

Visvesvaraya Technological University

JnanaSangama, Belagavi – 590018, Karnataka



A Mini Project Report on “E-learning Management System”

*Submitted in partial fulfillment of the requirement for the DBMS Laboratory with
Mini Project (18CSL58) of V Semester*

**Bachelor of Engineering
in
Computer Science and Engineering**

Submitted By

YASHWANTH S (1GA19CS190)

Under the Guidance of

Mrs. Snigdha Sen

Assistant Professor,

Dept. of CSE



GLOBAL ACADEMY OF TECHNOLOGY

Department of Computer Science and Engineering

(Accredited by NBA 2019-2022)

Rajarajeshwari Nagar, Bengaluru – 560 098

2021-2022





GLOBAL ACADEMY OF TECHNOLOGY

Department of Computer Science and Engineering

(Accredited by NBA 2019-2022)

Rajarajeshwari Nagar, Bengaluru – 560 098



CERTIFICATE

This is to certify that the V Semester Mini Project in DBMS Laboratory entitled **“E-learning Management System”** carried out by **Yashwanth S (1GA19CS190)** is submitted in partial fulfillment for the award of the **BACHELOR OF ENGINEERING** in Computer Science and Engineering from **Visvesvaraya Technological University, Belagavi** during the year 2021-2022. The DBMS Mini Project report has been approved as it satisfies the academic requirements in respect of the mini-project work prescribed for the said Degree.

Mrs. Snigdha Sen

Asst Prof, Dept. of CSE

GAT, Bengaluru

Dr. Bhagyashri R Hanji

HOD, Dept. of CSE

GAT, Bengaluru

Name of the Examiners

Signature with date

1. _____

2. _____



GLOBAL ACADEMY OF TECHNOLOGY

Department of Computer Science and Engineering

(Accredited by NBA 2019-2022)

Rajarajeshwari Nagar, Bengaluru – 560 098



DECLARATION

I, **YASHWANTH S**, bearing USN **1GA19CS190**, student of Fifth Semester B.E, Department of Computer Science and Engineering, Global Academy of Technology, Rajarajeshwarinagar Bengaluru, declare that the Mini Project entitled “**E-LEARNING MANAGEMENT SYSTEM**” has been carried out by me and submitted in partial fulfillment of the course requirements for the award of degree in Bachelor of Engineering in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi during the academic year 2021-2022.

YASHWANTH S 1GA19CS190 _____

Place: Bengaluru

Date:

ABSTRACT

In many contemporary sectors, E-learning is often regarded as a ‘new’ form of learning that uses the affordances of the Internet to deliver customized, often interactive, learning materials and programs to diverse local and distant communities of practice. This view, however, is historically disconnected from its antecedent instantiations, failing to recognize the extensive links between developing educational theories and practices that had shaped the use of E-learning over the past 40 years. In addition, the historic divide between Education and Training has led to both the concurrent development of different notions, foci, and labels for technology-enhanced learning in different contexts and situations, and different conceptual origins arising in acquisitive and participatory learning metaphors. The purpose of E-learning Management System is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with. E-learning Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help their organization in better utilization of resources. The organization can maintain computerized records without redundant entries. Which implies that one need not be distracted by information that Is not relevant, while being able to reach the required data.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance crowned my efforts with success.

I consider myself proud, to be part of **Global Academy of Technology** family, the institution which stood by my way in endeavors.

I express my deep and sincere thanks to our Principal **Dr. N. Ranapratap Reddy** for his support.

I am grateful to **Dr. Bhagyashri R Hanji**, Professor and HOD, Dept of CSE who is source of inspiration and of invaluable help in channelizing my efforts in right direction.

I wish to thank my internal guide **Mrs. Snigdha Sen**, Assistant Professor, Dept of CSE for guiding and correcting various documents of mine with attention and care. She has taken lot of pain to go through the document and make necessary corrections as and when needed.

I would like to thank the faculty members and supporting staff of the Department of CSE, GAT for providing all the support for completing the Project work.

Finally, I am grateful to my parents and friends for their unconditional support and help during the course of my Project work.

YASHWANTH S
(1GA19CS190)

TABLE OF CONTENTS

Chapter No.		Title	Page No.
		ABSTRACT	i
		LIST OF FIGURES	iv
		LIST OF TABLES	v
1.		INTRODUCTION	1
	1.1	INTRODUCTION TO SQL	1
	1.2	INTRODUCTION TO FRONT END SOFTWARE	3
	1.3	PROJECT REPORT OUTLINE	4
2.		REQUIREMENT SPECIFICATION	5
	2.1	SOFTWARE REQUIREMENTS	5
	2.2	HARDWARE REQUIREMENTS	5
3.		OBJECTIVE OF THE PROJECT	6
4.		IMPLEMENTATION	7
	4.1	ER DIAGRAM	8
	4.2	SCHEMA DIAGRAM	9
	4.3	NORMALIZE THE RELATIONS	10
	4.4	CREATION OF TABLES	12
	4.5	INSERTION OF TUPLES	15
	4.6	CREATION OF TRIGGERS	16
	4.7	CREATION OF STORED PROCEDURES	17
5.		FRONT END DESIGN	18
	5.1	CONNECTIVITY TO DATABASE	18
	5.2	FRONT END CODE	19
6.		TESTING	23
	6.1	TESTING PROCESS	23
	6.2	TESTING OBJECTIVES	23
	6.3	TEST CASES	24
7.		RESULTS	25
	7.1	SNAPSHOTS	25
8.		CONCLUSION	30
9.		REFERENCES	31

LIST OF FIGURES

Figure No.	Title	Page No.
4.1	ER diagram of E-learning Management System	8
4.2	Schema diagram of E-learning Management System	9
4.5.1	Insertion of Tuples into Tutor Table	15
4.5.2	Insertion of Tuples into Course Table	15
4.6.1	Creation of Trigger	16
4.7.1	Creation of Stored Procedure	17
7.1.1	Home Page	25
7.1.2	Student Sign-up page	25
7.1.3	Student Login page	26
7.1.4	Tutor Login page	26
7.1.5	Student Home page	27
7.1.6	Course page	27
7.1.7	Course Enrollment page	28
7.1.8	Student Profile Update page	28
7.1.9	Tutor Home page	29
7.1.10	Student Marks Update page	29

