toolStripButton7\_Click(object sender, EventArgs e)

{

Report RR = newReport(); RR.Show();

}

privatevoid toolStripButton8\_Click(object sender, EventArgs e)

{

About\_Us au = newAbout\_Us(); au.Show();

}

privatevoid toolStripButton9\_Click(object sender, EventArgs e)

{

Help h = newHelp(); h.Show();

}

privatevoid toolStripButton10\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

**Student.cs**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Windows.Forms; using System.Data.SqlClient; using System.IO;

namespace intercollege\_management\_system

{

public partial class Student : Form

{

public Student()

{

InitializeComponent();

}

public string connstring = "Data Source=DESKTOP- CN9M6DO;InitialCatalog=intercollege;Integrated Security=True";

SqlCommandcmd;

public void display\_data()

{

SqlConnection conn = new SqlConnection(connstring); conn.Open();

SqlCommandcmd = conn.CreateCommand(); cmd.CommandType = CommandType.Text; cmd.CommandText = "select \* from [Student]"; cmd.ExecuteNonQuery();

DataTabledta = new DataTable(); SqlDataAdaptersda = new SqlDataAdapter(cmd); sda.Fill(dta);

dataGridView1.DataSource = dta; conn.Close();

}

//save btn

private void button1\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection(connstring); conn.Open();

string sqlQuery = "insert into Student(Student\_ID,Name,Father\_Name,Gender,Address,Contact,State,City,Email)values('"

+ txt1.Text + "','" + txt2.Text + "','" + txt3.Text + "','" + txt4.Text + "','" + txt5.Text + "','" +

txt6.Text + "','" + txt7.Text + "','" + txt8.Text + "','" + txt10.Text + "')"; cmd = new SqlCommand(sqlQuery, conn);

int N = cmd.ExecuteNonQuery(); conn.Close();

MessageBox.Show(N.ToString() + "Data saved Successfully !!");

display\_data();

}

//dltbtn

private void button3\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection(connstring); conn.Open();

SqlCommandcmd = conn.CreateCommand(); cmd.CommandType = CommandType.Text;

cmd.CommandText = "delete from [Student]where Student\_ID='" + txt1.Text + "'"; cmd.ExecuteNonQuery();

conn.Close(); display\_data();

}

//bckbtn

private void button6\_Click(object sender, EventArgs e)

{

this.Close();

Dashboard d = new Dashboard(); d.Show();

}

//display btn

private void button2\_Click(object sender, EventArgs e)

{

display\_data();

}

//clrbtn

private void button4\_Click(object sender, EventArgs e)

{

txt1.Clear();

txt2.Clear();

txt3.Clear();

txt5.Clear();

txt6.Clear();

txt7.Clear();

txt8.Clear(); txt10.Clear();

}

//next btn

private void Student\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'intercollegeDataSet1.Student' table.

You can move, or remove it, as needed. this.studentTableAdapter.Fill(this.intercollegeDataSet1.Student);}}}

**Event details.cs**

namespace intercollege\_management\_system

{

partial class Event\_Detils

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.components = new System.ComponentModel.Container(); System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Event\_Detils));

this.txt2 = new System.Windows.Forms.TextBox(); this.txt1 = new System.Windows.Forms.TextBox();

this.label25 = new System.Windows.Forms.Label(); this.label26 = new System.Windows.Forms.Label(); this.label1 = new System.Windows.Forms.Label(); this.textBox1 = new System.Windows.Forms.TextBox(); this.textBox2 = new System.Windows.Forms.TextBox(); this.label2 = new System.Windows.Forms.Label(); this.label3 = new System.Windows.Forms.Label(); this.button3 = new System.Windows.Forms.Button(); this.button1 = new System.Windows.Forms.Button(); this.button4 = new System.Windows.Forms.Button(); this.button6 = new System.Windows.Forms.Button(); this.button5 = new System.Windows.Forms.Button(); this.button2 = new System.Windows.Forms.Button();

this.dateTimePicker1 = new System.Windows.Forms.DateTimePicker(); this.dataGridView1 = new System.Windows.Forms.DataGridView();

this.eventIDDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn(); this.eventNameDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn(); this.eventDateDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn(); this.eventVenueDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn(); this.priceDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();

this.eventdetalBindingSource = new System.Windows.Forms.BindingSource(this.components);

this.intercollegeDataSet3 = new intercollege\_management\_system.intercollegeDataSet3(); this.eventdetalTableAdapter = new intercollege\_management\_system.intercollegeDataSet3TableAdapters.eventdetalTableAdapt er();

((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.eventdetalBindingSource)).BeginInit(); ((System.ComponentModel.ISupportInitialize)(this.intercollegeDataSet3)).BeginInit(); this.SuspendLayout();

//

// txt2

//

this.txt2.BackColor = System.Drawing.Color.White;

this.txt2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.txt2.Location = new System.Drawing.Point(220, 70);

this.txt2.Multiline = true; this.txt2.Name = "txt2";

this.txt2.Size = new System.Drawing.Size(233, 30); this.txt2.TabIndex = 238;

//

// txt1

//

this.txt1.BackColor = System.Drawing.Color.White;

this.txt1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.txt1.Location = new System.Drawing.Point(220, 24);

this.txt1.Multiline = true; this.txt1.Name = "txt1";

this.txt1.Size = new System.Drawing.Size(233, 31); this.txt1.TabIndex = 237;

//

// label25

//

this.label25.BackColor = System.Drawing.Color.Black;

this.label25.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label25.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.label25.Location = new System.Drawing.Point(39, 69); this.label25.Name = "label25";

this.label25.Size = new System.Drawing.Size(133, 31); this.label25.TabIndex = 236;

this.label25.Text = "Event Name";

this.label25.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

//

// label26

//

this.label26.BackColor = System.Drawing.Color.Black;

this.label26.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.label26.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.label26.Location = new System.Drawing.Point(39, 24); this.label26.Name = "label26";

this.label26.Size = new System.Drawing.Size(133, 31); this.label26.TabIndex = 235;

this.label26.Text = "Event ID";

this.label26.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

//

// label1

//

this.label1.BackColor = System.Drawing.Color.Black;

this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.label1.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.label1.Location = new System.Drawing.Point(39, 114); this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(133, 31);

this.label1.TabIndex = 239; this.label1.Text = "Event Date";

this.label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

//

// textBox1

//

this.textBox1.BackColor = System.Drawing.Color.White;

this.textBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.textBox1.Location = new System.Drawing.Point(220, 213);

this.textBox1.Multiline = true; this.textBox1.Name = "textBox1";

this.textBox1.Size = new System.Drawing.Size(233, 30); this.textBox1.TabIndex = 243;

//

// textBox2

//

this.textBox2.BackColor = System.Drawing.Color.White;

this.textBox2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.textBox2.Location = new System.Drawing.Point(220, 167);

this.textBox2.Multiline = true; this.textBox2.Name = "textBox2";

this.textBox2.Size = new System.Drawing.Size(233, 31); this.textBox2.TabIndex = 242;

