**Database Management System (DBMS)**

**PROJECT REPORT**

**Student Learning Platform (Database)**

**(BMU\_Learning)**

****

**Submitted To:** Dr. Brij Bihari Dubey

Asst. Prof. [SOET]

**Submitted By:** Anant Kaul (CSE-1)

(1800201C203)

[**Table of Contents**](https://docs.google.com/document/d/1imP0AG0rqjmHFdH1_kVQ9kfJ4JgMnz48qCAkvmwfnSY/edit#heading=h.ji188d827jh8)

|  |  |  |
| --- | --- | --- |
| ***S.No.*** | ***Topic*** | ***Page No.*** |
| **1.** | **Introduction:**   * A brief descriptive title of the project. * An abstract/ summary of the project. | **4** |
| **2.** | **Objective:**   * The problem(s) that motivated/ required a solution provided by this project. * The specific problem(s) which this project is solving. * When & How this idea was first conceived? | **5** |
| **3.** | **Existing State-Of-The-Art vs The New Era:**   * Brief background of the existing knowledge. * The known way(s) about how others have tried to solve the same/ similar problem(s). * Drawbacks of the existing state-of-the-art & how this project overcomes (in tabular form). * Any prior art documentation or other material that explains or provides examples of such prior art efforts. * The features which are believed to be new and distinguished over the closest technology. | **8** |
| **4.** | **Alternatives:**   * Alternative way(s) of implementing this project. * Easy to come up with an alternative solution to the same problem that did not include details of this project? | **12** |
| **5.** | **Tools/ Methodology Used:**   * Hardware/ Software/ Tool used in the project. * Methodology used. | **13** |
| **6.** | **Functionalities:**   * A detailed explanation of how this project solves the problem(s). * Technical features and Elements of the project. * Block diagram(s) (ER Model). * Additional Information (Some Queries). | **14** |
| **7.** | **Deliverables/ Source Code/ Screenshots/ Results/ Outputs/ Evidences:**   * Source Code. * Screenshots of the Output/ Result. * Particulars of the first time it was successfully built or implemented (when, where, by whom, and evidence of this event including written/ on-line pointers to documentary evidence). | **23** |
| **8.** | **Conclusion:**   * The Concluding Note. * Status of the project. * The names of products that this project will be used in (if any)/ Future Aspects. | **50** |
| **9.** | **References** | **52** |

**Introduction**

* **A brief descriptive title of the project:**

Student Learning Platform (Database) named as ‘BMU\_Learning’ that can benefit students, faculty members, managers and academic coordinators.

* **An abstract/ summary of the project:**

This project presents the analysis and design of an innovative web based eLearning platform featuring BMU\_Learning. A further contribution could be built (future idea) presenting the effective construction of a web based platform having this database in the back-end. Following the selected framework(s), this report defines the requirements of this platform and proposes a solution to satisfy those requirements in terms of software artefacts. The design includes challenging non-functional requirements, such as scalability, security and performance. By introducing the new technology of eLearning to the new generation, makes it easy for all the students as well as the other faculty members involved to learn and manage all the stuff smartly and conveniently.

**Objective**

* **The problem(s) that motivated/ required a solution provided by this project:**
* A challenge in formative assessment is the potential loss of engagement of the student when a skill needs to be exercised repeatedly. A possible solution is the use of the same set of techniques applied in games to keep players motivated.
* Learning analytics continues ongoing trends on interactive learning analysis to apply them both in traditional eLearning systems and in new massive course platforms (MOOCs), where interaction data is generated at a very large scale (“big data”). Large-scale data collection, processing and analysis using suitable models and reasoning can improve the use of existing eLearning systems, as well as improving the system themselves.
* Existing learning analytical tools are very restricted to the types of data that can be collected and analysed, and do not have adaptability to treat "rich" data in terms of mutual interconnections and semantic context, relating to the learning style and actions and complex interactions of the student. In addition, usually the data visualization techniques are developed separately from the rest of the process of collection and analysis; therefore, they do not offer a complete solution from the collection of information to the presentation of the extracted knowledge. Finally, the usability of these tools is limited, and because of its sophistication, is particularly complex for teachers and non-technical users to learn how to use and exploit them.
* **The specific problem(s) which this project is solving:**

The main goal of this work is to analyse and design an eLearning platform aiming to provide learning management systems (LMS) with innovative services in terms of learning analytics and gamification. The development of this platform is the technological database used in the back-end.

This impact will be mainly achieved through three innovative pedagogical and technological axes:

* Formative assessment tools (FAT for brevity) that can provide immediate feedback by means of automatic assessment.
* Learning analytics that monitor the activity and progress of the on-line teaching and learning processes supported by eLearning systems and applications, combining this information with other sources of academic and historical information.
* Gamification as an incentive scheme in order to motivate students to practice more frequently and increase their engagement in the learning experience.

Moreover, this improvement can address both the perspective of instructors and learners, as well as providing program and university managers with sufficient tools to improve the educational portfolio.

