**TUITION MANAGEMENT SYSTEM**

A project submitted to

**UKA TARSADIA UNIVERSITY**

In partial fulfillment of the requirements for the degree of

**Bachelor of Science**

in

**Information Technology**

for

**5 Years Integrated M.Sc.(IT)**

By

**SOURABH SINGH (201806100110069)**

**VISHWA CHATURVEDI (201806100110127)**



**Babu Madhav Institute of Information Technology**

**Uka Tarsadia University**

**Bardoli – 394350**

****

**CERTIFICATE**

This is to certify that **SOURABH SINGH (201806100110069)** **VISHWA CHATURVEDI (201806100110127)** andhas submitted project entitled **“TUITION MANAGEMENT SYSTEM”** as the partial fulfillment for the award of the degree of Bachelor of Science in Information Technology for 5 Years Integrated M.Sc.(IT) in 2021 – 2022.

**Date:**

**Place:**

**Dr. Jitendra Nasriwala**

Guide Programme Coordinator,

**External Examiner**



**Babu Madhav Institute of Information Technology,**

**Uka Tarsadia University,**

**Bardoli – 394350**

**AKNOWLEDGEMENT**

I WOULD LIKE TO EXTEND OUR GRATITUDE TO OUR HEAD OF DEPARTMENT **“MR. JITENDRA NASRIWALA” SIR** FOR PROVIDING US WITH ALL THE FACILITY THAT WAS REQUIRED.

**Table of Content**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chapters** | **Particulars** | | | **Page no.** |
| **1** | **Introduction** | | |  |
| 1.1 | Problem Definition | |  |
| 1.2 | Initial Requirement Document | |  |
| 1.3 | Project Objective | |  |
| 1.4 | Product Scope | |  |
|  | | | | |
| **2** | **Overall Description** | | |  |
| 2.1 | Product Perspective/ Environment Description | |  |
|  | 2.1.1 | Hardware Interface/ Hardware Specification |  |
|  | 2.1.2 | Software Interface/ Software Specification |  |
|  | | | | |
| **3** | **System Planning** | | |  |
| 3.1 | Software Engineering Model | |  |
|  | 3.2 | CPM PERT | |  |
|  | | | | |
| **4** | **System Specific Requirements** | | |  |
| 4.1 | Functional Requirement | |  |
| 4.2 | Non-functional Requirement | |  |
|  | | | | |
| **5** | **System Analysis** | | |  |
| 5.1 | Use case Diagram | |  |
| 5.2 | Activity Diagrams | |  |
|  | | | | |
| **6** | **System Design** | | |  |
| 6.1 | Database Design | |  |
| 6.1.1 | Database Schema |  |
| 6.1.2 | Data Dictionary |  |
|  | | | | |
| **7** | **System Implementation** | | |  |
| 7.1 | Screenshots | |  |
|  | | | | |
| **8** | **Testing** | | |  |
| 8.1 | Test Cases | |  |
|  | | | | |
| **9** | **Future Enhancement** | | |  |
|  |  | | |  |
|  | **Conclusion** | | |  |
|  |  | | |  |
|  | **Bibliography** | | |  |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Figure Description** | **Page no.** |
| **5.1.1** | **Use Case Diagram for Admin** |  |
| **5.1.2** |  |  |
| **5.1.3** |  |  |
| **5.2.1** | **Activity Diagram for Manage Product** |  |
| **5.2.2** |  |  |
| **5.2.3** |  |  |
| **5.2.4** |  |  |
| **5.2.5** |  |  |
| **5.3.1** | **Sequence Diagram for “adding a product”** |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Table Description** | **Page no.** |
|  | Data Dictionary Tables | 50 - 60 |
| 8.1.1 | Test Cases for Login |  |
| 8.1.2 | Test Cases for Product |  |

**CHAPTER 1: INTRODUCTION**

This document is intended for different type of readers such as institute owner, the system designer, the system developer and the tester. By reading this document a reader can learn about what the project is implemented for and how it will present its basic idea. This document has a sequential overview of the whole project so if a reader reads the document from top to bottom, he/she will get a clear idea about the project.