//

// label2

//

this.label2.BackColor = System.Drawing.Color.Black;

this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.label2.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.label2.Location = new System.Drawing.Point(39, 212); this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(133, 31); this.label2.TabIndex = 241;

this.label2.Text = "Price";

this.label2.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

//

// label3

//

this.label3.BackColor = System.Drawing.Color.Black;

this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.label3.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.label3.Location = new System.Drawing.Point(39, 167); this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(133, 31); this.label3.TabIndex = 240;

this.label3.Text = "Event Venue";

this.label3.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

//

// button3

//

this.button3.BackColor = System.Drawing.SystemColors.ActiveCaptionText; this.button3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.button3.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.button3.Location = new System.Drawing.Point(636, 24); this.button3.Name = "button3";

this.button3.Size = new System.Drawing.Size(146, 38);

this.button3.TabIndex = 251;

this.button3.Text = "Delete"; this.button3.UseVisualStyleBackColor = false;

this.button3.Click += new System.EventHandler(this.button3\_Click); this.button1.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.button1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.button1.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.button1.Location = new System.Drawing.Point(484, 24); this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(146, 38); this.button1.TabIndex = 250;

this.button1.Text = "Save"; this.button1.UseVisualStyleBackColor = false;

this.button1.Click += new System.EventHandler(this.button1\_Click); this.button4.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.button4.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.button4.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.button4.Location = new System.Drawing.Point(484, 70); this.button4.Name = "button4";

this.button4.Size = new System.Drawing.Size(146, 38); this.button4.TabIndex = 253;

this.button4.Text = "Clear "; this.button4.UseVisualStyleBackColor = false;

this.button4.Click += new System.EventHandler(this.button4\_Click);

//

// button6

//

this.button6.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.button6.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.button6.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.button6.Location = new System.Drawing.Point(636, 70); this.button6.Name = "button6";

this.button6.Size = new System.Drawing.Size(146, 38); this.button6.TabIndex = 254;

this.button6.Text = "Back"; this.button6.UseVisualStyleBackColor = false;

this.button6.Click += new System.EventHandler(this.button6\_Click); this.button5.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.button5.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.button5.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.button5.Location = new System.Drawing.Point(636, 114); this.button5.Name = "button5";

this.button5.Size = new System.Drawing.Size(146, 38); this.button5.TabIndex = 256;

this.button5.Text = "Next"; this.button5.UseVisualStyleBackColor = false;

this.button5.Click += new System.EventHandler(this.button5\_Click); this.button2.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.button2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0))); this.button2.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(128)))), ((int)(((byte)(0)))));

this.button2.Location = new System.Drawing.Point(484, 114); this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(147, 38); this.button2.TabIndex = 255;

this.button2.Text = "Display";

this.button2.UseVisualStyleBackColor = false;

this.button2.Click += new System.EventHandler(this.button2\_Click); this.dateTimePicker1.Location = new System.Drawing.Point(220, 122); this.dateTimePicker1.Name = "dateTimePicker1"; this.dateTimePicker1.Size = new System.Drawing.Size(233, 20); this.dateTimePicker1.TabIndex = 257; this.dataGridView1.AutoGenerateColumns = false;

this.dataGridView1.AutoSizeColumnsMode =

System.Windows.Forms.DataGridViewAutoSizeColumnsMode.Fill; this.dataGridView1.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize; this.dataGridView1.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] { this.eventIDDataGridViewTextBoxColumn,

this.eventNameDataGridViewTextBoxColumn, this.eventDateDataGridViewTextBoxColumn, this.eventVenueDataGridViewTextBoxColumn, this.priceDataGridViewTextBoxColumn}); this.dataGridView1.DataSource = this.eventdetalBindingSource; this.dataGridView1.Location = new System.Drawing.Point(43, 261); this.dataGridView1.Name = "dataGridView1"; this.dataGridView1.Size = new System.Drawing.Size(739, 214); this.dataGridView1.TabIndex = 258;

this.eventIDDataGridViewTextBoxColumn.DataPropertyName = "Event\_ID"; this.eventIDDataGridViewTextBoxColumn.HeaderText = "Event\_ID"; this.eventIDDataGridViewTextBoxColumn.Name = "eventIDDataGridViewTextBoxColumn"; this.eventNameDataGridViewTextBoxColumn.DataPropertyName = "Event\_Name"; this.eventNameDataGridViewTextBoxColumn.HeaderText = "Event\_Name"; this.eventNameDataGridViewTextBoxColumn.Name = "eventNameDataGridViewTextBoxColumn"; this.eventDateDataGridViewTextBoxColumn.DataPropertyName = "Event\_Date"; this.eventDateDataGridViewTextBoxColumn.HeaderText = "Event\_Date";

this.eventDateDataGridViewTextBoxColumn.Name = "eventDateDataGridViewTextBoxColumn"; this.eventVenueDataGridViewTextBoxColumn.DataPropertyName = "Event\_Venue"; this.eventVenueDataGridViewTextBoxColumn.HeaderText = "Event\_Venue"; this.eventVenueDataGridViewTextBoxColumn.Name = "eventVenueDataGridViewTextBoxColumn"; this.priceDataGridViewTextBoxColumn.DataPropertyName = "Price"; this.priceDataGridViewTextBoxColumn.HeaderText = "Price"; this.priceDataGridViewTextBoxColumn.Name = "priceDataGridViewTextBoxColumn"; this.eventdetalBindingSource.DataMember = "eventdetal"; this.eventdetalBindingSource.DataSource = this.intercollegeDataSet3; this.intercollegeDataSet3.SchemaSerializationMode = System.Data.SchemaSerializationMode.IncludeSchema; this.eventdetalTableAdapter.ClearBeforeFill = true;

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F); this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackgroundImage =

((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage"))); this.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch; this.ClientSize = new System.Drawing.Size(855, 491); this.Controls.Add(this.dataGridView1); this.Controls.Add(this.dateTimePicker1);

this.Controls.Add(this.button5); this.Controls.Add(this.button2); this.Controls.Add(this.button6); this.Controls.Add(this.button4); this.Controls.Add(this.button3); this.Controls.Add(this.button1); this.Controls.Add(this.textBox1); this.Controls.Add(this.textBox2); this.Controls.Add(this.label2); this.Controls.Add(this.label3); this.Controls.Add(this.label1);

this.Controls.Add(this.txt2); this.Controls.Add(this.txt1); this.Controls.Add(this.label25); this.Controls.Add(this.label26); this.Name = "Event\_Detils";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen; this.Text = "Event\_Detils";

this.Load += new System.EventHandler(this.Event\_Detils\_Load); ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.eventdetalBindingSource)).EndInit(); ((System.ComponentModel.ISupportInitialize)(this.intercollegeDataSet3)).EndInit();

this.ResumeLayout(false); this.PerformLayout(); } #endregion

private System.Windows.Forms.TextBox txt2; private System.Windows.Forms.TextBox txt1; private System.Windows.Forms.Label label25; private System.Windows.Forms.Label label26; private System.Windows.Forms.Label label1;

private System.Windows.Forms.TextBox textBox1; private System.Windows.Forms.TextBox textBox2; private System.Windows.Forms.Label label2; private System.Windows.Forms.Label label3; private System.Windows.Forms.Button button3; private System.Windows.Forms.Button button1; private System.Windows.Forms.Button button4; private System.Windows.Forms.Button button6; private System.Windows.Forms.Button button5; private System.Windows.Forms.Button button2;

private System.Windows.Forms.DateTimePicker dateTimePicker1; private System.Windows.Forms.DataGridView dataGridView1; private intercollegeDataSet3 intercollegeDataSet3;

private System.Windows.Forms.BindingSourceeventdetalBindingSource;

private intercollegeDataSet3TableAdapters.eventdetalTableAdapter eventdetalTableAdapter;

private System.Windows.Forms.DataGridViewTextBoxColumneventIDDataGridViewTextBoxColu mn;

private System.Windows.Forms.DataGridViewTextBoxColumneventNameDataGridViewTextBoxC olumn;

private System.Windows.Forms.DataGridViewTextBoxColumneventDateDataGridViewTextBoxCol umn;

private System.Windows.Forms.DataGridViewTextBoxColumneventVenueDataGridViewTextBoxC olumn;

private System.Windows.Forms.DataGridViewTextBoxColumnpriceDataGridViewTextBoxColumn;

