



Project Title: University Management System

Student Name	Student Id	Contribution
ASHRAFUL ISLAM	20-42010-1	Introduction, Project proposal, Class Diagram, Use Case Diagram, Activity Diagram, ER diagram, User Interface
KAZI MUNTASHIR FAHAD	20-43834-2	Normalization, Up to 3rd NF, Schema Diagram, Scenario Description, Table Creation, trigger.
MT. ZUBIDA KHATUN	19-41677-3	Table Creation, Data Insertion, User Interface.
SUMAIA MAHMUDA	19-41622-3	Query Writing, Diagram Cheaking, Conclusion

**Course Name: ADVANCE DATABASE MANAGEMENT
SYSTEM.**

Section: B

Contents

Cover page.....	1
Introduction	3
Project Proposal.....	3
Class diagram.....	4
Use case diagram.....	5
Activity diagram	5
User Interface.....	6-7
Scenario description.....	8
ER diagram.....	9
Normalization.....	10-17
Schema diagram	18
Table creation and insertion	19-36
Query writing	37-57
Conclusion.....	58

Introduction:

University management system project is a web-based solution for colleges, universities, and schools. It was created for the university and its affiliated institutions to conduct, monitor, and analyze complicated activities such as student admission, examinations, and much more. It is a program that manages the complete student life cycle up to degree completion.

We have computers with large computing power and almost every business is going to take the advantages of using those technologies. But nowadays digital certifications itself become an essential component for every business infrastructure. Because it provides security and it can identify every unique individual. Besides, it also provides confidential communications to the users.

This university management system we developed it to solve the problems of the universities and make work smart instead of using papers now you can use this system to do all the work, every university or educational institution has challenges to overcome and manage the information of students, faculties, registrations courses and staff at the management level, this system designed to assist strategic planning and will help to ensure that the university can meet the minimum standard of the university.

The main aim of this Project University Management System is to manage student registration as well as staff registration so that the teachers can submit student exam results online and take a class attendance online during class sessions.

Project Proposal:

We have developed a database system, which aims to replicate an online university portal / management system.

Here we have user types such as students, teachers, admin and officers. Students will be able to see their information, results and apply for registering courses. Teachers will be able to evaluate students, update their marks and grades. The admin will allocate courses for students who have applied for registration, store and update various attributes of the student account and provide salary to the teachers. The officer will maintain the payment system for the university and create records relating to order and discipline.

Class Diagram: This is the class diagram of our project.



Fig 1: Class Diagram.

Use case Diagram: This is the use case diagram for our project.

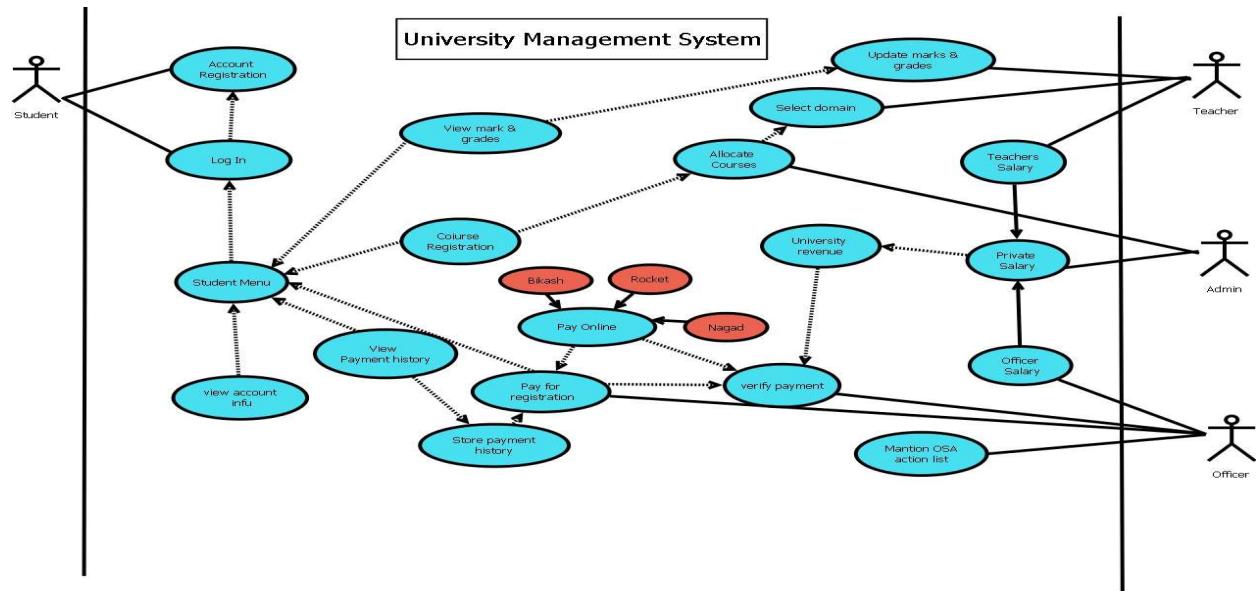


Fig 2: Use case diagram.