LIST OF TABLES

Table No.	Title	Page No.
4.4.1	List of all Tables	12
4.4.2	Creation of Student Table	12
4.4.3	Creation of Course Table	13
4.4.4	Creation of Enrollment Table	13
4.4.5	Creation of Tutor Table	14
4.4.6	Creation of Assessment Table	14
6.3.1	Test cases	24

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SQL

The Structured Query Language (SQL) is the language of databases. All modern relational databases, including Access, FileMaker Pro, Microsoft SQL Server and Oracle use SQL as their basic building block. In fact, it's often the only way that you can interact with the database itself. All of the graphical user interfaces that provide data entry and manipulation functionality are nothing more than SQL translators. They take the actions you perform graphically and convert them to SQL commands understood by the database.

SQL IS LIKE ENGLISH

At this point, you might be thinking that you're not a programmer and learning a programming language is certainly not up your alley. Fortunately, at its core, SQL is a simple language. It has a limited number of commands, and those commands are very readable and are almost structured like English sentences.

INTRODUCING DATABASES

To understand SQL, it's important to have a basic understanding of how databases work. If you're comfortable with terms like table, relation and query, feel free to flow right ahead! If not, you may wish to read the article Database Fundamentals before moving on.

Let's look at an example. Suppose you have a simple database designed to keep the inventory for a convenience store. One of the tables in your database might contain the prices of the items on your shelves indexed by unique stock numbers that identify each item.

You'd probably give that table a simple name like "Prices."

Perhaps you want to remove items from your store that are priced over \$25, you would "query" the database for a list of all these items. This is where SQL comes in.

YOUR FIRST SQL QUERY

Before we get into the SQL statement required to retrieve this information, let's try phrasing a question in plain English.

We want to “select all stock numbers from the prices table where the price is over \$25.” That’s a simple request when expressed in plain English, and it’s almost as simple in SQL. Here’s the corresponding SQL statement:

```
SELECT StockNumber
FROM Prices
WHERE Price > 5
```

It’s as simple as that! If you read the statement above out loud, you’ll find that it’s extremely like the English question we posed in the last paragraph.

INTERPRETING SQL STATEMENTS

Now let’s try another example. This time, however, we’ll do it backwards. First, I’ll provide you with the SQL statement and let’s see if you can explain it in plain English:

```
SELECT Price
FROM Prices
WHERE StockNumber = 3006
```

So, what do you think this statement does?

That’s right, it retrieves the price from the database for item 3006.

There’s one simple lesson you should take away from our discussion at this point:

SQL is like English. Don’t worry about how you construct SQL statements; we’ll get to that in the rest of our series. Just realize that SQL isn’t as intimidating as it may first appear.

THE RANGE OF SQL STATEMENTS

SQL provides a wide range of statements, of which SELECT is just one . Here are some examples of other common SQL statements:

SQL INSERT and SQL DELETE: Inserts or deletes a record from a table

SQL UPDATE: Modifies records in a table

SQL CREATE and SQL DROP: Creates or deletes a table

In addition to these SQL statements, you can use SQL clauses, among them the WHERE clause used in the previous examples. These clauses serve to refine the type of data to act on. In addition to the WHERE clause, here are other commonly-used clauses:

AND or OR

Combine multiple conditions to refine a SQL statement

LIKE: Compares a value to similar values using a wildcard

ORDER BY: Sorts data in ascending or descending order

If you are interested in further exploring SQL, SQL Fundamentals is a multi-part tutorial that explores the components and aspects of SQL in more detail.

1.2 INTRODUCTION TO FRONT END SOFTWARE

HTML

HTML (Hypertext Markup Language) is the most basic building block of the Web. "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. HTML uses "markup" to annotate text, images, and other content for display in a Web browser.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

FLASK

- Flask ([source code](#)) is a Python web framework built with a small core and easy-to-extend philosophy.
- Flask is considered more because in common situations the equivalent Flask web application is more explicit.
- Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.
- It was created on April 1, 2010.

- Pros would be that the framework is light, there are little dependency to update and watch for security bugs, con is that some time you will have to do more work by yourself or increase yourself the list dependencies by adding plugins.
- In the case of Flask, its dependencies are:
 - Werkzeug a WSGI utility library
 - Jinja2 which is its template engine

1.3 PROJECT REPORT OUTLINE

The report is arranged in the following way:

Chapter 1: Introduction to SQL about its database, sql query, interpreting sql statements, AND or OR and range if sql statements and introduction to Front End software.

Chapter 2: Requirement specification of hardware and software

Chapter 3: Objective of the Project, design of project and developing.

Chapter 4: Implementation of ER diagram, Schema diagram, Normalize the relations, Creation of Tables, Insertion of Tuples, Creation of Triggers and Creation of Stored Procedure.

Chapter 5: Front End Design, connecting to database using PHP, Front end code of the Project.

Chapter 6: Testing of project by different cases, it's process and testing objectives

Chapter 7: Outcome of the Project

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

Operating System : Windows7 or higher

Database : MYSQL

Tools : Xampp 7.3.11 or higher

2.2 HARDWARE REQUIREMENTS

Processor : Any Processor above 500 MHz

RAM : 4.00GB

Hard Disk : 1TB

Compact Disk : 700Mb

Input device : Keyboard

Output device : Laptop Display Screen

CHAPTER 3

OBJECTIVE OF THE PROJECT

The main objectives of this application is to:

1. Provides the searching facilities based on various factors. Such as ASSESSMENT, TEACHER, STUDENT, COURSES.
2. The E-learning Management System also manages the CLASS details, online ASSESSMENT details and all.
3. It tracks all the information of Student, CLASS, and ASSESSMENT etc.
4. Manage the information of students.
5. Shows the information and description of the Assignment.
6. To increase efficiency of managing the Assignment with Students.
7. It deals with monitoring the information and transactions of ASSESSMENT.
8. Manage the information of ASSESSMENT.
9. Editing, adding and updating of Records is improved which results in proper resource management of Assignment data.
10. Manage the information of ASSESSMENT.