* **When & How this idea was first conceived?**

The development of this Student Learning Platform (idea) was implemented keeping in mind, the main technological goal of a research project called ICT-FLAG, which is currently undertaken in the context of university degrees in the area of Information and Communication Technologies (ICT). BMU\_Learning aims in enhancing education through Formative Assessment, Learning Analytics and Gamification. Also, keeping the current situation of Coronavirus (Covid-19) in mind, it is the gateway for the new era.

**‘Existing State-of-The-Art’ vs ‘The New Era’**

* **Brief background of the existing knowledge:**

Historically, learning analytics appears from many researches about the processing, analysis and visualization of knowledge about the learning process. This knowledge may be gathered from large sets of events at different levels of abstraction, which cover the interactions of students with learning management systems, other students and instructors throughout the learning process.

Considering the historical method and the problem part, now we can say that in this way, learning analytics have addressed classical problems in eLearning, solving them at least partially.

* **The known way(s) about how others have tried to solve the same/ similar problem(s):**

Fortunately, there are some initiatives to develop standards for e-learning content interoperability. From these approaches, some of them have addressed the problem of student performance:

1. The IEEE Standard for Learning Technology standards' family.
2. The Experience API.

* **Drawbacks of the existing state-of-the-art & how this project overcomes (in tabular form):**

|  |  |  |  |
| --- | --- | --- | --- |
| ***S.No.*** | ***Existing State-of-The-Art*** | ***Drawbacks in the existing state-of-the-art.*** | ***How this Overcomes?*** |
| 1. | Classical Problem | There is a lack of integral solutions that can support all the previously mentioned aspects and incorporate them openly and transparently in current LMSs. | Experience API with self-assessment is a specification for collection and managing eLearning activities, enabling the communication of eLearning information between different LMS and tools. |
| 2. | Classical Problem | There is a lack of a  common data model for representing student interactions: typically, each system uses its own model, which hinders the construction of a LA model that manages information from different sources. | Provides a data model for tracking and exchanging information of student interactions with learning content and a communication specification that allows to the LMS to query collected information. |
| 3. | Exercise Problem | A challenge in formative assessment is the potential loss of engagement of the student when a skill needs to be exercised repeatedly. | A possible solution is the use of the same set of techniques applied in games to keep players motivated, as done in this project (future aspects). |
| 4. | e-Assessment Tools Problem | Regarding the inclusion of gamification in the current tools for e-assessment purposes, its use is still in its infancy and is based more on methodological aspects than technological ones. | In order to ensure that e-assessment tools are portable and can be integrated with different platforms and tools, it is necessary to consider aspects of integration of e-learning tools and the use of standards. |
| 5. | e-Assessment Tools Problem | Current e-learning systems do not have built-in analytical and gamification tools for learning, and use them separately, being unable to leverage the experience and the results achieved to improve the quality of education and learning. | Creating a built-in analytical and gamification tools for learning, leveraging the experience and the results achieved to improve the quality of education and learning with more smart and convenient modes (future aspect). |

* **Any prior art documentation or other material that explains or provides examples of such prior art efforts:**

In addition to dealing with multiple components and services as mentioned in the report, there are many ways in which others have tried to solve the same/similar problem from closed-source in-house solutions to open-source platforms such as Moodle.

* **The features which are believed to be new and distinguished over the closest technology:**
* Experience API with self-assessment.
* Enabling the communication of eLearning information between different LMS and tools.
* Providing a data model for tracking and exchanging information of student interactions with learning content and a communication specification that allows to the LMS to query collected information.
* Using the same set of techniques applied in games to keep players motivated, as done in this project.
* Considering aspects of integration of e-learning tools and the use of standards.
* Creating a built-in analytical and gamification tools for learning to improve the quality of education and learning with more smart and convenient modes.

**Alternatives**

* **Alternative way(s) of implementing this project:**

In addition to dealing with multiple components and services as mentioned in the report, there are many alternative ways in which others have tried to solve the same/similar problem from closed-source in-house solutions to open-source platforms such as Moodle.

* **Easy to come up with an alternative solution to the same problem that did not include details of this project?**

Do you really think that it would be easy to come up with an alternative solution to the same problem, that did not include the details of this project?

Honestly, if someone knew of my solution to the problem that is being solved by this project, it would be way more difficult for them to use my idea by not including the details of this project. It is so because this project includes many criteria depending upon the research done. Building a similar kind of Learning System, which doesn’t give an idea which this project tries to provide, would be a great drawback for them.

**Tools/Methodology Used**

* **Tool Used**: MySQL Workbench
* **Database**: MySQL
* **Methodology Used:**

The following **five viewpoints** can be considered to create an open and distributed system based on development standards:

* **Enterprise viewpoint:** Focuses on the purpose, scope and policies of the system, describing the information managed by the system and the structure and content type of the supporting data.
* **Information viewpoint:** Focuses on the semantics of the information and the information processing performed. It describes the information managed by the system and the structure and content type of the supporting data.
* **Computational viewpoint:** Enables distribution through functional decomposition of the system into objects which interact at interfaces. It describes the functionality provided by the system and its functional decomposition.
* **Engineering viewpoint:** Focuses on the mechanisms and functions required to support distributed interactions between objects in the system. It describes the distribution of processing performed by the system to manage the information and provide the functionality.
* **Technological viewpoint:** Focuses on the choice of technology for the system. It describes the technologies chosen to provide the processing, functionality and presentation of information.