Activity Diagram: This is the activity diagram of our project

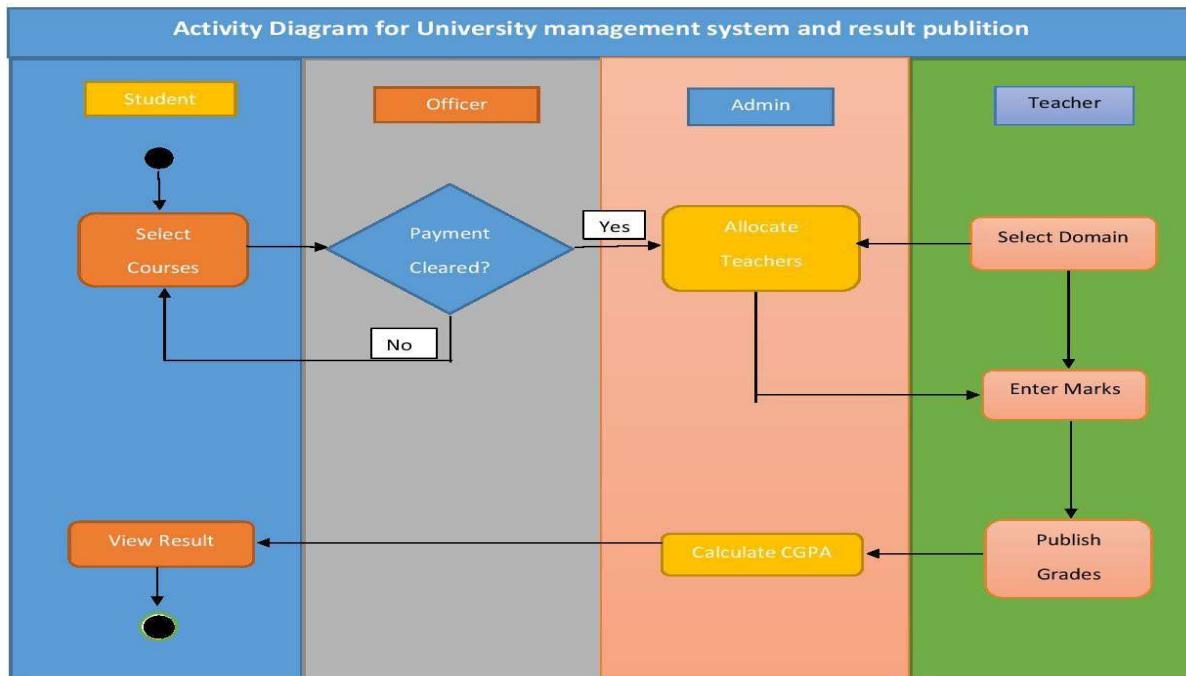
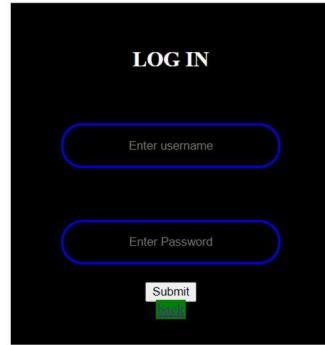


Fig 3: Activity Diagram

User Interface:



← → ⌛ ⓘ localhost/studentdetails.php

Student list

SL	Name	Id	View Profile	update
1	Maria Anders	19-41677-5	check profile	update
2	Francisco Chang	18-41567-3	check profile	update
3	janate	17-41677-3	check profile	update

← → ⌛ ⓘ localhost/td.php

Teacher list

SL	Name	Id	domain	Hiedate	balance
1	nazmus sakib	2001-4441-3	cse	17-5-2015	30000
2	nazia alfaz	2001-57896-3	cse	17-5-2022	2000
1	habiba khatun	2002-4441-2	EEE	17-5-2017	30000

← → ⌛ ⓘ localhost/course.php

Course List

course code	course Name	course prerequisite	credit count
cs123	operating system	microprocessor	3
eee897	digital logic and circuits	electric device	3
m345	physics 2	physics 1	3

OSA

student name	stdent id	osa serial	action	case	action date
mirza asif	19-41677-3	suspended	ragging	10-11-2022	

View Profile

Name	Maria Anders
Email	maria@gmail.com
Address	Dhaka
Phone	01720586978
ADmission Date	20-11-2019
Graduation Date	20-22-2025
Father Name	josim
Mother Name	jesmin

Scenario Description:

We have developed a database system which aims to replicate an university management system. Here we have user types such as students, teachers, admin for Office of Student Affair (OSA) and payment record officers.