Scope of the project E-learning Management System:

It may help collect perfect management in detail. In a very short time, the collection will be obvious, simple and sensible. It also helps in current work relative to the E-learning Management System. It will also reduce the cost of collecting the management & collection procedure will go on smoothly. Our project aims at Business process automation, i.e. we have tried to computerize various processes of E-learning Management System.

CHAPTER 4

IMPLEMENTATION

4.1 ER DIAGRAM

The following ER DIAGRAM shows the entity relationships of E-learning Management System.

Let the attributes of the STUDENT entity be sid, name, DOB, city, phone, email, password.

Similarly, let the attributes of TUTOR entity be tid, name, cid, phone, email.

COURSE entity attributes be cid, name, duration, description.

ENROLLMENT entity attributes be eid, enrollmentDate, sid, cid.

ASSESSMENT entity attributes be cid, sid, asgmt1, asgmt2, asgmt3, finalmarks.

Let the 1st relationship be Enrolls between STUDENT and COURSE entities I.e, Student enrolls into courses with cardinality ratio M:N, Partial Participation from Student and Partial Participation from Course.

2nd relationship be TEACHES between TUTOR and COURSE entities i.e, Tutor teaches courses with cardinality 1:1, Partial Participation from Tutor and Total Participation from Course.

3rd relationship be HAVE between COURSE and ASSESSMENT entities i.e, Course have assessment with cardinality 1:N, Total Participation from Course and Total Participation from Assessment.

4th relationship be TAKES between STUDENT and ASSESSMENT entities i.e, Student takes up assessment with cardinality 1:1, Partial Participation from Student and Total Participation from Assessment.

Let the last relationship be GRADES between TUTOR and ASSESSMENT entities i.e, Tutor grades assessment with cardinality 1:N, Partial Participation from Tutor and Total Participation from Assessment.