**Functionalities**

* **A detailed explanation of how this project solves the problem(s):**

This project works upon 2 categories. The first is the User Requirements, in which all the functionalities for the user are created at the developer level. A developer user is who handles/ creates the database on the back-end. End Users emerge in the second category. All of us, using this database for our daily work on the front-end of the software/ application/ website come under this category.

Describing an elaborated functionality and how each part of this project works, both the categories can be further classified as:

(A Clear image of the idea can be found in the attachment below)

1. **User Requirements:**

* A College is organized into ‘Departments’. Each department has unique name, a unique code/id, number of students and a particular ‘Faculty’ who supervises the department. We keep a track of the start date when the faculty began supervising the department
* A ‘Department’ offers multiple ‘Courses’. Every course has a name, a unique course code/id, semester, it’s department code/id, a pre-defined number of credits and some prerequisites. The courses have corresponding ‘Notes’ and ‘Tests’ that are uploaded by the ‘Faculty’.
* Each ‘Faculty’ has a unique id, name, gender, e-mail id, designation, it’s department code/id. Each faculty has to teach at least one course.
* A ‘Student’ of the college has a unique id, name, gender, e-mail id and DOB. Each student of the college must belong to a department, containing it’s department code/id. A track is kept of the number of tests the student has attempted on the platform and accordingly the number of stars are received.
* Each ‘Test’ has a unique id, name, it’s duration and it’s link and is uploaded by the ‘Faculty’ having faculty id, is based on a particular ‘Course’ having course id. ‘Students’ who attempt a particular ‘Test’ are allotted maximum marks and an average score of them are generated. Each ‘Test’ also contains it’s duration and it’s link upon which the students are able to go and attempt. Not every course has tests but each test is based on a particular course.
* A ‘Course’ may or may not have ‘Notes’. The notes for any particular course have content links, it’s code/id, name, course code/id with unit (chapter name), id with name of the faculty who uploaded it and the number of students who like those notes material.

1. **End Users:**

These are the ones for which this project is mainly designed for. All of us, who (will) use this on the front-end are considered to be in this category. These are mainly students, faculty members, managers and academic coordinators, who get the most benefit of ‘BMU\_Learning’.

* **Technical features and Elements of the project:**

Some of the features of this project are listed as follows:

* **Customization:** Simple selection of which components are deployed in each installation.
* **Distribution:** Enables the distribution of components along different installations.
* **Extensibility:** Enables the easy extension of an installation with new components.

Further all the elements including the tables, views, triggers that are created/ edited for this database are given below in a tabular form:

(For a clear image, screenshots of results and outputs are provided below with the source code)

Table(s) details are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table No.*** | ***Table Name*** | ***Number of Attributes*** | ***Number of tuples inserted*** |
| 1. | Department | 3 | 4 |
| 2. | Student | 8 | 15 |
| 3. | Courses | 6 | 8 |
| 4. | Faculty | 6 | 10 |
| 5. | Notes | 6 | 5 |
| 6. | Test | 8 | 8 |
| 7. | Test\_Details | 4 | 23 |
| 8. | Notes\_Details | 3 | 10 |

View(s) details are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| ***View No.*** | ***View Name*** | ***Number of Attributes*** | ***Number of tuples generated*** |
| 1. | Department | 4 | 120 |

Trigger(s) details are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Trigger No.*** | ***Trigger Name*** | ***Operation Done on (Table)*** | ***Table Affected (Update Table)*** |
| 1. | Student\_AFTER\_INSERT | Student | Department |
| 2. | Student\_AFTER\_DELETE | Student | Department |

* **Block Diagram(s):**

All the block diagrams, including **Entity-Relationship (ER) Diagram** & **Conceptual Schema** are listed below:

(Link for a pdf format of both the diagrams are listed)

1. **Entity-Relationship Diagram (ER): (Link:** [**ER\_Diagram**](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/ER%20Diagrams/PDF/ER%20Diagram.pdf)**)**

**A picture containing text

Description automatically generated**

1. **Conceptual Schema: (Link:** [**Conceptual\_Schema**](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/ER%20Diagrams/PDF/Conceptual%20Schema.pdf)**)**

**A close up of a map

Description automatically generated**

* **Additional Information (Some Queries):**

Below mentioned are some queries, which can used to perform certain actions as commented against each of them.

