TABLE OF CONTENTS

1. Abstract…………………………………………………………………………………………………………….2

2. Project Overview………………………………………………………………………………………….……3

2.1 Purpose, Features and Objectives…………………………………………………...............3

2.1.1 Purpose of the System…………………………………………………………………..3

2.1.2 Features……………………………………………………………………………………….4

2.1.3 Scope…………………………………………………………………………………………..5

2.2 Key Roles………………………………………………………………………………………………5

2.3 Assumptions and Constraints………………………………………………………………….5

2.3.1 Assumptions………………………………………………………………………………..5

2.3.2 Constraints………………………………………………………………………………….5

2.4 Modules………………………………………………………………………………………………...6

2.5 Webpage Design…………………………………………………………………………………….7

3. Diagrams……………………………………………………………………………………………………………8

3.1 ER Diagram……………………………………………………………………………………………..8

3.2 Use Case Diagram……………………………………………………………………………………9

3.3 Class Diagram…………………………………………………………………………………………10

1. ABSTRACT:

Institute Management System (IMS) is capable of managing each and every data regarding course enquiry, admission, course selection, fees payment and course completion. IMS helps in managing institute in an extremely efficient way. Entry of data into the system starts with a student enquiry about a course. On enrollment to a course, the enquiries are converted into admission. Fee and course details are maintained for the enrolled students.

The institute has many courses being offered. Students enquire about the course and if they decide to opt for the course, they will get enrolled to the course. Depending on the strength of the course various batches are generated. The same course will be started once in every fortnight.

The institute has permanent trainers (Full time trainers -FTT) and depending on the requirement and course generation demand trainers will be hired from outside on contract and are mentioned as Vendor trainers.

The system will help to generate various reports for

1. Identifying the enquiries that are not converted to admissions
2. Enquiry admission conversion rates
3. Course not opted by students
4. Course opted mostly by the students
5. List of fee out standings

This project will help to monitor the performance of the institute in an efficient manner. The system is user friendly and maintains data of current and previous years. The system can further be developed for data mining purpose also.

1. Project Overview

The project has users such as admin, accounts and students.

*Admin* can perform CRUD operations to maintains course enquiry, admission and fee details of students and generates reports and export the details to excel.

*Accounts* user will have access to enrolled students and fee details.

*Students* role can edit their personal details and can view course and fee details.

Various modules of the system are given in figure 1.1. When a student wishes to enquire about a new course, he/she has to register then the student will be given by one individual username and password. On successful authentication, the users will enter into another page. In that page student can select two options that are update details and view details.

Figure 1.1: Modules of Institute Management System

Institute Management System

Course Enquiry

Course Admission

Report Generations

Fee Management

2.1 System Requirements

* **Hardware Requirements**

System Type : 32-bit/64-bit

RAM : 4GB

* **Software Requirement**

Operating System : Windows 8 & Higher

Back End : SQL Server 2014 Database

Front End : HTML, CSS, JavaScript

Architecture : ASP.NET

Environment : CLR

Platform : Visual Studio

Server : IIS

* 1. Project modules

1. Login Module

User login and authentication

1. Admin Module

CRUD on course, admission, enquiry and fee details

1. Accounts Module

Read and update operations allowed on fee details and automatic updation on fee payment to student details.

1. Student Module

Update option on personal details

1. Reports Module

Report generation from operations of various modules. Exporting of report in Excel format also.

* 1. Webpage Design

Home page

Login

Course listing

About us

Contact us

Testimonials

Login

User wise login (Admin / Accounts / Students

Course Enquiry Page

Student enrollment Page

Reports Generation Page

Options will vary depends on the user logged in

1. Table design and description

**Enquiry Form Table**

**Column Name Data type Constraints**

Enquiry\_no integer Primary key

Enquiry\_Name varchar Not Null

Contact\_No varchar Not null

Course\_choice 1

Course\_choice2

Course\_choice3 varchar Not null

Preference\_date date Check(pref date>=current date)

**Admission Form Table**

**Field Name Data Type Constraint Link**

Admin\_no Varchar(20) Primary key

Stu\_name Varchar(20) Not null

Stu\_add1 Varchar(20) Not null

Stu\_add1 Varchar(20) Not null

Stu\_locid Varchar(20) Identity Location table

Stu\_contact Varchar(20) Not null

Stu\_mail Varchar(20) Not null

Stu\_qual Varchar(20) Not null

Stu\_gender Char check

Stu\_dob Varchar(20) check

Stu\_course Varchar(20) Not null C.code of Course table

Stu\_doj Varchar(20) Check

Stu\_batchid Varchar(20) Not null Batch table

Stu\_certno Varchar(20) Not null

**Location Table**

**COLUMN NAME CONSTRAINT DATA TYPE**

LocID Primary Key Identity (Linked to Stu\_locid & Trainer\_locid)

City Unique varchar(20)

State Unique varchar(20) Table Level Unique

Country Unique varchar(20)

ZipCode Unique varchar(20)

**Course Table**

**FIELD NAME Data Type Constraints**

Course\_ code Identity Primary Key

Course\_ ID varchar

Course\_ Name Varchar Not Null

Duration int Check ( > 0)

Fees Decimal Check ( > 0)

Course \_Abbr varchar Not null

**Batch Table**

**Column Name Constraints Data Type**

Batch\_id Primary Key Integer

Course\_code identity

Start\_date check Date

End\_date check Date

Student\_count check identity

Trainer\_id Not null

**Trainer Table**

**COLUMN NAME CONSTRAINTS DATATYPE**

TrainerID Primarykey Varchar(20)

TrainerName NOT NULL Varchar(30)

LocationID Identity Varchar(30)

DOJ Check Varchar(30)

Type(FTT or Vendor) Check Varchar(30)

**Trainer Specialization**

**Column Name Constraint Datatype**

TrainerID PrimaryKey Varchar(30)

CCode Identity Varchar(30)

**Fees Details**

create table FeeHeader

(

FeeId int identity(1,1) primary key,

AdmNo int ,

AdminDate date,

FeeHeadAmt decimal(7,2),check(FeeHeadAmt>0),

);

create table FeeDetail

(

FeeDetailid int identity(1,1) primary key,

FeeId int foreign key references FeeHeader(FeeId),

FeeDate date,

FeeAmt decimal(7,2),check(FeeAmt>0),

);