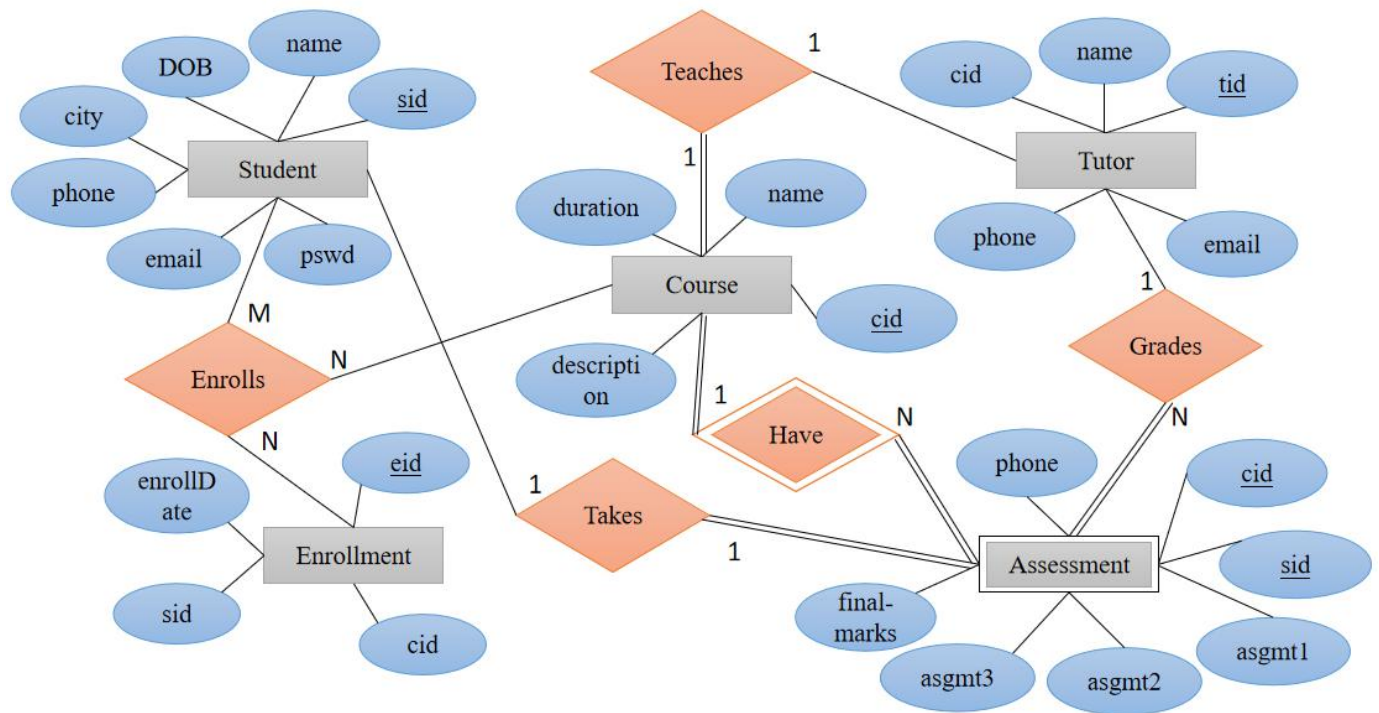


Figure 4.1 – ER Diagram of E-learning Management System

4.2 SCHEMA DIAGRAM

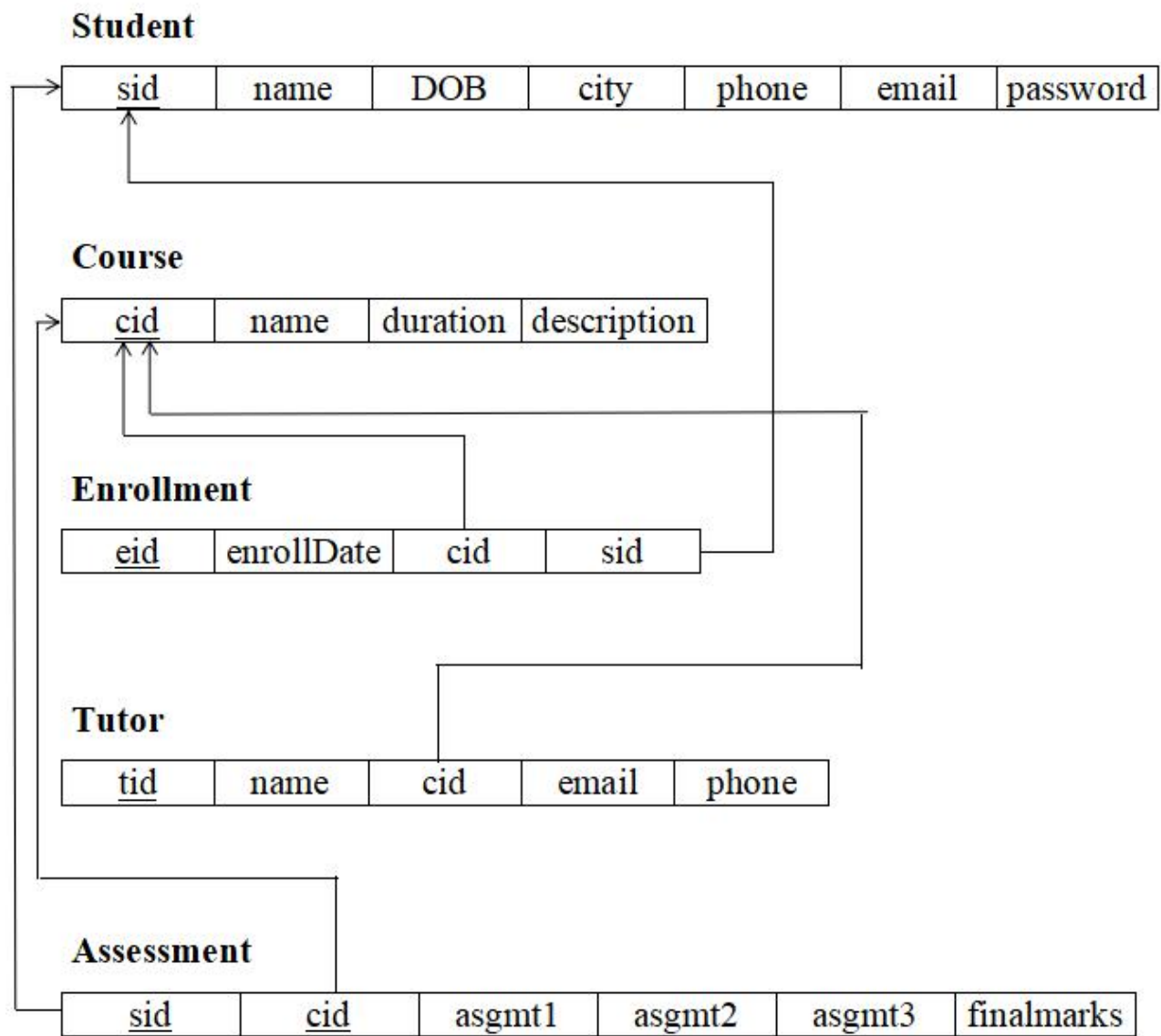


Figure 4.2 – Schema Diagram of E-learning Management System

4.3 NORMALIZE THE RELATIONS

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

1. Update anomalies - If data items are scattered and are not linked to each other properly, then it could lead to strange situations. For example, when try to update one data item having its copies scattered over several places, a few instances get updated properly while a few others are left with old values.

Such instances leave the database in an inconsistent state.

2. Deletion anomalies - When tried to delete a record, but parts of it were left undeleted because of unawareness, the data is also saved somewhere else.

3. Insert anomalies - When tried to insert data in a record that does not exist at all.

NORMALIZATION FORMS:

1. First Normal Form

First Normal Form is defined in the definition of relations (tables) itself. This rule defines that all the attributes in a relation must have atomic domains. The values in an atomic domain are indivisible units.

2. Second Normal Form

Before learning about the Second Normal Form, need to understand the following

- **Prime attribute** - An attribute, which is a part of the candidate-key, is known as a prime attribute.
- **Non-prime attribute** - An attribute, which is not a part of the prime-key, is said to be a non-prime attribute.

If we follow second normal form, then every non-prime attribute should be fully functionally dependent on the prime key attribute. That is, if $X \rightarrow A$ holds, then there should not be any proper subset Y of X , for which $Y \rightarrow A$ also holds true, partial dependency is not allowed in Second Normal Form.

3. Third Normal Form

For a relation to be in Third Normal Form, it must be in Second Normal Form and the following must satisfy

- No non-prime attribute is transitively dependent on prime key attribute.
- For any non-trivial functional dependency, $X \rightarrow A$, then either :-
 - ★ X is a super key or,
 - ★ A is a prime attribute, so there does not exist a transitive dependency.

4.4 CREATION OF TABLES

All the tables of ‘E-learning Management System’ are listed below:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> assessment	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> course	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> enrollment	Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_general_ci	48.0 KiB	-
<input type="checkbox"/> student	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> trig	Browse Structure Search Insert Empty Drop	25	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> tutor	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	48.0 KiB	-
6 tables	Sum	61	InnoDB	utf8mb4_general_ci	192.0 KiB	0 B

Table 4.4.1 – List of all Tables

```
CREATE TABLE Student (
    Sid int PRIMARY KEY,
    Name varchar(20),
    Dob date,
    City varchar(20),
    Phone number(10),
    Email varchar(20),
    Password varchar(100)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	sid	int(11)			No	None		AUTO_INCREMENT	Change Drop
<input type="checkbox"/> 2	name	varchar(20)	utf8mb4_general_ci		No	None			Change Drop
<input type="checkbox"/> 3	dob	date			No	None			Change Drop
<input type="checkbox"/> 4	city	varchar(20)	utf8mb4_general_ci		No	None			Change Drop
<input type="checkbox"/> 5	phone	bigint(10)			No	None			Change Drop
<input type="checkbox"/> 6	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop
<input type="checkbox"/> 7	password	varchar(1000)	utf8mb4_general_ci		No	None			Change Drop

Table 4.4.2 – Creation of Student Table