* 1. **Problem Definition:**

Educational institutions are burdened by cumbersome paperwork and manual processes, and they find it difficult to maintain records on attendance, fees, admissions, etc., and track the information they need. Even though as we consider the current scenario of pandemic where everything is switching online, and when it comes to the extra tutoring it is nearly negligible that any pupil would go for it by risking the health of individual and of the family.

With the advent in technology and with the perpetual increase in pandemic, we propose a “TUTION MANAGEMENT SYSTEM”. This will eliminate all the traditional (manual) way of teaching and provide ease in gaining education, exchanging materials, solving their doubts and testing their knowledge via testes. It takes the work to digital platform where it provides an automation procedure of studying, improve the learning process for students and deliver all the required and relevant material at their screens at ease and quick. From Administration point of view, the procedure of collecting the fees hand-to-hand can be eliminated, as the pupils can pay their tuition fees online.

**1.2 Initial Requirement Document**

|  |  |
| --- | --- |
| Title of the project | TUITION MANAGEMENT SYSTEM |
| Stakeholders involved in capturing requirement | Sourabh Singh, Vishwa Chaturvedi |
| Techniques used for requirement capturing | Discussion, Brainstorming |
| Name of the person along with designation | - |
| Date | 16/01/2022 |
| Users of the system | Student, Administration Staff, Teachers |
| Version | 1.0 |
| **Consolidated list of initial requirements:**   1. The system is to be implemented which can manage all the tuition management task. 2. There are three users in this system namely: a) Administrator b)Faculty c)Student 3. The system shall be able to access through username and password for the users. 4. The system shall be able to display availability of all the subjects to be taught grade(standard )wise. 5. The system shall be able to view faculty details and subscription details to admin. 6. The system shall be providing some special offers to the user on basis of marks and performance (marks 🡪Student, performance 🡪faculty). 7. The system shall be able to notify student for the upcoming class. 8. The system shall be able to accept online fees. 9. The system shall be able to generate student progress report. 10. The system shall be able to generate faculty report 11. The faculty shall be able to take attendance of students through system. 12. The faculty shall be able to view time table schedule. 13. The faculty shall be able to view class attendance. 14. The faculty shall be able to view student progress report. 15. The admin shall be able to maintain all the details of subjects, subscription, fees, special offer, and schedule. 16. The admin shall be able to view all the report generated by the system. 17. The admin shall be able to view payment details. 18. System shall be developed in such a way that it allows the system admin to create the batch of the students, and allocate the schedule to the teachers and students. 19. System shall be developed in such a way that it allows the students to apply for the leave. 20. System shall be developed in such a way that the admin can schedule the total number of tests which is to be conducted in a particular time. 21. System shall be developed in such a way that the teachers can schedule the upcoming test. 22. System shall be developed in such a way that the teachers and student can discuss their doubts in group chat for that respective subject. 23. System shall be developed in such a way that the provide the service of paying the tuition fees online, by proving the time interval in which the fees must be deducted. 24. System shall be developed in such a way that the students can passively ask their doubts regarding individual subjects via a virtual group chat. | |

**Abbreviations:**

* **TMS-** Tuition management system

**1.3 PROJECT OBJECTIVE:**

The main objective of TMS is to help students get over the traditional methods of learning and make them accustomed to the internet and e-learning where they will be provided with all the study materials for their respective subject and also that they can solves their doubts just by one click.

This TMS also helps the institute manager to manage the classes more effectively and efficiently by computerizing the lectures, exams, attendance, doubt solving session, student group communication and fees process.

This system will furthermore help the manager to make appropriate educational decisions for the institute as the transactions and other information of students and staff will be stored in the system. The whole system is designed for a general computerize digital tutoring institute. So that any tutoring owner can get it and start automated process of his own institute.