}

}

**Student participation.cs**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Windows.Forms; using System.Data.SqlClient; using System.IO;

namespace intercollege\_management\_system

{

public partial class Student\_Particaption : Form

{

public Student\_Particaption()

{

InitializeComponent();

}

public string connstring = "Data Source=DESKTOP- CN9M6DO;InitialCatalog=intercollege;Integrated Security=True";

SqlCommandcmd;

public void display\_data()

{

SqlConnection conn = new SqlConnection(connstring); conn.Open();

SqlCommandcmd = conn.CreateCommand(); cmd.CommandType = CommandType.Text; cmd.CommandText = "select \* from [studentparticipation]"; cmd.ExecuteNonQuery();

DataTabledta = new DataTable(); SqlDataAdaptersda = new SqlDataAdapter(cmd); sda.Fill(dta);

dataGridView1.DataSource = dta; conn.Close(); }

private void Student\_Particaption\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'intercollegeDataSet4.studentparticipation' table. You can move, or remove it, as needed. this.studentparticipationTableAdapter.Fill(this.intercollegeDataSet4.studentparticipation);

}

//add btn

private void button1\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection(connstring); conn.Open();

string sqlQuery = "insert into studentparticipation(Student\_Name,Student\_College\_Name,Event\_ID,Event\_Name,Event\_D ate,Event\_Price,Event\_Venue)values('"+txt2.Text+"','"+textBox1.Text+"','"+textBox2.Text+" ','"+textBox3.Text+"','"+dateTimePicker1.Text+"','"+textBox5.Text+"','"+textBox4.Text+"')"; cmd = new SqlCommand(sqlQuery, conn);

int N = cmd.ExecuteNonQuery(); conn.Close();

MessageBox.Show(N.ToString() + "Data saved Successfully !!");

display\_data();

}

//dltebtn

private void button3\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection(connstring); conn.Open();

SqlCommandcmd = conn.CreateCommand(); cmd.CommandType = CommandType.Text;

cmd.CommandText = "delete from [studentparticipation]where Student\_Name='" + txt2.Text

+ "'";

cmd.ExecuteNonQuery(); conn.Close(); display\_data();

}

//clear btn

private void button4\_Click(object sender, EventArgs e)

{

txt2.Clear(); textBox1.Clear(); textBox2.Clear(); textBox3.Clear(); textBox5.Clear(); textBox4.Clear();

}

//back btn

private void button6\_Click(object sender, EventArgs e)

{

this.Close();

Dashboard d = new Dashboard(); d.Show();

}

//display btn

private void button2\_Click(object sender, EventArgs e)

{

display\_data();

}

//next btn

private void button5\_Click(object sender, EventArgs e)

{

}

} }

**Events.cs**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Windows.Forms;

namespace intercollege\_management\_system

{

public partial class Events : Form

{

public Events()

{

InitializeComponent();

}

}

}

**Bill.cs**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Windows.Forms; using System.Data.SqlClient; using System.IO;

namespace intercollege\_management\_system

{ public partial class Bill : Form

{

public Bill()

{

InitializeComponent();

}

public string connstring = "Data Source=DESKTOP- CN9M6DO;InitialCatalog=intercollege;Integrated Security=True";

SqlCommandcmd;

public void display\_data()

{

SqlConnection conn = new SqlConnection(connstring);

conn.Open();

SqlCommandcmd = conn.CreateCommand(); cmd.CommandType = CommandType.Text; cmd.CommandText = "select \* from [bill]"; cmd.ExecuteNonQuery();

DataTabledta = new DataTable(); SqlDataAdaptersda = new SqlDataAdapter(cmd);

sda.Fill(dta);

dataGridView1.DataSource = dta; conn.Close(); }

//save btn

private void button1\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection(connstring); conn.Open();

SqlCommandcmd = conn.CreateCommand(); cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into bill(Sub\_Total,Discount,Grand\_Total,Paid\_Amount,Return\_Amount)values('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox6.Text + "','" + textBox5.Text + "','" + textBox4.Text + "')";

cmd.ExecuteNonQuery(); conn.Close();

MessageBox.Show("data saved successfully"); display\_data(); }

//back btn

private void button6\_Click(object sender, EventArgs e)

{

this.Hide();

Dashboard d = new Dashboard(); d.Show();

}

private void Bill\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'intercollegeDataSet.bill' table. You can move, or remove it, as needed.

this.billTableAdapter.Fill(this.intercollegeDataSet.bill); } private void button2\_Click(object sender, EventArgs e)

{

display\_data();}}}

**About\_Us.cs**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Windows.Forms;

namespace intercollege\_management\_system

{

public partial class About\_Us : Form

{

public About\_Us()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Dashboard d = new Dashboard(); d.Show();

}

}

}

**SMS\_System.cs**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms; using System.Net;

using System.Collections.Specialized; using System.IO;

namespace intercollege\_management\_system

{

public partial class SMS\_System : Form

{

public SMS\_System()

{

InitializeComponent();

}

private void btnSend\_Click(object sender, EventArgs e)

{

{

String result;

string apiKey = txtapi.Text;

string numbers = txtnum.Text; // in a comma seperated list string message = txtmess.Text;

string send = btnSend.Text;

String url = "https://api.textlocal.in/send/?apikey=" + apiKey + "&numbers=" + numbers + "&message=" + message + "&sender=" + send;

//refer to parameters to complete correct url string StreamWriter myWriter = null;

HttpWebRequest objRequest = (HttpWebRequest)WebRequest.Create(url); objRequest.Method = "POST";

objRequest.ContentLength = Encoding.UTF8.GetByteCount(url); objRequest.ContentType = "application/x-www-form-urlencoded"; try

{

myWriter = new StreamWriter(objRequest.GetRequestStream()); myWriter.Write(url);

}

catch (Exception ex)

{

//return e.Message;

MessageBox.Show(null, "the error is" + ex, MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

myWriter.Close();

}

HttpWebResponse objResponse = (HttpWebResponse)objRequest.GetResponse();

using (StreamReader sr = new StreamReader(objResponse.GetResponseStream()))

{

result = sr.ReadToEnd();

// Close and clean up the StreamReader sr.Close();

}

//return result; MessageBox.Show(result);