(Try it for clarity and in the context of additional information)

(Link: [Additional\_Queries.sql](../../../Database%20(SQL%20Files)/Additional_Queries.sql))

**/\*To select all the students who have taken maximum number of test\*/**

SELECT Student\_ID, Name FROM Student

WHERE Student\_ID IN

(SELECT Student\_ID FROM Test\_Details

GROUP BY Student\_ID

HAVING COUNT(Student\_ID) =

(SELECT MAX(Marks) FROM

(SELECT COUNT(Student\_ID) AS Marks

FROM Test\_Details GROUP BY Student\_ID));

**/\*To select all the subjects in which students have taken maximum number of test\*/**

SELECT Test\_Name FROM Test\_Details

GROUP BY Test\_Name

HAVING COUNT(Test\_Name) =

(SELECT MAX(Marks) FROM

(SELECT COUNT(Test\_Name) AS Marks FROM Test\_Details GROUP BY Test\_Name));

**/\* To find average marks of students in a course which is uploaded by a particular faculty\*/**

SELECT AVG(Marks) FROM Test\_Details

WHERE Test\_Name IN

(SELECT Test\_Name FROM Test WHERE Faculty\_ID =

'LSEM500')group by Test\_Name;

**/\* How many courses a particular student has taken\*/**

SELECT COUNT(\*) FROM Course\_Details

WHERE Student\_ID = '1800201C200';

**/\* To select total number of students from each department\*/**

SELECT Department\_ID, COUNT(\*) FROM student

GROUP BY Department\_ID;

**/\* How many credits a student has taken \*/**

SELECT Student\_ID, SUM(Courses.Number\_Of\_Credits)

FROM Courses, Students\_Enrolled

WHERE Students\_Enrolled.Course\_ID = Courses.Course\_ID

GROUP BY Student\_ID;

**/\*To find average of all the subjects\*/**

SELECT Test\_Name, AVG(Marks) FROM Test\_Details

GROUP BY Test\_Name;

**/\* Select notes names where likes = 1 or if it belongs to a CS Department\*/**

SELECT DISTINCT Notes.Name

FROM Notes, Courses, Department

WHERE Number\_Of\_Likes = 1 OR

Department.Department\_ID = Courses.Department\_ID and Department.Department\_ID = 'CS' and Courses.Course\_ID = Notes.Course\_ID;

**Deliverables/ Source Code/ Screenshots/ Results/ Outputs/ Evidences:**

**Links:**

* [Create\_Database.sql](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Database%20(SQL%20Files)/Create_Database.sql)
* [Insert\_Values.sql](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Database%20(SQL%20Files)/Insert_Values.sql)
* [Create\_View.sql](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Database%20(SQL%20Files)/Create_View.sql)
* [Create\_Trigger.sql](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Database%20(SQL%20Files)/Create_View.sql)
* **Source Code (Database):**

**/\*DROP Database ‘BMU\_Learning as well as DROP all the tables before CREATING\*/**

DROP DATABASE BMU\_Learning;

CREATE DATABASE BMU\_Learning;

DROP TABLE Student;

DROP TABLE Department;

DROP TABLE Courses;

DROP TABLE Faculty;

DROP TABLE Notes;

DROP TABLE Test;

DROP TABLE Test\_Details;

DROP TABLE Course\_Details;

DROP TABLE Notes\_Details;

**//Creating table ‘Department’ and Inserting values**

CREATE TABLE Department

(

Department\_ID VARCHAR(10) PRIMARY KEY,

Department\_Name VARCHAR(60) NOT NULL,

Number\_Of\_Students INTEGER

);

INSERT into Department values('CS','Computer Science',11);

INSERT into Department values('EC','Electronics and Communication',1);

INSERT into Department values('EEE','Electrical Engineering',1);

INSERT into Department values('ME','Mechanical Engineering',2);

**//Creating table ‘Student’ and Inserting values**

CREATE TABLE Student

(

Student\_ID VARCHAR(15) PRIMARY KEY,

Name VARCHAR(60) NOT NULL,

Sex VARCHAR(1) NOT NULL,

DOB DATE,

Email VARCHAR(60) UNIQUE,

Stars INTEGER,

Number\_Of\_Tests INTEGER,

Department\_ID VARCHAR(10) NOT NULL,

FOREIGN KEY(Department\_ID) REFERENCES Department(Department\_ID)

);

INSERT into Student values('1800201C200','Sunil Kumar','M','1999/01/12','sunilkumar@gmail.com',1,2 ,'CS');

INSERT into Student values('1800202C200','Aman Gupta','M','1999/11/30','amangupta@gmail.com',1,3 ,'CS');

INSERT into Student values('1800203C200','Tulika Saxena','F','2000/02/29','tulikasaxena@gmail.com',1,1 ,'EC');

INSERT into Student values('1800204C200','Abhishek Tiwari','M','1999/09/18','abhiskek@gmail.com',1,0 ,'CS');

INSERT into Student values('1800205C200','Ankitha Anand','F','1999/03/23','ankitha@gmail.com',1,1 ,'CS');

INSERT into Student values('1800206C200','Granth Kohli','M','1999/05/07','granthkohli@gmail.com',1,0 ,'CS');