```
CREATE TABLE Course(
    Cid varchar(20) PRIMARY KEY,
    Name varchar(20),
    Duration varchar(20),
    Description varchar(20)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 cid	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	2 name	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 duration	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 description	varchar(1000)	utf8mb4_general_ci		No	None			Change Drop More

Table 4.4.3 – Creation of Course Table

```
CREATE TABLE Enrollment(
    Eid int PRIMARY KEY,
    Enrollmentdate date,
    Sid REFERENCES STUDENT(sid) ON DELETE CASCADE,
    Cid REFERENCES COURSE(cid) ON DELETE CASCADE
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 eid	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 enrollmentdate	date			No	None			Change Drop More
<input type="checkbox"/>	3 sid	int(11)			No	None			Change Drop More
<input type="checkbox"/>	4 cid	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More

Table 4.4.4 – Creation of Enrollment Table

```
CREATE TABLE Tutor(
    Tid int PRIMARY KEY,
    Name varchar(20),
    Phone number(10),
    Email varchar(50),
    Cid REFERENCES COURSE(cid) ON DELETE CASCADE
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	tid	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	name	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	phone	bigint(10)			No	None			Change Drop More
<input type="checkbox"/> 4	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 5	cid	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More

Table 4.4.5 – Creation of Tutor Table

```
CREATE TABLE Assessment(
    Sid REFERENCES STUDENT(sid) ON DELETE CASCADE,
    Cid REFERENCES COURSE(cid) ON DELETE CASCADE,
    Asgmt1 number(2), Asgmt2 number(2), Asgmt3 number(2),
    Finalmarks number(2)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	sid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	cid	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	asgmt1	int(2)			No	None			Change Drop More
<input type="checkbox"/> 4	asgmt2	int(2)			No	None			Change Drop More
<input type="checkbox"/> 5	asgmt3	int(2)			No	None			Change Drop More
<input type="checkbox"/> 6	finalmarks	int(11)			No	None			Change Drop More

Table 4.4.6 – Creation of Assessment Table

4.5 INSERTION OF TUPLES

```
INSERT INTO `tutor` (`tid`, `name`, `phone`, `email`, `cid`) VALUES ('4', 'Vivek',
'9901684109', 'vivek007@gmail.com', 'FCS01');
INSERT INTO `tutor` (`tid`, `name`, `phone`, `email`, `cid`) VALUES ('6', 'Kusuma',
'7895568923', 'kusuma@gmail.com', 'FCS02');
INSERT INTO `tutor` (`tid`, `name`, `phone`, `email`, `cid`) VALUES ('4', 'Harshith',
'967758931', 'harshith@gmail.com', 'FCS03');
```



















<div><div><div>←</div><div>T</div><div>→</div></div></div>					tid	name	phone	email	cid		
<input type="checkbox"/>		Edit		Copy		Delete	4	Yashwanth	7411990418	yash1ths21@gmail.com	FCS01
<input type="checkbox"/>		Edit		Copy		Delete	5	Vivek	9901684109	vivek007@gmail.com	FCS01
<input type="checkbox"/>		Edit		Copy		Delete	6	Kusuma	7895568923	kusuma@gmail.com	FCS02
<input type="checkbox"/>		Edit		Copy		Delete	7	Harshith	967758931	harshith@gmail.com	FCS03
<input type="checkbox"/>		Edit		Copy		Delete	8	Nischitha	7758904310	nischitha@gmail.com	FCS04
<input type="checkbox"/>		Edit		Copy		Delete	9	Lasya	6996905431	lasya@gmail.com	FCS05

Figure 4.5.1 – Insertion of Tuples into Tutor Table

```
INSERT INTO `tutor` (`cid`, `name`, `phone`, `duration`, `description`) VALUES('FCS01',
'Scratch', '3 months', 'Mon: 7pm-8pm/\r\nWed: 7pm-8pm/\r\nFri:7pm-8pm;');
INSERT INTO `tutor` (`cid`, `name`, `phone`, `duration`, `description`) VALUES('FCS02',
'Roblox', '6 months', 'Tue: 6pm-7pm/\r\nThur: 6pm-7pm;');
INSERT INTO `tutor` (`cid`, `name`, `phone`, `duration`, `description`) VALUES('FCS03',
'Python', '4 months', 'Tue: 7pm-8pm/\r\nThur: 7pm-8pm/\r\nSat:7pm-8pm;');
```

<div><div><div>←</div><div>T</div><div>→</div></div><div></div></div>				cid	name	duration	description
<input type="checkbox"/>	 Edit	 Copy	 Delete	FCS01	Scratch	3 months	Mon: 7pm-8pm/ Wed: 7pm-8pm/ Fri: 7pm-8pm;
<input type="checkbox"/>	 Edit	 Copy	 Delete	FCS02	Roblox	6 months	Tue: 6pm-7pm/ Thur: 6pm-7pm;
<input type="checkbox"/>	 Edit	 Copy	 Delete	FCS03	Python	4 months	Tue: 7pm-8pm/ Thur: 7pm-8pm/ Sat: 7pm-8pm;
<input type="checkbox"/>	 Edit	 Copy	 Delete	FCS04	Communication	3 months	Sat: 6pm-7pm/ Sun: 6pm-7pm;
<input type="checkbox"/>	 Edit	 Copy	 Delete	FCS05	Java	3 months	Wed: 6pm-7pm/ Fri: 6pm-7pm;

Figure 4.5.2 – Insertion of Tuples into Course Table

4.6 CREATION OF TRIGGERS

CREATE TRIGGER `Finalmarks`

BEFORE INSERT/UPDATE

ON `assessment`

FOR EACH ROW

set NEW.finalmarks=(NEW.asgmt1+NEW.asgmt2+NEW.asgmt3)/3;

The screenshot shows a web-based interface for creating a trigger. It has a 'Details' tab and a 'Definition' section. The 'Trigger name' is 'Finalmarks', the 'Table' is 'assessment', the 'Time' is 'BEFORE', and the 'Event' is 'INSERT'. The trigger definition is shown in a code editor with line numbers. The 'Definer' is 'root@localhost'.

Trigger name	Finalmarks
Table	assessment
Time	BEFORE
Event	INSERT
Definition	<pre>1 set NEW.finalmarks= (NEW.asgmt1+NEW.asgmt2+NEW.asgmt3) / 3</pre>
Definer	root@localhost

Figure 4.6.1 – Creation of Trigger

4.7 CREATION OF STORED PROCEDURES