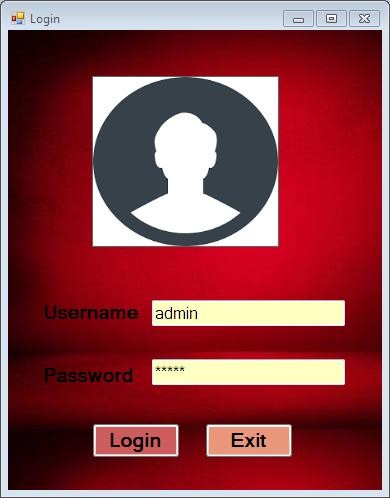
}

} }

}

**SCREENSHOTS OF DESIGN:**

#### Login:



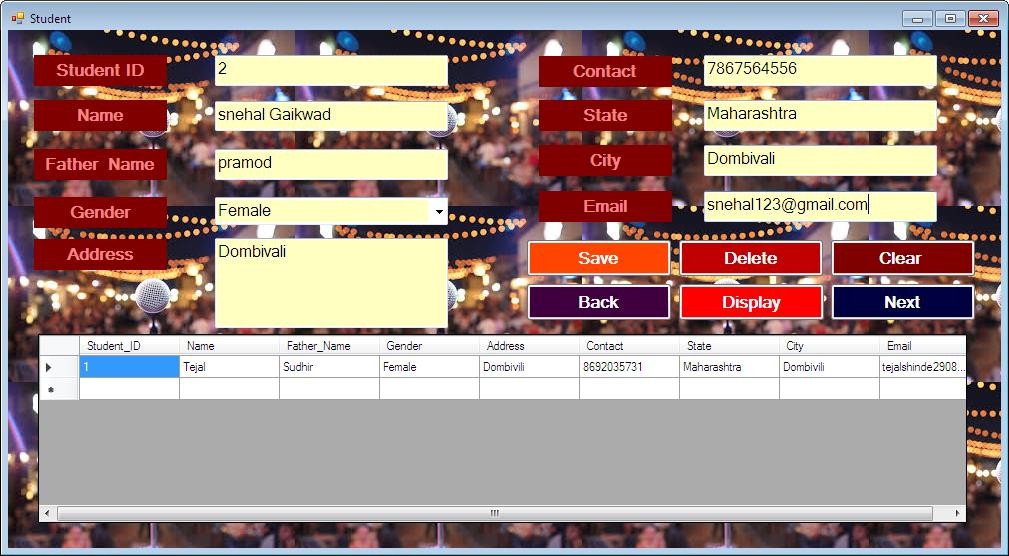
**Dashboard:**



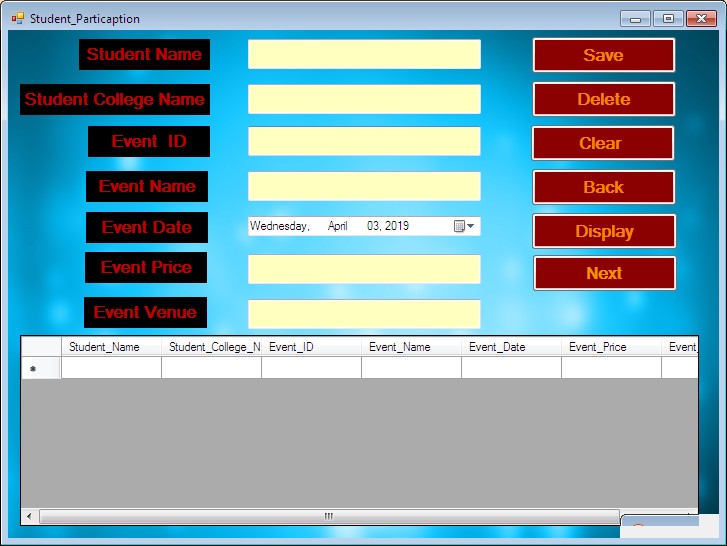
**About\_Us:**



**Student Registration:**



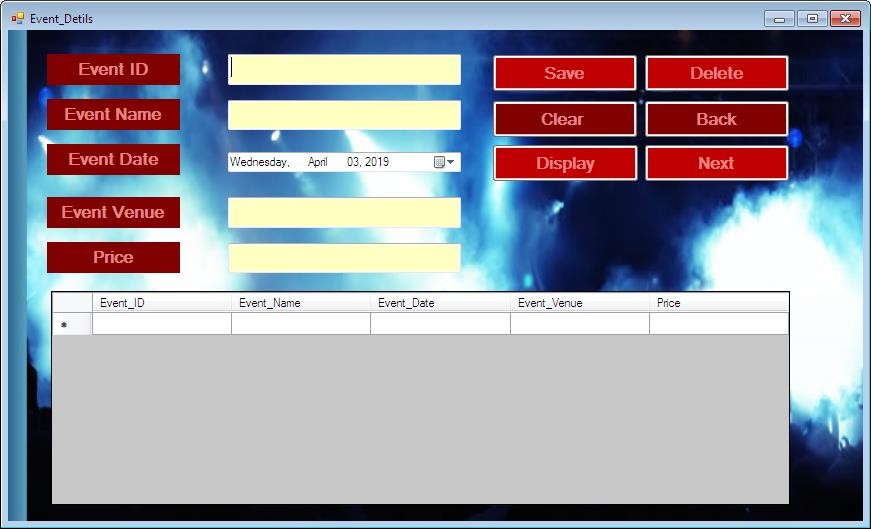
**Student Participation:**



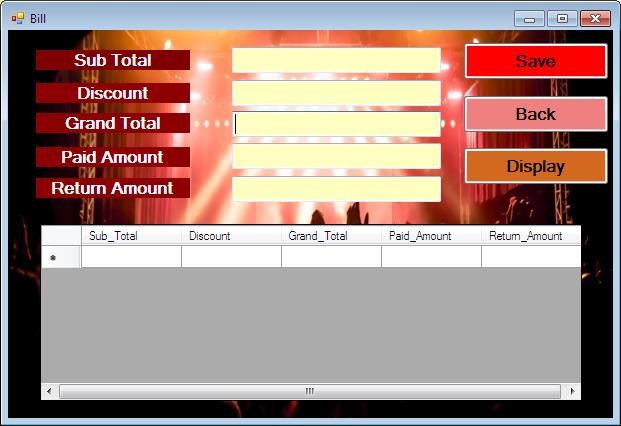
**Events:**



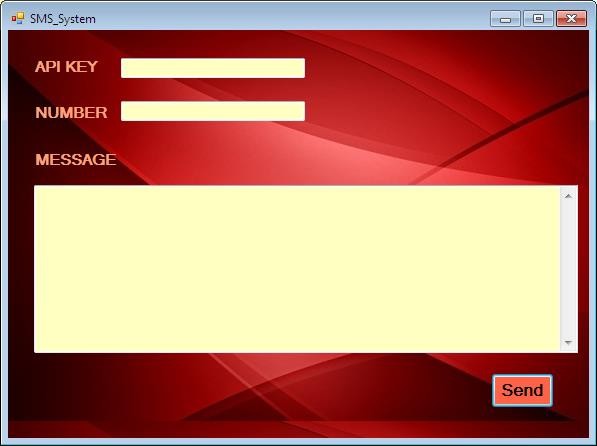
**Events\_Details:**



**Bill:**



**SMS\_System:**



* 1. **TESTING APPROACHES**

**--White-Box Testing:**

In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

Testing based on an analysis of the internal structure of the component or system. Procedure to derive and select test cases based on an analysis of the internal structure of a component or system.