In this management system student have their own information. They have their own ID, name, CGPA, program and also how much credit he or she has completed. Students will be able to see their information, results and apply for registering courses. Teachers have their own information. They have their own ID, name, hiredate, domain of their study and also balance information. Teachers will be able to evaluate students, update their marks and grades.

Students and Teachers has been identify by their unique ID. The admin will allocate courses for students who have applied for registration, store and update various attributes of the student account and also provide salary to the teachers. Payment record will also capture by an officer. Who will have information about paid amount, payable amount.

Officer also the information of the student id, student name for identity a student. Serial is unique for the Payment Record officer.

ER Diagram: This is the ER diagram of our project

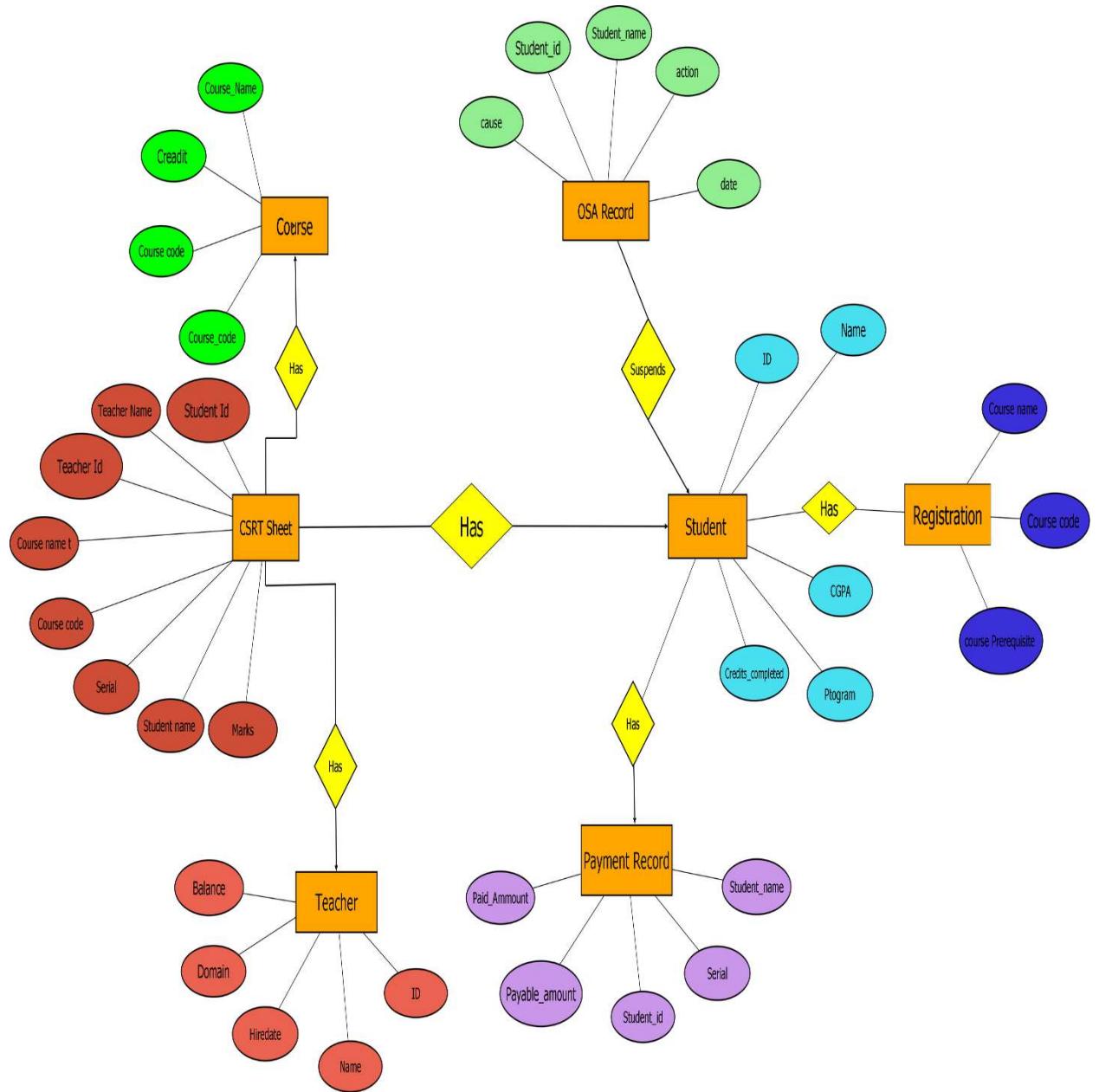


Fig 4: ER Diagram

Normalization: This is the basic normalization that required.