INSERT into Student values('1800207C200','Aditya Verma','M','2000/04/08','adityaverma@gmail.com',1,4 ,'CS');

INSERT into Student values('1800208C200','Surabhi Jain','F','1999/06/18','surabhijain@gmail.com',1,0 ,'EEE');

INSERT into Student values('1800209C200','Nikitha Bhatiya','F','1999/06/15','nikitha@gmail.com',1,2 ,'CS');

INSERT into Student values('1800210C200','Monika Agarwal','F','1999/09/10','monikaagarwal@gmail.com',1,0 ,'CS');

INSERT into Student values('1800211C200','Rahul Mishra','M','1999/07/21','rahulmishra@gmail.com',1,0 ,'CS');

INSERT into Student values('1800212C200','Gaurav Kumar','M','1999/11/29','gauravkumar@gmail.com',1,0 ,'CS');

INSERT into Student values('1800213C200','Aman Sharma','M','1999/12/31','amansharma@gmail.com',1,0 ,'ME');

INSERT into Student values('1800214C200','Shikar Mishra','M','1999/08/25','shikar@gmail.com',1,3 ,'ME');

INSERT into Student values('1800215C200','Isha Verma','F','1999/01/12','ishaverma@gmail.com',1,4 ,'CS');

**//Creating table ‘Courses’ and Inserting values**

CREATE TABLE Courses

(

Course\_ID VARCHAR(10) PRIMARY KEY,

Name VARCHAR(60) NOT NULL UNIQUE,

Department\_ID VARCHAR(10),

FOREIGN KEY(Department\_ID) REFERENCES Department(Department\_ID),

Number\_Of\_Credits INTEGER NOT NULL,

Semester INTEGER NOT NULL,

Prerequisites VARCHAR (10)

);

INSERT into Courses values('CSPY000001','Python','CS',4,1,'None');

INSERT into Courses values('CSCP000002','C Programming','CS',4,2,'None');

INSERT into Courses values('CSDS000003','Data Structures','CS',4,3,'CSCP000002');

INSERT into Courses values('CSAL000004','Algorithms','CS',4,4,'CSDS000003');

INSERT into Courses values('ECVL000001','VLSI','EC',4,5,'None');

INSERT into Courses values('ECSS000002','Signal and system','EC',4,4,'None');

INSERT into Courses values('MEFM000001','Fluid Mechanics','ME',4,3,'None');

INSERT into Courses values('EETS000001','Transmission System','EEE',4,6,'None');

**//Creating table ‘Faculty’ and Inserting values**

CREATE TABLE Faculty

(

Faculty\_ID VARCHAR(15) PRIMARY KEY,

Name VARCHAR(60) NOT NULL,

Sex CHAR(1),

Email VARCHAR(60) NOT NULL UNIQUE,

Designation VARCHAR(30) NOT NULL,

Department\_ID VARCHAR(10) NOT NULL,

FOREIGN KEY(Department\_ID) REFERENCES Department(Department\_ID)

);

INSERT into Faculty values('LSEM500','Savitha Sood','F','savithasood@gmail.com','Lecturer','CS');

INSERT into Faculty values('LSEM100','Anand Kumar','M','anandkumar@gmail.com','Professor','CS');

INSERT into Faculty values('LSEM800','Poonam Bisht','F','poonambisht@gmail.com','Professor','EC');

INSERT into Faculty values('LSEM200','Anu Banerjee','F','anubanerjee@gmail.com','Professor','EEE');

INSERT into Faculty values('LSEM300','Hari Chidella','M','harichindella@gmail.com','Lecturer','ME');

INSERT into Faculty values('LSEM900','Sunitha A','F','sunitha@gmail.com','Lecturer','EEE');

INSERT into Faculty values('LSEM050','Deepak Ahuja','M','deepakahuja@gmail.com','Associate Professor','CS');

INSERT into Faculty values('LSEM600','Gauri Govind','F','gaurigovind@gmail.com','Lecturer','EC');

INSERT into Faculty values('LSEM700','Vishal Kandpal','M','viahalkandpal@gmail.com','Assistant Professor','EC');

INSERT into Faculty values('LSEM400','Chandrashekhar B V','M','chandrashekar@gmail.com','Professor','ME');

**//Creating table ‘Notes’ and Inserting values**

CREATE TABLE Notes

(

Notes\_ID varchar(10) PRIMARY KEY,

Name VARCHAR(60) NOT NULL UNIQUE,

Course\_ID VARCHAR(10) NOT NULL,

Number\_Of\_Likes INTEGER,

Faculty\_ID VARCHAR(10) NOT NULL,

Link VARCHAR(100) NOT NULL UNIQUE,

FOREIGN KEY (Course\_ID) REFERENCES Courses(Course\_ID),

FOREIGN KEY(Faculty\_ID) REFERENCES Faculty(Faculty\_ID)

);

INSERT into Notes values('1010101010','Trees','CSDS000003',0,'LSEM100','https://drive.google.com/open?id=0B8F7Frp3Od9maV8yUVFDcWVMbDQ');