--**Black-Box Testing:**

Black box testing has little or no regard to the internal logical structure of the system, it only examines the fundamental aspect of the system. It makes sure that Input is properly accepted and output is correctly produced.

Black box testing is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software.

This type of testing is based entirely on software requirements and specifications.

.

**--Functional Testing:**

Functional tests involve exercising the code with nominal input values which gives The expected results and boundary values are known.

Functional Testing is defined as a type of testing which verifies that each **function** of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing and it is not concerned about the source code of the application.

Each and every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results.

**--Performance Testing:**

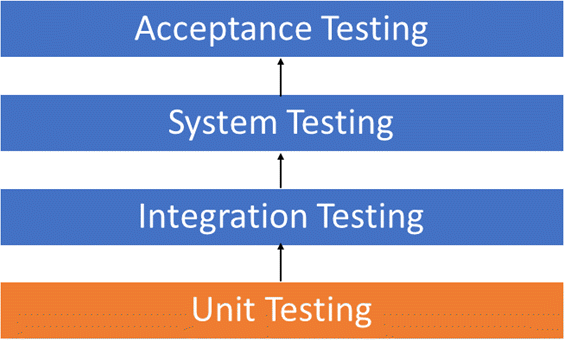
Performance tests are designed to verify response time. If the wrong data is entered then the system does not allow it and calculations are not performed.

Performance Testing is defined as a type of software testing to ensure software applications will perform well under their expected workload.

Features and Functionality supported by a software system is not the only concern. A software application's performance like its response time, reliability, resource usage and scalability do matter. The goal of PerformanceTesting is not to find bugs but to eliminate performance bottlenecks.

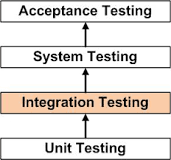
* + 1. **Unit Testing**

**Unit Testing** is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.



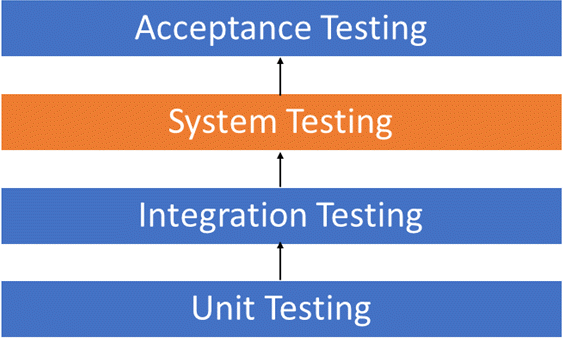
### Integration Testing

**Integration Testing** is a level of software testingwhere individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in **Integration Testing**.



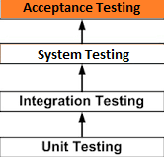
### System Testing

**System Testing** is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.



### Acceptance Testing

**Acceptance Testing** is a level of software testingwhere a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.



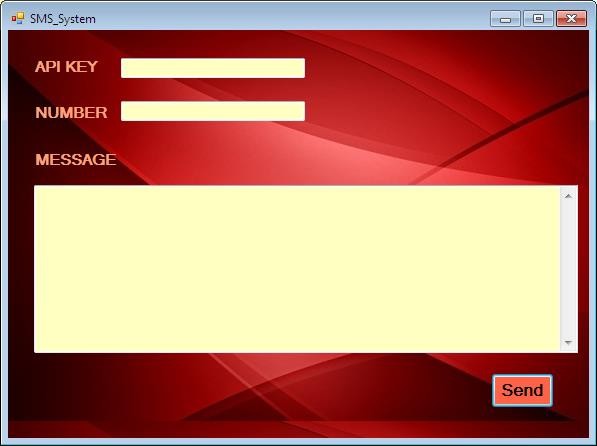
#### MODIFICATION & IMPROVEMENTS



* + - Here, you can see we have added some new features i.e. Bill of the respective events will be generated as well as notifications will be send to the students after their registration.

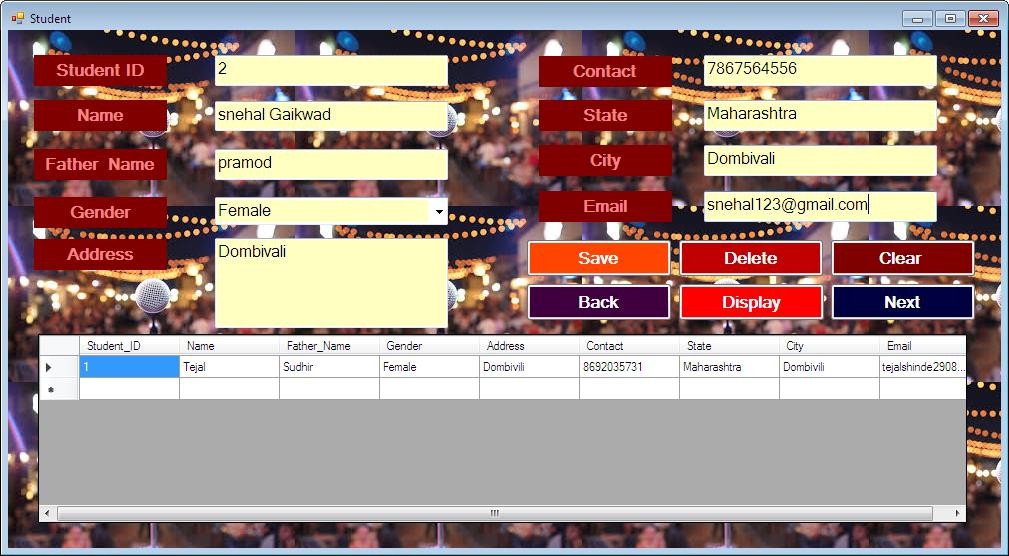


* + - We have added SMS\_System so that notifications will be send to the students and if in case their is change in schedule of the events then it will be notified to the student through this SMS\_System.