Has:

UNF:

Has (course_code, course_prerequisite_code, credit_count, course_name, ID, name, hiredate, domain, balance)

1NF: There is no multilevel attribute.

course_code, course_prerequisite_code, credit_count, course_name, ID, name, hiredate, domain, balance

2NF:

course_code, course_prerequisite_code, credit_count, course_name
ID, name, hiredate, domain, balance

3NF: There is no transitive dependency.

course_code, course_prerequisite_code, credit_count, course_name
ID, name, hiredate, domain, balance

Table Creation:

course_code, course_prerequisite_code, credit_count, course_name
ID, name, hiredate, domain, balance, course_code

Suspends:

UNF:

Suspends (student_id, student_name, action, cause, date, ID, name, CGPA, program, credits_completed)

1NF: There is no multilevel attribute.

student_id, student_name, action, cause, date, ID, name, CGPA, program, credits_completed

2NF:

student_id, student_name, action, cause, date ID, name, CGPA, program, credits_completed

3NF: There is no transitive dependency.

student_id, student_name, action, cause, date
ID, name, CGPA, program, credits_completed

Table Creation:

student_id, student_name, action, cause, date
ID, name, CGPA, program, credits_completed, student_id

Has:

UNF:

Has (ID, name, CGPA, program, credits_completed, serial, paid_amount, payable_amount, student_id, student_name)

1NF: There is no multilevel attribute.

ID, name, CGPA, program, credits_completed, serial, paid_amount, payable_amount, student_id, student_name

2NF:

ID, name, CGPA, program, credits_completed
serial, paid_amount, payable_amount, student_id, student_name

3NF: There is no transitive dependency.

ID, name, CGPA, program, credits_completed
serial, paid_amount, payable_amount, student_id, student_name

Table Creation:

ID, name, CGPA, program, credits_completed
serial, paid_amount, payable_amount, student_id, student_name, ID

Has:

UNF:

Has (course_code, course_prerequisite_code, credit_count, course_name, ID, name, CGPA, program, credits_completed)

1NF: There is no multilevel attribute.

course_code, course_prerequisite_code, credit_count, course_name, ID, name, CGPA, program, credits_completed

2NF:

course_code, course_prerequisite_code, credit_count, course_name
ID, name, CGPA, program, credits_completed

3NF: There is no transitive dependency.

course_code, course_prerequisite_code, credit_count, course_name
ID, name, CGPA, program, credits_completed

Table Creation:

course_code, course_prerequisite_code, credit_count, course_name
ID, name, CGPA, program, credits_completed, course_code

Has:

UNF:

Has (course_code, course_prerequisite_code, credit_count, course_name, ID, name, CGPA, program, credits_completed)

1NF: There is no multilevel attribute.

course_code, course_prerequisite_code, credit_count, course_name, ID, name, CGPA, program, credits_completed

2NF:

course_code, course_prerequisite_code, credit_count, course_name
ID, name, CGPA, program, credits_completed

3NF: There is no transitive dependency.

course_code, course_prerequisite_code, credit_count, course_name
ID, name, CGPA, program, credits_completed

Table Creation:

course_code, course_prerequisite_code, credit_count, course_name
ID, name, CGPA, program, credits_completed, course_code

Has:

UNF:

Has (ID, name, CGPA, program, credits_completed, course_name, course_code, course_prerequisite)

1NF: There is no multilevel attribute.

ID, name, CGPA, program, credits_completed, course_name, course_code, course_prerequisite

2NF:

ID, name, CGPA, program, credits_completed
course_name, course_code, course_prerequisite

3NF: There is no transitive dependency.

ID, name, CGPA, program, credits_completed
course_name, course_code, course_prerequisite

Table Creation:

ID, name, CGPA, program, credits_completed
course_name, course_code, course_prerequisite

Has:

UNF:

Has (student_name, student_id, teacher_name, teacher_id, course_name, course_code, serial, marks, course_name, course_code, credit)

1NF: There is no multilevel attribute.

student_name, student_id, teacher_name, teacher_id, course_name, course_code, serial, marks, course_name, course_code, credit

2NF:

student_name, student_id, teacher_name, teacher_id

course_name, course_code, serial, marks, course_name, course_code, credit

3NF: There is no transitive dependency.

student_name, student_id, teacher_name, teacher_id

course_name, course_code, serial, marks, course_name, course_code, credit

Table Creation:

student_name, student_id, teacher_name, teacher_id

course_name, course_code, serial, marks, course_name, credit

Temporary Table:

course_code, course_prerequisite_code, credit_count, course_name
ID, name, hiredate, domain, balance, **course code**

student_id, student_name, action, cause, date

ID, name, CGPA, program, credits_completed, **student id**,
ID, name, CGPA, program, credits_completed
serial, paid_amount, payable_amount, student_id, student_name, **ID**
course_code, course_prerequisite_code, credit_count, course_name
ID, name, CGPA, program, credits_completed, **course code, student id**,
—ID, name, CGPA, program, credits_completed
course_name, course_code, course_prerequisite
student_name, student_id, teacher_name, teacher_id
course_name, course_code, serial, marks, course_name