**1.4 PRODUCT SCOPE:**

A tuition management system is used in building up the networking to help the service providers gain students and introduce a new education providing services. This TMS can be used by all the general public. This system is primarily structured to create a great customer base business which provides an ease in maintaining education for all/education at ease motive. TMS typically includes business-to-consumer (B2C) service. This system will help to manage and run the institute systematically.

The primary TMS areas are as follows:

* student management
* staff management
* Online exams management
* Advance notices management

**CHAPTER 2: OVERALL DESCRIPTION:**

**2.1 PRODUCT PERSPECTIVE**

The TUITION Management shall be developed using client/server architecture and compatible with Microsoft Windows Operating System .The front-end of the System will developed using REACT JS and the back-end will developed using MONGOOSE .

Back-end Database

Front-end client application (maintain stock, sales calculation)

**2.1.1 HARDWARE INTERFACES**

**2.1.2 SOFTWARE INTERFACES**

a) node js

b) vs code

c) mongoDB

d) post man

**CHAPTER 3: SYSTEM SPECIFICATION REQUIREMENTS**

**3.1 Functional Requirement**

**Module 1: Login and Registration**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| FR1 | Student can apply to register on the system by providing following details:  First Name, Middle Name, Last Name, Guardian Name, Date of Birth, Contact, Email ID, Address, Class, School, Profile Picture, Course | Student Registration Request |
| FR2 | Staff can apply to register on the system by providing following details:  First Name, Middle Name, Last Name, Date of Birth, Contact, Qualification, Email ID, Address, Profile Picture | Faculty Registration Request |
| FR3 | Admin can look at the list of students who are requesting to be registered on the system and approve or reject the request on the basis of information provided by student. | Student Registration List |
| FR4 | Admin can look at the list of faculties who are requesting to be registered on the system and can approve or reject the request on the basis of information provided by faculty. | Faculty Registration List |
| FR5 | Students and faculties whose registration has been approved by the admin can login into the system using provided email id and default password. | Login |
| FR6 | Admin have to add a remark while rejecting the registration request of any staff and student. | Remark |
| FR7 | Admin, staff member and students can recover or reset their account password if they forgot by clicking on forgot password. | Forgot Password |

**Module 2: Profile Management**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| FR1 | Faculty, Student and Admin can update the following profile information from the system:  First name, Middle name, Last name, contact, address and profile picture. | Update Profile |
| FR2 | Admin can remove any student and faculty from the system if any of the information is found suspicious and false. | Remove User |
| FR3 | Admin can view the list of all the staff member and students registered in the system. | View users |
| FR4 | Staff member can view the information of the students registered in the system. | View Student |
| FR5 | A user can reset or change their password. | Change Password |

**Module 3: Subject Management**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| FR1 | Admin can add a specific subject as per the classes by providing following details:  Subject code, Year, Subject Name, Book Name, Publisher, Subject Remark | Add subject |
| FR2 | Admin can remove a subject from the subject list. | Remove Subject |
| FR3 | Admin can allocate subjects to staff member. | Allocate subject |
| FR4 | Admin and staff member can view all the subjects. | Subject List |

**Module 4: Course Management**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| FR1 | Admin can add package including the following information about the course:  Course Name, Subjects, Course Duration, Fee | Add Course |
| FR2 | Admin can modify the course information or can add any new subject if there exist any in the system. | Modify Course |
| FR3 | Admin can remove the course from the system. | Remove Course |
| FR4 | Admin, staff member and students can view the courses. | View Course |

**Module 5: Test Management**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| FR1 | Staff member can schedule test by providing following details:  Subject Name, Date, From time, To time, Test Type (Subjective and Objective) | Schedule Test |
| FR2 | Staff member can reschedule and upload new test paper for scheduled subject. | Update Test |
| FR3 | Staff member can conduct a remedial test for students who have missed the test. | Remedial Test |
| FR4 | Students can submit the soft copy of solution for the subjective test. | Test Submission |
| FR5 | Staff members can cancel the scheduled test. | Cancel Test |
| FR6 | Staff member can upload the marks for the test which has been conducted as per the student’s submitted solution. | Test Mark Submission |
| FR7 | Staff member look at the marks secured by student. | Test Mark List |
| FR8 | Students can view their result of previous test. | View Marks |