INSERT into Notes values('2020202020','Graphs','CSDS000003',0,'LSEM050','https://drive.google.com/open?id=0B8F7Frp3Od9mRGRyejlMSV81WDg');

INSERT into Notes values('3030303030','DataTypes','CSCP000002',0,'LSEM500','https://drive.google.com/open?id=0B8F7Frp3Od9ma0d0SU03akJqM2M');

INSERT into Notes values('5050505050','Structures','CSCP000002',0,'LSEM500','https://drive.google.com/open?id=0B8F7Frp3Od9mbkpIa2lqUmw2SU0');

INSERT into Notes values('4040404040','Fluid Mechanics','MEFM000001',0,'LSEM300','https://drive.google.com/open?id=0B8F7Frp3Od9mV1JXRlJHY09IOWc');

**//Creating table ‘Test’ and Inserting values**

CREATE TABLE Test

(

Test\_ID VARCHAR(10) PRIMARY KEY,

Test\_Name VARCHAR(50) NOT NULL UNIQUE,

Duration\_min INTEGER NOT NULL,

Max\_Marks\_Allotted INTEGER NOT NULL,

Avg\_Score INTEGER ,

Course\_ID VARCHAR(10),

Faculty\_ID VARCHAR(15) ,

Link VARCHAR(100) NOT NULL,

FOREIGN KEY(Course\_ID) REFERENCES Courses(Course\_ID),

FOREIGN KEY(Faculty\_ID) REFERENCES Faculty(Faculty\_ID)

);

INSERT into Test values(1,'data types in C',10,10,0,'CSCP000002','LSEM500','https://www.geeksforgeeks.org/c-language-2-gq/data-types-gq/');

INSERT into Test values(2,'List Tuples and Dictionary in Python',20,30,0,'CSPY000001','LSEM600','https://www.techbeamers.com/python-programming-questions-list-tuple-dictionary/');

INSERT into Test values(3,'Queue',15,10,0,'CSDS000003','LSEM100','https://www.geeksforgeeks.org/data-structure-gq/queue-gq/');

INSERT into Test values(4,'Stack',10,10,0,'CSDS000003','LSEM100','https://www.geeksforgeeks.org/data-structure-gq/stack-gq/');

INSERT into Test values(5,'Fluid Mechanics',30,20,0,'MEFM000001','LSEM400','https://examtimequiz.com/multiple-choice-questions-fluid-mechanics/');

INSERT into Test values(6,'Vlsi Design and technology',25,30,0,'ECVL000001','LSEM800','https://electronicspost.com/multiple-choice-questions-and-answers-on-vlsi-design-technology/');

INSERT into Test values(7,'Transmission System',15,30,0,'EETS000001','LSEM200','https://scholarexpress.com/multiple-choice-questions-mcq-on-transmission-system/');

INSERT into Test values(8,'data types in python',10,10,0,'CSPY000001','LSEM500','https://www.geeksforgeeks.org/c-language-2-gq/data-types-gq/');

**//Creating table ‘Test\_Details’ and Inserting values**

CREATE TABLE Test\_Details

(

Test\_ID VARCHAR(10) NOT NULL,

FOREIGN KEY(Test\_ID) REFERENCES Test(Test\_ID),

Test\_Name VARCHAR(50) NOT NULL ,

FOREIGN KEY(Test\_name) REFERENCES Test(Test\_name),

Student\_ID VARCHAR(15) NOT NULL ,

FOREIGN KEY(Student\_ID) REFERENCES Student(Student\_ID),

Marks INTEGER

);

INSERT into Test\_Details values(6,'Vlsi Design and technology','1800203C200',15);

INSERT into Test\_Details values(1,'data types in C','1800201C200',10);

INSERT into Test\_Details values(1,'data types in C','1800202C200',9);

INSERT into Test\_Details values(1,'data types in C','1800205C200',7);

INSERT into Test\_Details values(1,'data types in C','1800207C200',10);

INSERT into Test\_Details values(1,'data types in C','1800215C200',10);

INSERT into Test\_Details values(1,'data types in C','1800209C200',10);

INSERT into Test\_Details values(2,'List Tuples and Dictionary in Python','1800209C200',25);

INSERT into Test\_Details values(2,'List Tuples and Dictionary in Python','1800201C200',25);

INSERT into Test\_Details values(2,'List Tuples and Dictionary in Python','1800215C200',28);

INSERT into Test\_Details values(2,'List Tuples and Dictionary in Python','1800214C200',28);

INSERT into Test\_Details values(2,'List Tuples and Dictionary in Python','1800202C200',28);

INSERT into Test\_Details values(2,'List Tuples and Dictionary in Python','1800207C200',30);

INSERT into Test\_Details values(3,'Queue','1800215C200',8);

INSERT into Test\_Details values(3,'Queue','1800214C200',5);

INSERT into Test\_Details values(3,'Queue','1800202C200',8);

INSERT into Test\_Details values(3,'Queue','1800212C200',8);