```
CREATE PROCEDURE `Remarks` (IN `id` INT)
SELECT cid,asgmt1,asgmt2,asgmt3,finalmarks,
(
CASE
    WHEN finalmarks BETWEEN 23 AND 30 THEN 'Outstanding'
    WHEN finalmarks BETWEEN 18 AND 22 THEN 'Average' ELSE 'Weak'
END
) AS Remarks FROM assessment WHERE id=sid;
```

The screenshot shows a database management interface with a 'Details' tab. The 'Routine name' is 'Remarks' and the 'Type' is 'PROCEDURE'. Under the 'Parameters' section, there is a table with columns 'Direction', 'Name', 'Type', 'Length/Values', and 'Options'. A single parameter is defined: 'id' with direction 'IN' and type 'INT'. Below this is an 'Add parameter' button. The 'Definition' section contains the SQL code for the stored procedure, which is a SELECT statement with a CASE statement for grading based on finalmarks. The code is as follows:

```
1 SELECT cid,asgmt1,asgmt2,asgmt3,finalmarks,
2 (
3     CASE
4         WHEN finalmarks BETWEEN 23 AND 30 THEN
5         'Outstanding'
6         WHEN finalmarks BETWEEN 18 AND 22 THEN
7         'Average'
8         ELSE 'Weak'
9     END
10 )
11 AS Remarks FROM assessment WHERE id=sid
```

At the bottom right of the interface are 'Go' and 'Close' buttons.

Figure 4.7.1 – Creation of Stored Procedure

CHAPTER 5

FRONT END DESIGN

5.1 CONNECTIVITY TO DATABASE

- Most Web Applications :- Retrieve information from a database to alter their on-screen display-Store user data such as orders, tracking, credit card, etc. in a database.
- Permits them to adapt individual users, and provide fresh changing content.

FLASK: Database Access

- To add database functionality to Flask app, we will use SQLAlchemy
- SQLAlchemy is a Python SQL toolkit and object relational mapper(ORM) that enables python to communicate with the SQL database you prefer: MySQL, PostgreSQL, SQLite, and others
- SQLAlchemy is basically a bridge between Python and SQL database.
- Flask-SQLAlchemy is an extension for Flask that adds SQLAlchemy to your Flask app.

High-Level Process of Using MySQL from FLASK

- Create a database connection.
- Select database you wish to use.
- Perform a SQL query.
- Do something processing on query results.
- Close database connection.

CODE FOR CONNECTING FLASK TO MYSQL DATABASE

#DB connection

```
#app.config["SQLALCHEMY_DATABASE_URI"]="mysql://username:password@localhost/d  
atabase_table_name" (no password for our server)
```

```
app.config["SQLALCHEMY_DATABASE_URI"]="mysql://root:@localhost/e-learning ms"  
db=SQLAlchemy(app)
```

5.2 FRONT END CODE

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.[citation needed] The UML has become the standard language in object-oriented analysis and design.[citation needed] It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.[citation needed] System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation. In other words the first step in the solution to the problem is the design of the project.

HTML CODE FOR HEADER

```
<header id="header" class="fixed-top">
<div class="container d-flex align-items-center">
<h1 class="logo me-auto"><a href="#">FreeCodingSchool</a></h1>
<nav id="navbar" class="navbar order-last order-lg-0">
<ul>
<li><a class="active" href="/student">Home</a></li>
<li><a href="/courses">Courses</a></li>
<li><a href="#contact">Contact</a></li>
<li><a href="/logout">Logout</a></li>
</ul>
<i class="bi bi-list mobile-nav-toggle"></i>
<a href="/studentsignup" class="get-started-btn">Register</a>
</div>
</header>
```

HTML CODE FOR FOOTER

```
<footer id="footer">
<div class="footer-top">
<div class="container">
<div class="row">
<div id="contact" class="col-lg-3 col-md-6 footer-contact">
<h3>FreeCodingSchool</h3>
<p>
San Francisco, USA <br>
Bangalore, India<br>
<br>
<strong>Phone:</strong> +91 630-036-3638<br>
<strong>Email:</strong> info@freecodingschool.org<br>
</p>
<div class="container d-md-flex py-4">
<div class="me-md-auto text-center text-md-start">
<div class="copyright">
&copy; Copyright <strong><span>FreeCodingSchool</span></strong>. All Rights Reserved
```

</div>
 <div class="credits">
 Designed by Yashwanth and Vivek
 </div>
 </div>
 </div>
 </div>
 </footer>

HTML CODE FOR HOMEPAGE (INDEX PAGE)

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta content="width=device-width, initial-scale=1.0" name="viewport">

<title>FreeCodingSchool</title>

<meta content="" name="description">

<meta content="" name="keywords">

<link href="static/assets/css/style.css" rel="stylesheet">

</head>


<body>

<section id="hero" class="d-flex justify-content-center align-items-center">

<div class="container position-relative" data-aos="zoom-in" data-aos-delay="100">

<h1>Creating A Local Community<br>To Empower High School Students<br>To Learn
Programming</h1>

<h2 style="text-align:left">A Simple Mission To Close The Computer Science Gap<br>In High
Schools And Local Communities.</h2>

<a href="/studentsignup" class="btn-get-started">Register</a>

</div>

</section>

<main id="main">

<section id="about" class="about">

<div class="container" data-aos="fade-up">
```

```
<br>
<br>
<div class="row">
  <div class="col-lg-6 order-1 order-lg-2" data-aos="fade-left" data-aosdelay="100">
    
  </div>
  <div class="col-lg-6 pt-4 pt-lg-0 order-2 order-lg-1 content">
    <h3>BRINGING FREE AND OPEN LEARNING TO ANYONE INTRESTED TO LEARN CS
  </h3><br>
    <p>We focus on building sustainable Computer Science courese with a proven curriculum for
    high school students by pairing school teachers with CS grade students and tech professionals in
    the local community.
  </p>
  <br>

  <p>
    A simple mission to close the Computer Science gap in high schools and local
    communities.<br><br> We partner with schools/ organizations around the world to make
    sustainable CS curriculum accessible for high school students in our local communities.
  </p>
</div>
</div>
</div>
</section>
</main>
</body>
</html>
```