Final Table:

course_code, course_prerequisite_code, credit_count, course_name
ID, name, hiredate, domain, balance, **course code**

student_id, student_name, action, cause, date

serial, paid_amount, payable_amount, student_id, student_name, **ID**
ID, name, CGPA, program, credits_completed, **course code, student id**
course_name, course_code, serial, marks, course_name
course_name, course_code, serial, marks, course_name

Schema diagram: This is the schema diagram of our project

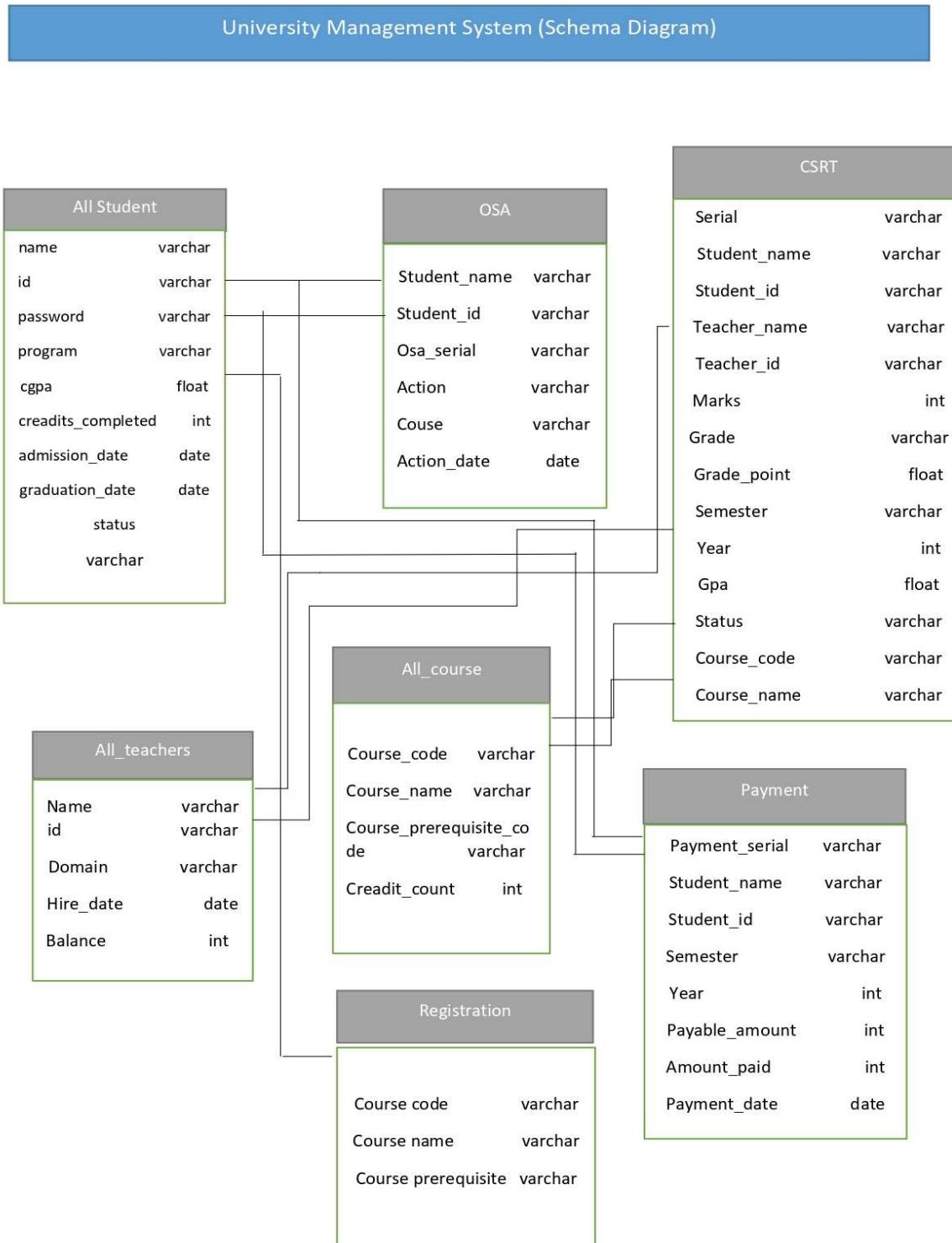


Fig 5: Schema diagram

Table Creation & Data Insertion:

Student Table:

```
all_student
create table all_student (
    name      varchar(50) not null,
    id        int constraint pk_all_student primary key, password
    varchar(25) not null,
    program   varchar(10), cgpa
    float, credits_completed    int,
    admission_date
    date,
    graduation_date
    date, status
    varchar(20));
```

```
create sequence all_student_id
    increment by 1
    start with 1
    nocycle
    nocache;
```

```
CREATE INDEX student_idx ON
    all_student(name);
```

Insertion:

```
insert into all_student values
('Waskiqure',all_student_id.nextval,'0161','EEE',3.50, 19,to_date('10-dec-22','dd-
mm-yyyy'), null,'Studying');
```

```
insert into all_student values
('Shahriar Zaman',all_student_id.nextval,'1234','CSE',3.84, 16,to_date('19-dec-
21','dd-mm-yyyy'), null,'Studying');
```

```
insert into all_student values
('sumsur nahar',all_student_id.nextval,'0159','EEE',3.25, 15,to_date('19-dec- 22','dd-mm-
yyyy'), null,'Studying');
```

```
insert into all_student values
('AMANULLAH',all_student_id.nextval,'0160','CSE',2.98, 15,to_date('19-dec-
21','dd-mm-yyyy'), null,'Studying');
```

```
insert into all_student values
('Zisnara',all_student_id.nextval,'0161','CSE', 3.98, 14,to_date('10-dec-22','dd-
mm-yyyy'), null,'Studying');
```

127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE Database Express Edition

User: SYS

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
create table all_student
(
    name      varchar(50) not null,
    id        int constraint pk_all_student primary key,
    password  varchar(25) not null,
    program   varchar(10),
    gpa       float,
    credits_completed  int,
    admission_date    date,
    graduation_date   date,
    status     varchar(20)
);
```

Results Explain Describe Saved SQL History

Table created.

0.19 seconds

Language: en-us

127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE Database Express Edition

User: SYS

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
CREATE INDEX student_idx
ON all_student(name);
```

Results Explain Describe Saved SQL History

Index created.

0.05 seconds

← ⌂ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
create sequence all_student_id
    increment by 1
    start with 1
    nocycle
    nocache;
```

Results Explain Describe Saved SQL History

Sequence created.

0.03 seconds

← ⌂ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
select * from all_student;
```

Results Explain Describe Saved SQL History

NAME	ID	PASSWORD	PROGRAM	CGPA	CREDITS_COMPLETED	ADMISSION_DATE	GRADUATION_DATE	STATUS
Waskique	1	0161	EEE	3.5	19	10-DEC-22	-	Studying
Shahriar Zaman	2	1234	CSE	3.84	16	19-DEC-21	-	Studying
sumsur nahar	3	0159	EEE	3.25	15	19-DEC-22	-	Studying
AMANULLAH	4	0160	CSE	2.98	15	19-DEC-21	-	Studying
Zisnara	5	0161	CSE	3.98	14	10-DEC-22	-	Studying

5 rows returned in 0.12 seconds

[CSV Export](#)

Teachers Table:

```
create teacher
```

```
create table all_teachers ( name  
                           varchar(50) not null,  
                           id      int constraint pk_all_teachers primary key, domain  
                           varchar(10),  
                           hire_date date, balance  
                           int);
```

```
create sequence all_teachers_id  
           increment by 1  
           start with 1  
           nocycle  
           nocache;
```

```
CREATE INDEX teacher_idx  
  ON all_teachers(name,id);
```

Insertion:

```
insert into all_teachers values  
('Nazmul Hossain',all_teachers_id.nextval,'CSC',to_date('17-11-2022','dd-mm-  
yyyy'),30000 );
```

```
insert into all_teachers values  
('kuwsur imron',all_teachers_id.nextval,'CSC',to_date('5-10-2022','dd-mm-  
yyyy'),30000 );
```

```
insert into all_teachers values  
('nazia alfaz',all_teachers_id.nextval,'CSC',to_date('5-1-2022','dd-mm- yyyy'),50000  
);
```

```
insert into all_teachers values  
('tanjil amin',all_teachers_id.nextval,'EEE',to_date('2-8-2021','dd-mm-  
yyyy'),50000 );
```

```
insert into all_teachers values  
('asif iqbal',all_teachers_id.nextval,'BBA',to_date('2-6-2021','dd-mm- yyyy'),20000 );
```

User: SYS
Home > SQL > SQL Commands

Autocommit Display 10

```
create table all_teachers (
    name  varchar(50) not null,
    id    int constraint pk_all_teachers primary key,
    domain varchar(10),
    hire_date date,
    balance int
);
```

Results Explain Describe Saved SQL History

Table created.