INSERT into Test\_Details values(3,'Queue','1800207C200',8);

INSERT into Test\_Details values(8,'data types in python','1800201C200',5);

INSERT into Test\_Details values(8,'data types in python','1800210C200',7);

INSERT into Test\_Details values(8,'data types in python','1800207C200',9);

INSERT into Test\_Details values(8,'data types in python','1800215C200',9);

INSERT into Test\_Details values(8,'data types in python','1800214C200',7);

**//Creating table ‘Notes\_Details’ and Inserting values**

CREATE TABLE Notes\_Details

(

Student\_ID VARCHAR(15) NOT NULL ,

Notes\_ID varchar(10) NOT NULL,

FOREIGN KEY (Notes\_ID) REFERENCES Notes(Notes\_ID),

FOREIGN KEY(Student\_ID) REFERENCES Student(Student\_ID),

Likes INTEGER

);

INSERT INTO Notes\_Details values('1800201C200','1010101010',1);

INSERT INTO Notes\_Details values('1800202C200','1010101010',1);

INSERT INTO Notes\_Details values('1800215C200','1010101010',1);

INSERT INTO Notes\_Details values('1800212C200','1010101010',1);

INSERT INTO Notes\_Details values('1800210C200','1010101010',1);

INSERT INTO Notes\_Details values('1800204C200','1010101010',1);

INSERT INTO Notes\_Details values('1800201C200','2020202020',1);

INSERT INTO Notes\_Details values('1800210C200','2020202020',0);

INSERT INTO Notes\_Details values('1800213C200','4040404040',1);

INSERT INTO Notes\_Details values('1800214C200','4040404040',0);

**//Creating View for Students enrolled in each of the Courses**

CREATE VIEW `Students\_Enrolled` AS

SELECT

S.Student\_ID AS Student\_ID,

S.Student\_Name AS Name,

C.Name AS Course\_Name,

C.Course\_ID AS Course\_ID

FROM

(Student S JOIN Courses C);

**/\*Creating Triggers for Updating ‘Number\_Of\_Students’ in ‘Department’ table for each insertion or deletion of the row in ‘Student’ table\*/**

DROP TRIGGER IF EXISTS `BMU\_Learning`.`Student\_AFTER\_INSERT`;

DELIMITER $$

USE `BMU\_Learning`$$

CREATE DEFINER = CURRENT\_USER TRIGGER `BMU\_Learning`.`Student\_AFTER\_INSERT` AFTER INSERT ON `Student` FOR EACH ROW

BEGIN

UPDATE Department SET Number\_Of\_Students = Number\_Of\_Students + 1

WHERE Department\_ID = NEW.Department\_ID;

END$$

DELIMITER ;

DROP TRIGGER IF EXISTS `BMU\_Learning`.`Student\_AFTER\_DELETE`;

DELIMITER $$

USE `BMU\_Learning`$$

CREATE DEFINER = CURRENT\_USER TRIGGER `BMU\_Learning`.`Student\_AFTER\_ DELETE ` AFTER DELETE ON `Student` FOR EACH ROW

BEGIN

UPDATE Department SET Number\_Of\_Students = Number\_Of\_Students - 1

WHERE Department\_ID = OLD.Department\_ID;

END$$

DELIMITER ;

* **Screenshots (Results/ Outputs):**

SELECT \* FROM Department; (Link: [Department\_Table](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Table/Department_Table.png))

**A close up of a sign

Description automatically generated**

SELECT \* FROM Student; (Link: [Student\_Table](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Table/Student_Table.png))

A screenshot of a computer

Description automatically generated

SELECT \* FROM Courses; (Link: [Courses\_Table](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Table/Courses_Table.png))

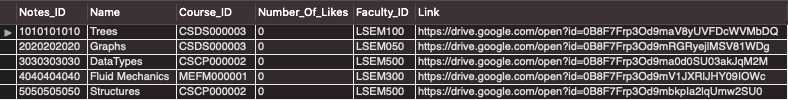
A screenshot of a cell phone

Description automatically generated

SELECT \* FROM Faculty; (Link: [Faculty\_Table](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Table/Faculty_Table.png))

A screenshot of text

Description automatically generated

SELECT \* FROM Notes; (Link: [Notes\_Table](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Table/Notes_Table.png))

A picture containing computer

Description automatically generatedSELECT \* FROM Test; (Link: [Test\_Table](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Table/Test_Table.png))

SELECT \* FROM Test\_Details; (Link: [Test\_Details\_Table](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Table/Test_Details_Table.png))

A close up of text on a black background

Description automatically generated

SELECT \* FROM Notes\_Details; (Link: [Notes\_Details\_Table](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Tables/Notes_Details_Table.png))

A picture containing sitting, large, white, pizza

Description automatically generated

SELECT \* FROM Students\_Enrolled; (Link: [Students\_Enrolled\_View(1)](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Tables/Students_Enrolled_View(1).png))A close up of text on a black background