**Module 6: Attendance Management**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| FR1 | Admin can mark the attendance of faculties and student. | Mark Attendance |
| FR2 | Staff member can mark the attendance of students. | Student Attendance |
| FR3 | Admin can mark the date of holidays or special occasions. | Mark Holidays |
| FR4 | Admin and staff member can view the monthly attendance report of students. | View Report |
| FR5 | Admin and staff member change or update the attendance of student. | Update Status |

**Module 7: Fee Management**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| FR1 | Student can pay online fee as per the package selected (Monthly, Quarterly). | Fee Payment |
| FR2 | Admin can view the report of fees of the which has been paid. | Fee Report |
| FR3 | Admin can view the report of students whose fee is pending. | Pending Fee |

**Module 8: Batch Management**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| FR1 | Admin can create a new batch by providing following information of the batch:  From Time, To Time, Student List | Batch Allocation |
| FR2 | Admin can change the allocated time of the batch with the students within it, as per the requirement. | Update Batch |
| FR3 | Admin, staff member and students can view the list of students in a batch. | List Students |
| FR4 | Admin and staff member can view the list of batches. | View Batch |
| FR5 | Admin can remove the entire batch in case none of the students exist or when a new session begins. | Remove Batch |

**3.1 Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| NFR1 | All the stored password will be encrypted using AES128 encryption algorithm. | Security |
| NFR2 | SQL Injection proof. | Security |
| NFR3 | Secured API call with JWT Tokens | API Security |
| NFR4 | Responsiveness | Design |

**CHAPTER 4: SYSTEM ANALYSIS**

**5.1 USE CASE**

**CHAPTER 5: SYSTEM DESIGN**

**5.1 Database Design**

Symbol of key=Primary key [PF], Foreign key[FK]

**Table Name: tblUser**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | UserID | Int | Primary Key, Auto Increment | Uniquely identifies each user. |
| 2. | First Name | String | Not Null | First Name of user. |
| 3. | Middle Name | String | Not Null | Middle Name of the user. |
| 4. | Last Name | String | Not Null | Last Name of the user. |
| 5. | EmailID | String | Unique key, Not Null | Unique Email ID for each user. |
| 6. | Contact | String | Unique Key, Not Null | Unique Contact for each user. |
| 7. | Address | String | Not Null | Address of user. |
| 8. | Date Of Birth | Date | Not Null | Date of Birth of the user. |
| 9. | Role | String | Not Null | Defines the role of user in system. |
| 10. | Status | String | Not Null | Defines the status of user in system. |
| 11. | Password | String | Not Null | Stores the hashed password of user. |
| 12. | Profile Picture | Blob | - | Stores the display picture of user. |
| 13. | RegisteredOn | Date | - | Specify the date on which the user was registered. |

**Table Name: tblStudent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | UserID | Int | Foreign Key, Not Null | Specify unique identity of the student. |
| 2. | CourseID | Int | Foreign Key, Not Null | Specify the course that student choose. |
| 3. | PackageID | Int | Foreign Key, Not Null | Specify the package that student chooses. |
| 4. | Standard | Int | Not Null | Specify the standard (class) in which student studies. |
| 5. | BatchID | Int | Foreign Key, Not Null | Specify the batch of the student. |

**Table Name: tblBatch**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | BatchID | Int | Primary Key, Not Null | Uniquely specify the batches. |
| 2. | FromTime | Time | Not Null | Specify the time from when batch begins. |
| 3. | ToTime | Time | Not Null | Specify the time upto when batch ends. |
| 4. | StartingDate | Date | Not Null | Specify the date from when the batch begin. |
| 5. | EndDate | Date | Not Null | Specify the expected date when the batch ends. |
| 6. | Standard | Int | Not null | Specify the standard’s batch going |