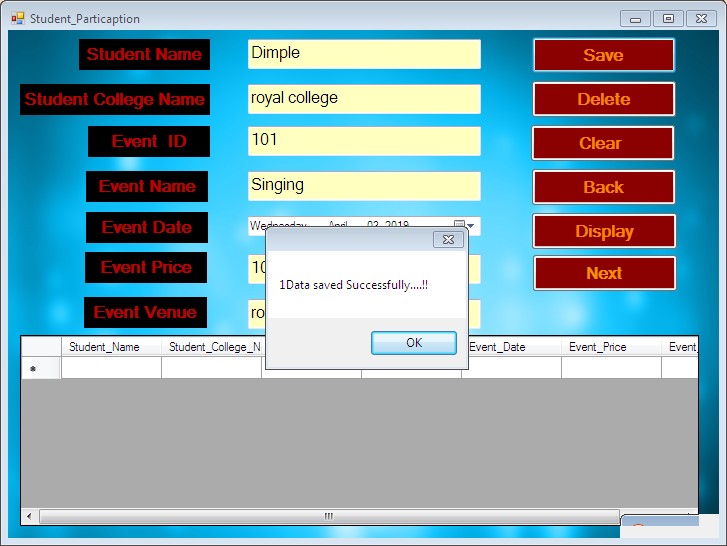


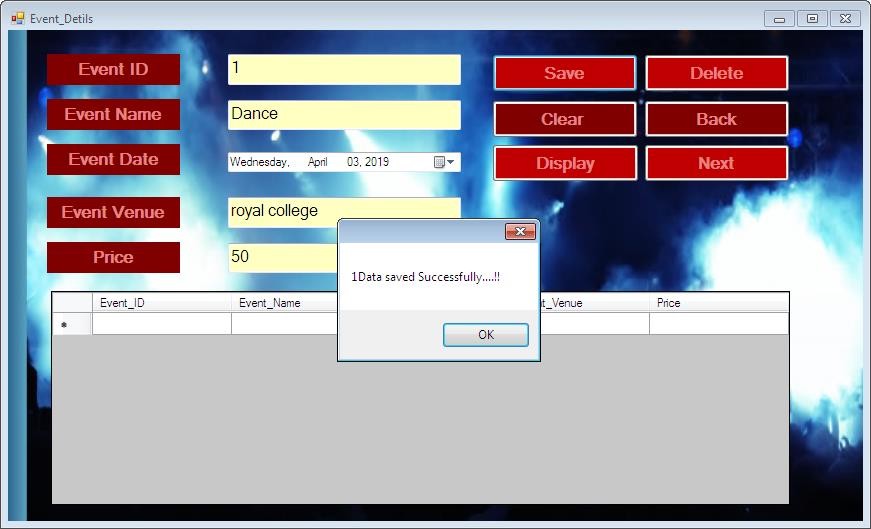
#### TEST CASES

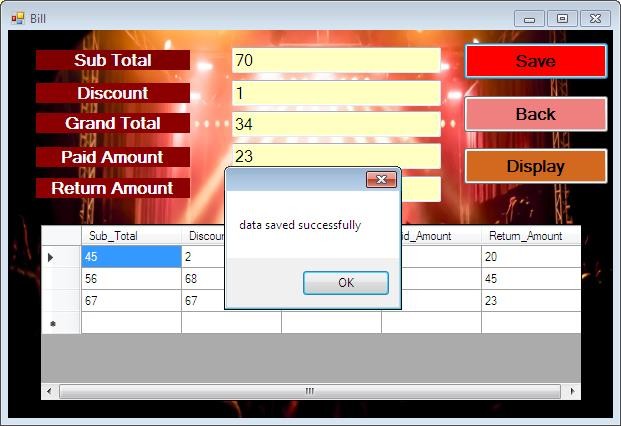
Only Admin is allowed to handle/manage this system, if in case apart from Admin,others tried to login the system then it will generate a message **“Please check username and Password” .**



\







### Manual Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Name** | **Input** | **Expected Output** | **Actual Output** | **Result** |
| 1. | Login Page | Valid Credentials | Navigates to  requested pages | Navigates to  requested pages | Pass |
| 2. | Login Page | Invalid Credentials | Showing invalid Password for wrong user | Showing invalid Password for wrong user | Pass |
| 3. | Student reg | Valid Credentials | Register students | Register Students | Pass |
| 4. | Student reg | Invalid  Credentials | Display Error  Message | No Error  message | Fail |
| 5. | Student\_Participation | Valid Credentials | Participation Successful | Participation Successful | Pass |
| 6. | Student\_Participation | Invalid  Credentials | Student NOT  Registered | Student NOT  Registered | Pass |
| 7. | Events | Valid Credentials | Event Types | Event Types | Pass |
| 8. | Events | Invalid  Credentials | --- | --- |  |
| 9. | Event\_Details | Valid Credentials | Displays Event Details | Displays Event Details | Pass |
| 10. | Event\_Details | Invalid Credentials | Do not  Display Event Details Properly | Display Event Details | Fail |
| 11. | Bills | Valid Credentials | Correct Bills | Correct Bills | Pass |
| 12. | Bills | Invalid  Credentials | Incorrect Bill | Incorrect Bills | Pass |
| 13. | Reports | Valid Credentials | Generates accurate Reports | Generates accurate Reports | Pass |
| 14. | Reports | Invalid Credentials | Do not  Generate accurate  Reports | Do not  Generate accurate  Reports | Pass |
| 15. | SMS\_System | Valid Credentials | Students will Receive SMS | Students will Receive SMS | Pass |
| 16. | SMS\_System | Invalid Credentials | SMS should not be send | SMS delivers to Students | Fail |

**CHAPTER 6**

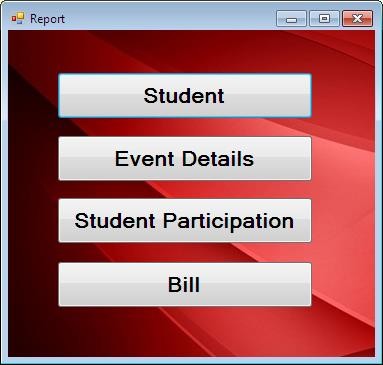
#### RESULTS & DISCUSSIONS

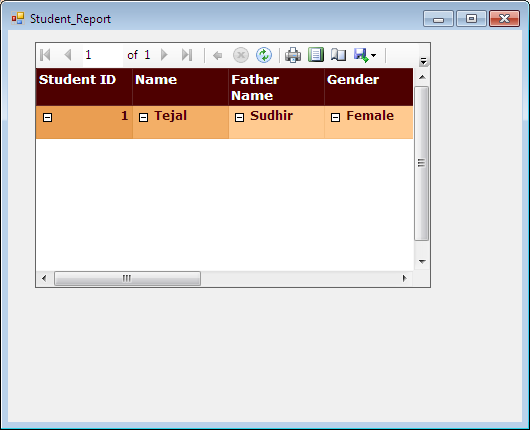
#### TEST REPORTS

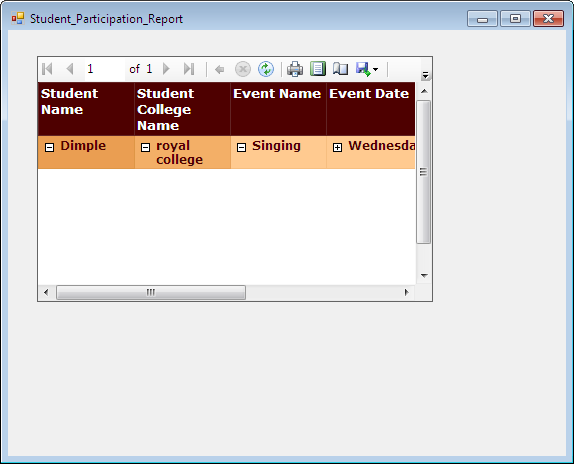
#### USER DOCUMENTATION

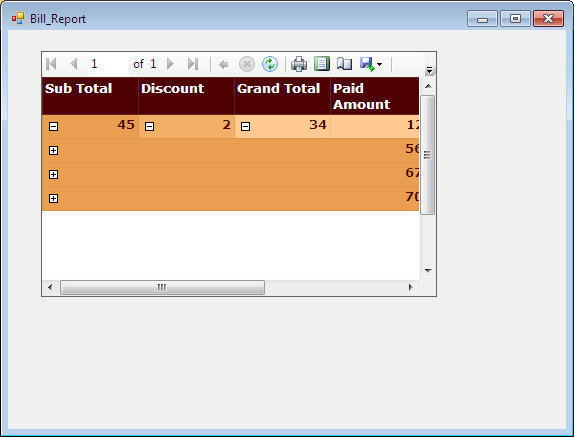
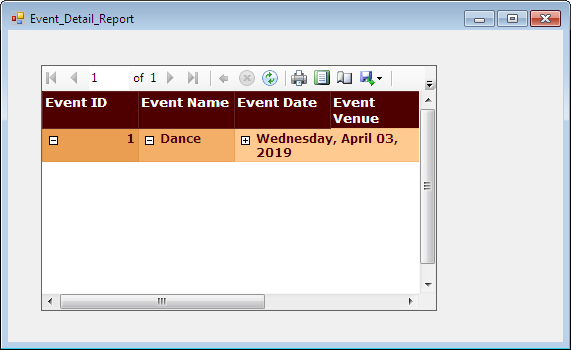
* 1. **TEST REPORTS**