Description automatically generated

SELECT \* FROM Students\_Enrolled; (Continue) (Link: [Students\_Enrolled\_View(2)](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Tables/Students_Enrolled_View(2).png))A close up of text on a black background

Description automatically generated

SELECT \* FROM Students\_Enrolled; (Continue) (Link: [Students\_Enrolled\_View(3)](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Tables/Students_Enrolled_View(3).png))A close up of text on a black background

Description automatically generated

SELECT \* FROM Students\_Enrolled; (Continue) (Link: [Students\_Enrolled\_View(4)](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Tables/Students_Enrolled_View(4).png))A close up of text on a black background

Description automatically generated

* **Particulars of the first time it was successfully built or implemented (when, where, by whom, and evidence of this event including written/ on-line pointers to documentary evidence):**

Particulars of the first time it was successfully built or implemented (when, where, by whom, and evidence of this event including written/ on-line pointers to documentary evidence) are provided below as well as in the attachment.

This evidence can be only given by showing the screenshots of the ‘MySQL Workbench’ working on my machine (OS: MacOS) named as BMU\_Learning, which is unique as there is no database present of such name in the eye of BML Munjal University.

(For a clear format of Evidences, Links are available)

**Proof (Tables):** **(Link:** [**Proof\_Tables**](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Proof/Proof_Tables.png)**)A screen shot of a computer

Description automatically generated**

**Proof (Indexes):** **(Link:** [**Proof\_Indexes**](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Proof/Proof_Indexes.png)**)A screen shot of a computer

Description automatically generated**

**Proof (Columns):** **(Link:** [**Proof\_Columns(1)**](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Proof/Proof_Columns(1).png)**)A screen shot of a computer

Description automatically generated**

**Proof (Columns): (Continue)** **(Link:** [**Proof\_Columns(2)**](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Proof/Proof_Columns(2).png)**)A screen shot of a computer

Description automatically generated**

**Proof (View):** **(Link:** [**Proof\_Views**](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Proof/Proof_Views.png)**)A screenshot of a computer screen

Description automatically generated**

**Proof (Triggers):** **(Link:** [**Proof\_Triggers**](file:///Users/avinashkaul/Desktop/Anant%20BMU%20Sem-IV/DBMS/Project/Report/1800201C203_AnantKaul/BMU_Learning/Screenshots/Proof/Proof_Triggers.png)**)A screenshot of a computer screen

Description automatically generated**

**Conclusion**

* **The Concluding Note:**

This report(project) has presented the first steps into the development of an eLearning platform called **BMU\_Learning**. The design of this platform has been described following the methodology, which considers five distinct viewpoints: enterprise, information, computation, engineering and technology.

Features such as scalability, modularity or security are a primary concern during the design process. Another relevant requirement is the need to integrate with a wide variety of eLearning tools, such as automated tools for formative assessment or different learning management systems. To this end, a client library that offers the **BMU\_Learning (Database)**.

* **Status of the project:**

This project is the main database that can be used in the back-end/ development phase of the full idea as mentioned above. But, status of this is **‘Built’** (Only the database part). For the full implementation of this idea, above is fully stated explanation. This can be further used in many forms mentioned in the future aspects section (next).

* **The names of the products that this project will be used in (if any)/ Future Aspects:**

This is the brief background of the database. Other components such as a BI module that allow BMU\_Learning to provide complex learning analytics is to be built-up as future aspects. For this course, this much was required (only database part).

As future work, there is a plan to improve the definition of this report for learning analytics, increasing the amount of customization offered to learners and instructors. There is also a plan to define a Domain-Specific Language (DSL) in order to let instructors define the gamification mechanics that should be implemented in their course.

Similarly, it can also be set up at database levels in the back-end for various Learning Management Systems (LMS) or many open-source platforms such as Moodle, Maitri, etc. which many of the schools, institutes, colleges and universities use for their management basis.

**References**

[1] <https://1000projects.org/projects/bca-projects>

[2] <https://www.w3schools.com/sql/default.asp>

[3] <https://app.diagrams.net/?src=about>

[4]<https://www.researchgate.net/publication/305278804_Analysis_and_Design_of_an_eLearning_Platform_Featuring_Learning_Analytics_and_Gamification>

[5] <https://www.geeksforgeeks.org/c-language-2-gq/data-types-gq/>

[6] <https://www.techbeamers.com/python-programming-questions-list-tuple-dictionary/>

[7] <https://www.geeksforgeeks.org/data-structure-gq/queue-gq/>

[8] <https://www.geeksforgeeks.org/data-structure-gq/stack-gq/>

[9] <https://examtimequiz.com/multiple-choice-questions-fluid-mechanics/>

[10] <https://electronicspost.com/multiple-choice-questions-and-answers-on-vlsi-design-technology/>

[11] <https://scholarexpress.com/multiple-choice-questions-mcq-on-transmission-system/>

[12] <https://www.geeksforgeeks.org/c-language-2-gq/data-types-gq/>

[13] https://github.com/topics/dbms-project?o=desc&s=updated