0.04 seconds

User: SYS
Home > SQL > SQL Commands

Autocommit Display 10

```
CREATE INDEX teacher_idx
ON all_teachers(name,id);
```

Results Explain Describe Saved SQL History

Index created.

0.02 seconds

Language: en-us

127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > **SQL Commands**

Autocommit

```
create sequence all_teachers_id
    increment by 1
    start with 1
    nocycle
    nocache;|
```

Results Explain Describe Saved SQL History

Sequence created.

0.02 seconds

127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > **SQL Commands**

Autocommit

```
insert into all_teachers values('Nazmul Hossain',all_teachers_id.nextval,'CSC',to_date('17-11-2022','dd-mm-yyyy'),30000 );
insert into all_teachers values('kuwsur imron',all_teachers_id.nextval,'CSC',to_date('5-10-2022','dd-mm-yyyy'),30000 );
insert into all_teachers values('nazia alfaz',all_teachers_id.nextval,'CSC',to_date('5-1-2022','dd-mm-yyyy'),50000 );
insert into all_teachers values('tanjil amin',all_teachers_id.nextval,'EEE',to_date('2-8-2021','dd-mm-yyyy'),50000 );
insert into all_teachers values('asif iqbal',all_teachers_id.nextval,'BBA',to_date('2-6-2021','dd-mm-yyyy'),20000 );
```

```
select * from all_teachers;
```

Results Explain Describe Saved SQL History

NAME	ID	DOMAIN	HIRE_DATE	BALANCE
Nazmul Hossain	1	CSC	17-NOV-22	30000
nazia alfaz	2	CSC	05-JAN-22	50000
kuwsur imron	3	CSC	05-OCT-22	30000
tanjil amin	4	EEE	02-AUG-21	50000
asif iqbal	5	BBA	02-JUN-21	20000

5 rows returned in 0.00 seconds [CSV Export](#)

Course Table:

```
create table all_course (
    course_code int constraint pk_all_course primary key, course_name
    varchar(50) not null, course_prerequisite_code varchar(10),
    credit_count int);

create sequence all_course_id
    increment by 1
    start with 1
    nocycle
    nocache;

CREATE INDEX course_idx
    ON all_course(course_name);
```

Insertion:

```
insert into all_course values
(all_course_id.nextval,'Economy',null,3);

insert into all_course values
(all_course_id.nextval,'physics',null,3);

insert into all_course values
(all_course_id.nextval,'computer graphis',null,3);

insert into all_course values
(all_course_id.nextval,'software enginnering',null,3);

insert into all_course values
(all_course_id.nextval,'data base',null,3);
```

← ⏪ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > **SQL Commands**

Autocommit

```
create table all_course
(
    course_code int constraint pk_all_course primary key,
    course_name varchar(50) not null,
    course_prerequisite_code varchar(10),
    credit_count int
);
```

Results Explain Describe Saved SQL History

Table created.

0.03 seconds

← ⏪ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > **SQL Commands**

Autocommit

```
CREATE INDEX course_idx
ON all_course(course_name);
```

Results Explain Describe Saved SQL History

Index created.

0.01 seconds

← ⏪ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
create sequence all_course_id
    increment by 1
    start with 1
    nocycle
    nocache;
```

Results Explain Describe Saved SQL History

Sequence created.

0.00 seconds

← ⏪ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
insert into all_course values(all_course_id.nextval,'Economy',null,3);
insert into all_course values(all_course_id.nextval,'physics',null,3);
insert into all_course values(all_course_id.nextval,'computer graphis',null,3);
insert into all_course values(all_course_id.nextval,'software enginnering',null,3);
insert into all_course values(all_course_id.nextval,'data base',null,3);

select * from all_course;
```

Results Explain Describe Saved SQL History

COURSE_CODE	COURSE_NAME	COURSE_PREREQUISITE_CODE	CREDIT_COUNT
1	Economy	-	3
2	physics	-	3
3	computer graphis	-	3
4	software enginnering	-	3
5	data base	-	3

5 rows returned in 0.00 seconds [CSV Export](#)

OSA Table:

```
create table OSA
( student_name      varchar(50),
  student_id       int constraint fk_id references all_student, osa_serial  int
constraint pk_OSA primary key,
  action           varchar(30),
  cause            varchar(100),
  action_date      date);
```

```
create sequence osa_serial
  increment by 1
  start with 1
  nocycle
```

```
nocache;
```

```
CREATE INDEX
  osa_id
x
ON osa(action);
```