CHAPTER 6

TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TESTING PROCESS

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

6.2 TESTING OBJECTIVES

The main objectives of testing process are as follows.

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding undiscovered error.
- A successful test is one that uncovers the undiscovered error.

6.3 TEST CASES

The test cases provided here test the most important features of the project.

Table 6.3.1 - Test Cases

SL. NO.	TEST INPUTS	EXPECTED RESULTS	OBSERVED RESULTS	REMARKS
1	Insert a record	New tuple should be inserted	Query OK 1 row affected or inserted	PASS
2	Search a record	Search from existing records	Query OK 1 row affected or searched	PASS
3	Delete a record	Delete a record	Query OK 1 row affected or deleted	PASS
4	Create trigger	Trigger created	Query OK Trigger created	PASS
5	Create stored procedure	Stored procedure created	Query OK Stored Procedure Created	PASS

CHAPTER 7

RESULTS

This section describes the screens of the “E-LEARNING MANAGEMENT SYSTEM”. The snapshots are shown below for each module.

7.1 SNAPSHOTS

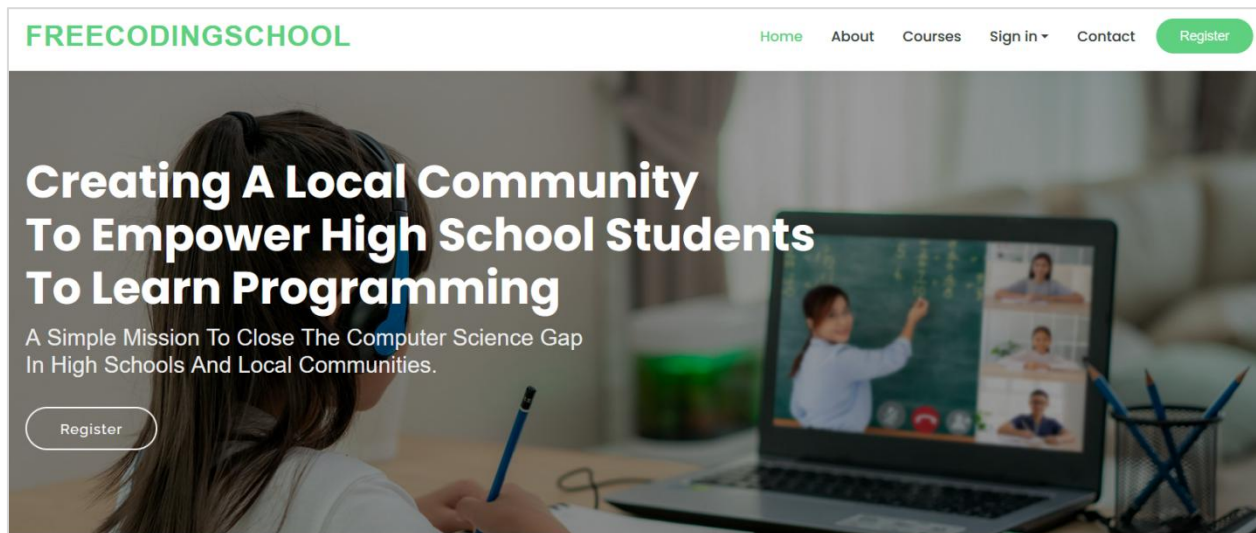


Figure 7.1.1 – Home Page

The image shows the sign-up page of FreeCodingsSchool. It has the same navigation bar as the homepage. The main content area is titled 'Sign Up Here' in a green box. Below this, there is a form with the following fields: 'Name' (with placeholder 'UserName'), 'Phone' (with placeholder 'Phone'), 'DOB' (with placeholder 'dd / mm / yyyy'), 'City' (with placeholder 'city'), 'Email address' (with placeholder 'Enter email'), and 'Password' (with placeholder 'Password'). A green 'Signup' button is located below the password field. At the bottom, there is a link that says 'Already a user ? Login'.

Figure 7.1.2 – Sign up Page

FREECODINGSCHOOL[Home](#)[About](#)[Courses](#)[Sign in](#)[Contact](#)[Register](#)

Student Login

Email address

Password

Login

Not a user ? [Signup](#)

FreeCodingSchool
San Francisco, USA
Bangalore, India

Phone: +91 630-036-3638
Email: info@freecodingschool.org

Useful Links

- > [Home](#)
- > [About us](#)
- > [Services](#)
- > [Terms of service](#)
- > [Privacy policy](#)

Our Services

- > [Volunteers](#)
- > [Teachers](#)
- > [Organizers](#)
- > [Partnered Schools](#)
- > [Sponsored Compaines](#)

Figure 7.1.3 - Student login page

FREECODINGSCHOOL[Home](#)[About](#)[Courses](#)[Sign in](#)[Contact](#)[Register](#)

Tutor Login

Email address

Password

Login

FreeCodingSchool
San Francisco, USA
Bangalore, India

Phone: +91 630-036-3638
Email: info@freecodingschool.org

Useful Links

- > [Home](#)
- > [About us](#)
- > [Services](#)
- > [Terms of service](#)
- > [Privacy policy](#)

Our Services

- > [Volunteers](#)
- > [Teachers](#)
- > [Organizers](#)
- > [Partnered Schools](#)
- > [Sponsored Compaines](#)

Figure 7.1.4 - Tutor login page

FREECODINGSCHOOL

[Home](#)
[Courses](#)
[Contact](#)
[Logout](#)

HELLO SANJAY!