**CHAPTER 6**





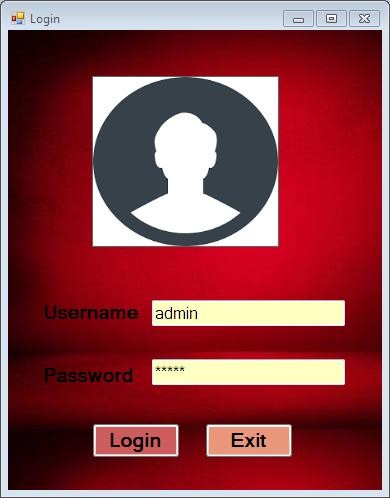




* 1. **USER DOCUMENTATION**

**Step 1:**

This system can be managed by on admin. Admin is only allowed to insert, update or delete the data from the system. Once, the application get started Admin have to Login with appropriate Username and Password. And if, Username and Password is incorrect then admin will not be allowed to control/operate the system.



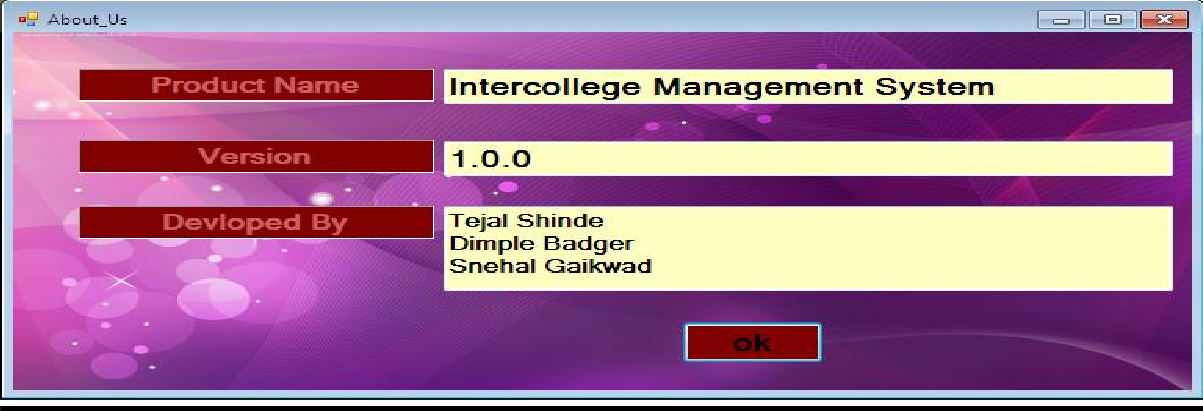
### Step 2:

Once, the Admin entered the correct details then Dashboard i.e the main page of the system will appear. There you will find different options.



### Step 3:

Then, if you clicked on About\_US there you will get to know about the system like the name of the product, its version and by whom it is been developed.



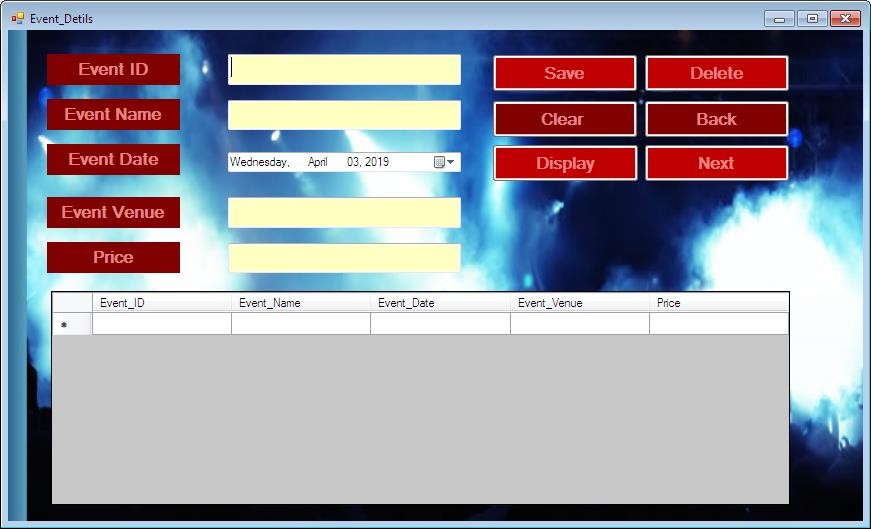
### Step 4:

Then, if you clickedon Events then all the Intracollegiate and Intercollegiate events will be appear.



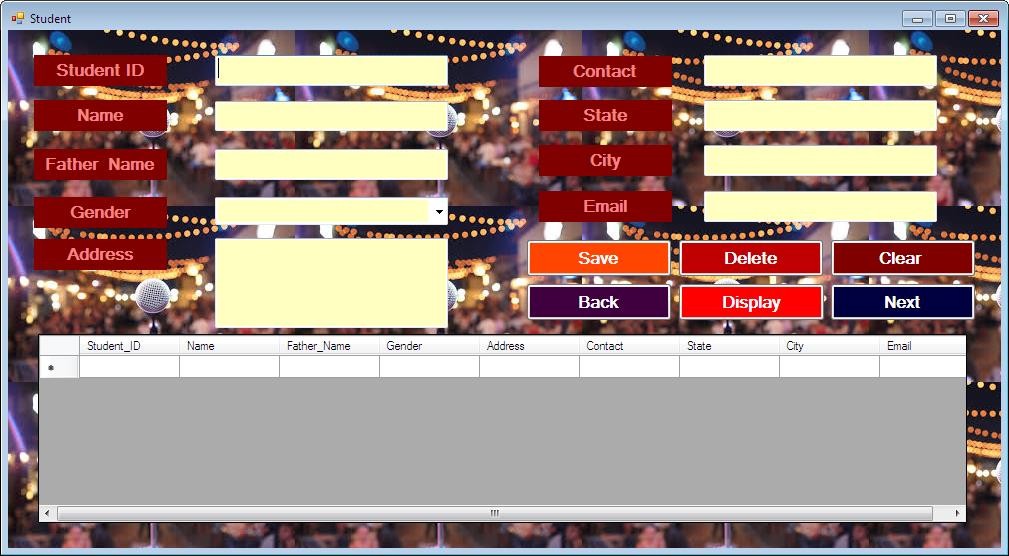
### Step 5:

Then, In Event\_Details ,details of the event will be displayed.