**Table Name: tblStaff**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | UserID | Int | Foreign Key, Not Null | Specify unique identity of the staff. |
| 2. | Experience | Int | Not Null | Specify the years of experience the staff had. |
| 3. | SubjectOfInterest | String | Not Null | Specify the subject in which the staff is interested. |
| 4. | Salary | Int | Not Null | Specify the salary of the staff. |
| 5. | Qualification | String | Not Null | Specify the qualification of the staff. |

**Table Name: tblSubject**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | SubjectID | Int | Primary Key, Auto Increment | Uniquely identifies the subjects. |
| 2. | SubjectCode | String | Not Null | Specify the subject code. |
| 3. | SubjectName | String | Not Null | Specify the name of the subject. |
| 4. | Description | String | Not Null | Description about the subject. |
| 5. | Remark | String | Not Null | Gives a remark about the subject. |
| 6. | Fees | Int | Not null | Subject wise fees |

**Table Name: tblsubject\_allocation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | ID | Int | Primary Key, Auto Increment | Uniquely specify. |
| 2. | Subject\_id | int | Foreign key ,Not Null | Specify the Subject. |
| 3. | Staff\_id | Int | Foreign key,Not Null | Specify the Staff. |
| 4. | Batch\_id | Int | Foreign key,Not Null | Specify the batch |

**Table Name: tblPackage**

( normal package provided by the tution )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | PackageID | Int | Primary Key, Auto Increment | Uniquely Specify the package. |
| 2. | PackageName | String | Not Null | Specify the name of the package. |
| 3. | Package\_Subjects\_ID | Int | Foreign key, Not Null |  |
| 4. | Description | String | Not Null | Gives a brief description of the package. |
| 5. | Duration | Int | Not Null | Specify the duration of the course. |
| 6. | Fee | Int | Not Null | Specify the total amount for the package. |

**Table Name: tblcustomized\_package**

( selective subject which student will select )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | PackageID | Int | Primary Key, Auto Increment | Uniquely Specify the package. |
| 2. | UserID | Int | Foreign key, Not Null |  |
| 3. | Package\_Subjects\_ID | Int | Foreign key, Not Null | Specifies Per student customized package id |
| 4. | PackageName | String | Not Null | Specify the name of the package. |
| 5. | Description | String | Not Null | Gives a brief description of the package. |
| 6. | Duration | Int | Not Null | Specify the duration of the course. |
| 7. | Fee | Int | Not Null | Specify the total amount for the package. |

**Table Name: tblSubjectsPackage**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | Package\_Subjects\_ID | Int | Primary key, Not Null | Uniquely Specify . |
| 2. | Package\_ID | Int | Foreign Key, Not Null | Specify the Package Name. |
| 3. | Subjects\_ID | Int | Foreign Key, Not Null | Specify the Subject Name. |

**Table Name: tblAttendance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | UserID | Int | Foreign Key, Not Null | Uniquely Identifies the user. |
| 2. | Date | Date | Not Null | Defines the date of attendance. |
| 3. | Status | Boolean | Not Null | Defines the attendance status. |
| 4. | FromTime | Time | Not Null | Specifies the time. |
| 5. | ToTime | Time | Not Null | Specifies the time. |

**Table Name: tblFee**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Field Name | Data Type | Constraint | Description |
| 1. | StudentID | Int | Foreign Key, Not Null | Uniquely specifies the student. |
| 2. | Status | String | Not Null | Specifies the status of the fee. |
| 3. | PaymentMode | String | Not Null | Specifies the payment mode. |
| 4. | TransactionID | Int | - | Specifies the transaction id. |
| 5. | Date | Date | - | Specifies the transaction date. |

**CHAPTER 6: System Implementation**

**6.1 Screenshots**

**CHAPTER 9: FUTURE ENHANCEMENT**

The system could be modified suitably to be used by any TUITION owner and start a business. The system could be used on large platforms and also more features can be added and system can be enhanced further.

* **Conclusion:**

The project will allow managing and maintaining a TUITIONING effectively and overcome the issues of finding and maintaining STUDENT history and transaction details every time for getting information whenever required.