Time Table				
Course ID	Name	Duration	Timings:	Unenrollment
FCS04	Communication	3 months	Sat: 6pm-7pm/ Sun: 6pm-7pm;	Un-enroll
FCS05	Java	3 months	Wed: 6pm-7pm/ Fri: 6pm-7pm;	Un-enroll
FCS03	Python	4 months	Tue: 7pm-8pm/ Thur: 7pm-8pm/ Sat: 7pm-8pm;	Un-enroll

Assessment Marks					
Course ID	Asgmt 1	Asgmt 2	Asgmt 3	Final Marks	Remarks
FCS03	22	20	19	20	Average
FCS04	27	27	28	27	Outstanding
FCS05	13	14	17	15	Weak

[Update Your Profile](#)


Figure 7.1.5 - Student home page

FREECODINGSCHOOL

[Home](#)
[About](#)
[Courses](#)
[Sign in](#)
[Contact](#)
[Register](#)

Courses

FreecodingSchool provides a network of volunteers from top companies and universities to help high schools create sustainable CS and digital education programs in their local community. We connect classroom teachers with our generous volunteers, through remote and classroom learning.




Scratch

Intro to Scratch programming

Scratch is a free programming language and online community where you can create your own interactive stories, games, and animations.

Yashwanth S

50
42




Roblox

Roblox Game Development

Create anything you can imagine with Roblox's free and immersive creation engine. Start creating experiences today!

Kusuma

35
42



Python

Intro to Python programming

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.

Harshith

20
85

Figure 7.1.6 - Course page

Dept. of CSE, GAT

2021-2022

27

FREECODINGSCHOOL[Home](#)[Courses](#)[Contact](#)[Logout](#)

Enrollment

Select Course

Date

Enroll

FreeCodingSchool

San Francisco, USA
Bangalore, India

Phone: +91 630-036-3638
Email: info@freecodingschool.org

Useful Links

- > Home
- > About us
- > Services
- > Terms of service
- > Privacy policy

Our Services

- > Volunteers
- > Teachers
- > Organizers
- > Partnered Schools
- > Sponsored Compaines

© Copyright **FreeCodingSchool**. All Rights Reserved
Designed by [Yashwanth](#) and [Vivek](#)

Figure 7.1.7 - Course enrollment page

FREECODINGSCHOOL[Home](#)[Courses](#)[Contact](#)[Logout](#)

Profile Update

Name

Phone

Email

Confirm Password

Update

FreeCodingSchool

San Francisco, USA
Bangalore, India

Phone: +91 630-036-3638
Email: info@freecodingschool.org

Useful Links

- > Home
- > About us
- > Services
- > Terms of service
- > Privacy policy

Our Services

- > Volunteers
- > Teachers
- > Organizers
- > Partnered Schools
- > Sponsored Compaines

© Copyright **FreeCodingSchool**. All Rights Reserved
Designed by [Yashwanth](#) and [Vivek](#)

Figure 7.1.8- Student profile update page

FREECODINGSCHOOL

[Home](#) [Courses](#) [Contact](#) [Logout](#)

Hello Instructor!

STUDENT DETAILS

Select Course Select your option ▾

Enter

Student ID	Course ID	Enrollment Date	Name	Email	Phone
9	FCS03	2022-01-13	ddd	ddd@gmail.com	4567890123
6	FCS03	2021-06-22	aaa	aaa@gmail.com	1234567890

Update Student Assessment Marks (click here)

Triggred Data				
Sl No	Student ID	Course ID	Activity	Date
24	9	FCS01	ENROLLED	2022-01-17
25	10	FCS01	ENROLLED	2022-01-17
26	6	FCS01	UNENROLLED	2022-01-17
27	8	FCS04	ENROLLED	2022-01-29
31	10	FCS01	UNENROLLED	2022-01-29
39	7	FCS05	UNENROLLED	2022-01-30

Figure 7.1.9 - Tutor home page

FREECODINGSCHOOL

[Home](#) [Courses](#) [Contact](#) [Logout](#)

Update Marks

SID

Enter Student id

Select Course

Select your option ▾

Assignment 1

Marks

Assignment 2

Marks

Assignment 3

Marks

Update

FreeCodingSchool

San Francisco, USA
Bangalore, India

Phone: +91 630-036-3638
Email: info@freecodingschool.org

Useful Links

[Home](#)
[About us](#)
[Services](#)
[Terms of service](#)
[Privacy policy](#)

Our Services

[Volunteers](#)
[Teachers](#)
[Organizers](#)
[Partnered Schools](#)
[Sponsored Compaines](#)

Figure 7.1.10 - Student marks update page

CONCLUSION

With the theoretical inclination of our syllabus it becomes very essential to take the utmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project “E-LEARNING MANAGEMENT SYSTEM” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development.

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project.

The project also provides us the opportunity of interacting with our teachers and to gain from their best experience.

REFERENCES

- [1] Fundamentals of Database System, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
- [2] Database Management System, Ramakrishna and Gehrke, 3rd Edition, 2014, McGraw Hill.
- [3] Welcome To Flask - Flask Documentation (<https://flask.palletsprojects.com/en/2.0.x/>)
- [4] ARK PRO CODER - <https://youtu.be/7Ge8JQcluol>
- [5] <https://www.w3schools.com>
- [6] <https://www.stackoverflow.com>
- [7] <https://www.tutorialspoint.com>
- [8] <https://getbootstrap.com>
- [9] <https://bootstrapmade.com>