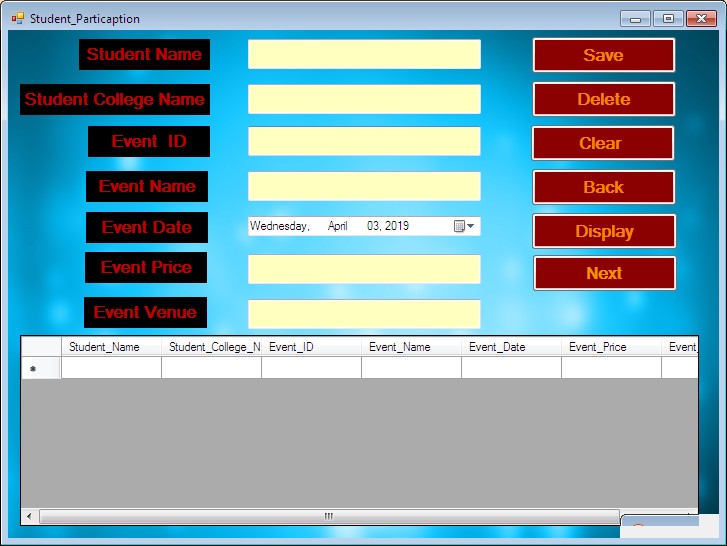
### Step 6:

After this ,Admin have to register the Students and their details will be stored in the database.

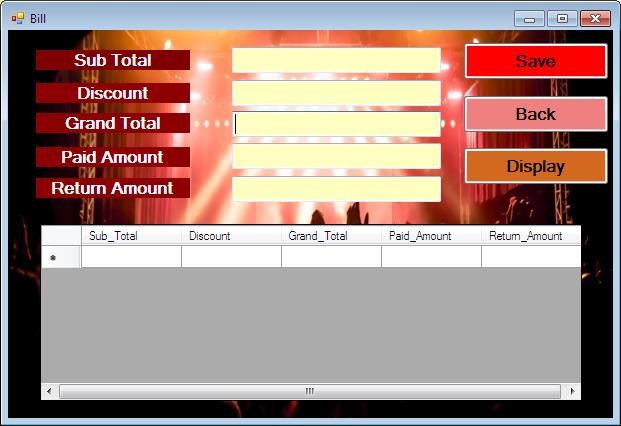


### Step 7:

Then Student\_Participation page will appear here students who want to participate in the respective events can Participate after filling this form.



### Step 8:



**Step 9:**



**CHAPTER 7**

#### CONCLUSION

#### 7.1.1 SIGNIFICANCE OF THE SYSTEM

#### LIMITATIONS OF THE SYSTEM

#### FUTURE SCOPE OF THE SYSTEM

**CHAPTER 7**

1. **CONCLUSION**

**Intercollegiate Event management System** is the application of project **management** to the creation and development of large scale **events** such as festivals, conferences, ceremonies, formal parties, concerts, or conventions**”**.

**Intercollegiate Event Management System** is user friendly and cost effective system, it is customized with activities are related to event management . IEMS System always keep your objectives and goals on top priority while developing any plan of work.

Different teams are needed to hire to manage an event but still there are some problems in sending invitations and record maintenance. These all things but time consuming. We have developed a solution for everyone’s problems and name it as **Intercollegiate Event Management System**. IEMS is a system that will reduce your work burden along with your budget. . It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future.

### SIGNIFICANCE OF THE SYSTEM

First and most important thing is **Reduce Budget**. Seriously it will reduce your budget the question is how? So answer is in your hands, it will manage all your event **registration**, **ticketing**, **marketing** etc. and you don’t have to hire extra manpower for this purpose.

It sends automatic **email notifications** to the colleges and make them feel that event manager cares for them and their presence is really important. Once someone register for the event, **Intercollegiate Event Management System** will start its work to share brief details and will also offer them support to make their experience excellent.

It can **Customized Notifications** according to their desire and make them able to feel that we respect their desire. All data and reports updated in IEMS could be exported in your desired format with **Reports feature. Intercollegiate Event Management System** store all records of your events in a single Database System, so it becomes very easy to search any record if needed.

### LIMITATIONS OF SYSTEM

Although this project has a lot of advantage and will make a lot of things easier, it does have few limitations as well, which are:

* + - The payment process cant be shifted online because some event organizers receive only on-the-spot cash payment and some users might not be familiar with online payment.
    - Complete details about events in which teams from different universities participate may not be provided, especially if the event takes place in a different university as the organizers there may have a different way of maintening event details.

### FUTURE SCOPE OF THE PROJECT

* + - The project will consist of 3 parts, Admin, Teacher and User(student). As discussed before, the organizer will add a particular event, along with the teachers or organizers who will conduct the event and will also provide complete details and schedule for that event.
    - This Project will bring the entire manual process of event management (sports or other events) online.Teachers or event organizers will inform admin about a particular event and the admin will then add that particular event along with its complete schedule and details and then willing students can register in events in which they are willing to participate. The students will be notified about a new event when its added or they can also search for events they are interested in by searching the event type (sports,cultural etc.) and then list of upcoming events of that type will be displayed.
    - Registration for different events can be done online.
    - The organizer will have the list of participants saved in a database.
    - Register users will be notified about an upcoming event.
    - The complete information regarding a particular event will be available online.
    - In case of competition based events, the results will be calculated and will be available online.
    - The project will consist of 3 parts, Admin, Teacher and User(student). As discussed before, the organizer will add a particular event, along with the teachers or organizers who will conduct the event and will also provide complete details and schedule for that event.

### REFERENCES

* **FOR .NET INSTALLATION**

[www.support.mircosoft.com](http://www.support.mircosoft.com/)

* **FOR DEPLOYMENT AND PACKING ON SERVER**

[www.developer.com](http://www.developer.com/)

[www.15seconds.com](http://www.15seconds.com/)

* **FOR SQL**

<https://www.youtube.com/watch?v=xybe0fpj_10>

* **FOR ASP.NET**

[www.msdn.microsoft.com/net/quickstart/aspplus/default.com](http://www.msdn.microsoft.com/net/quickstart/aspplus/default.com)

[www.asp.net](http://www.asp.net/)

[www.fmexpense.com/quickstart/aspplus/default.com](http://www.fmexpense.com/quickstart/aspplus/default.com)

[www.asptoday.com](http://www.asptoday.com/)

[www.aspfree.com](http://www.aspfree.com/)

[www.4guysfromrolla.com/index.aspx](http://www.4guysfromrolla.com/index.aspx)

### GLOSSARY

##### **IEMS** : Intercollegiate Event Management System

**DB** : Database

**UT :** Unit Testing

##### **MT** : Manual Testing

**RAD** : Rapid Application Development

**Login:** Fully Secured. Only Admin is allowed to access the system.

**Events:** Events like Intracollegiate as well as Intercollegiate events are

Available.

**Event\_Details:** Details of every events will be given here.

##### **Student :** If Student want to participate then he/she must register themselves

First.

**Student\_Participation**: Only registered students will be allowed to participate

##### in the respective events

**Reports:** Reports will be generated as per the student participation.

**SMS\_System :** It will send the notifications to the participated students about

The events.