Insertion:

```
insert into osa values
('Minhaz Ayon','1',osa_serial.nextval,'Suspended','Ragging',to_date('19-JUL- 22','dd-mm-yyyy'));
```

```
insert into osa values
('Minhaz      Ayon','1',osa_serial.nextval,'Suspended','Ragging',to_date('19-JUL- 22','dd-mm-yyyy'));
```

```
insert into osa values
('Prokhor Roy','2',osa_serial.nextval,'Suspended','Longoverduepayments',to_date( '19-JUL-22','dd-mm-yyyy'));
```

```
insert into osa values
('moni saha ','3',osa_serial.nextval,'Suspended','Longoverduepayments',to_date('8- jun-22','dd-mm-yyyy'));
```

```
insert into osa values
('afifa islam ','4',osa_serial.nextval,'Suspended','Longoverduepayments',to_date('8- jan-22','dd-mm-yyyy'));
```

127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE Database Express Edition

User: SYS

Home > SQL > SQL Commands

Autocommit Display 10

```
CREATE INDEX osa_idx
ON osa(action);
```

Results Explain Describe Saved SQL History

Index created.

0.01 seconds

127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE Database Express Edition

User: SYS

Home > SQL > SQL Commands

Autocommit Display 10

```
create sequence osa_serial
    increment by 1
    start with 1
    nocycle
    nocache;
```

Results Explain Describe Saved SQL History

Sequence created.

0.01 seconds

← ⏪ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User SYS

Home > SQL > **SQL Commands**

Autocommit Display 10 ▾

```
insert into osa values('Minhaz Ayon','1',osa_serial.nextval,'Suspended','Ragging',to_date('19-JUL-22','dd-mm-yyyy'));
insert into osa values('Minhaz Ayon','1',osa_serial.nextval,'Suspended','Ragging',to_date('19-JUL-22','dd-mm-yyyy'));
insert into osa values('Prokhor Roy','2',osa_serial.nextval,'Suspended','Longoverduepayments',to_date('19-JUL-22','dd-mm-yyyy'));
insert into osa values('moni saha ','3',osa_serial.nextval,'Suspended','Longoverduepayments',to_date('8-jun-22','dd-mm-yyyy'));
insert into osa values('afifa islam ','4',osa_serial.nextval,'Suspended','Longoverduepayments',to_date('8-jan-22','dd-mm-yyyy'));
```

select * from osa;

Results Explain Describe Saved SQL History

STUDENT_NAME	STUDENT_ID	OSA_SERIAL	ACTION	CAUSE	ACTION_DATE
Minhaz Ayon	1	1	Suspended	Ragging	19-JUL-22
Minhaz Ayon	1	2	Suspended	Ragging	19-JUL-22
Prokhor Roy	2	3	Suspended	Longoverduepayments	19-JUL-22
moni saha	3	4	Suspended	Longoverduepayments	08-JUN-22
afifa islam	4	5	Suspended	Longoverduepayments	08-JAN-22

5 rows returned in 0.02 seconds

[CSV Export](#)

Payment Table:

```
create table Payment(
    student_name varchar(50) not null,
    s_id varchar2(8) constraint fk_cid references all_student(id), payment_serial int
        constraint pk_Payment primary key,
    semester      varchar(20),
    year int, payable_amount
    int, amount_paid int,
    payment_date date
);

create sequence pay
    increment by 1
    start with 1
    nocycle
    nocache;

CREATE INDEX pay_idx
    ON payment(semester);
```

Insertion:

```
insert into payment values
( 'Shahriar Zaman', '2', pay.nextval, 'Spring', 2022, 86400, 86400, to_date('19-NOV-21', 'dd-mm-yyyy'));

insert into payment values
( 'Waskiqure', '1', pay.nextval, 'Spring', 2022, 86400, 86400, to_date('19-NOV-21', 'dd-mm-yyyy'));

insert into payment values
( 'samsur nahar', '3', pay.nextval, 'Spring', 2022, 86400, 86400, to_date('22-NOV-21', 'dd-mm-yyyy'));

insert into payment values
( 'Amanullah', '4', pay.nextval, 'Spring', 2022, 86400, 86400, to_date('25-NOV-21', 'dd-mm-yyyy'));

insert into payment values
( 'Zisnara', '5', pay.nextval, 'Spring', 2022, 86400, 86400, to_date('19-NOV-21', 'dd-mm-yyyy'));
```

```
create table Payment(
    student_name varchar(50) not null,
    s_id int references all_student(id),
    payment serial int constraint pk_Payment primary key,
    semester varchar(20),
    year int,
    payable_amount int,
    amount_paid int,
    payment_date date
);
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

Results Explain Describe Saved SQL History

STUDENT_NAME	S_ID	PAYMENT_SERIAL	SEMESTER	YEAR	PAYABLE_AMOUNT	AMOUNT_PAID	PAYMENT_DATE
Shahriar Zaman	1	1	Spring	2022	86400	86400	19-NOV-21
Waskique	1	2	Spring	2022	86400	86400	19-NOV-21
samsur nahar	3	3	Spring	2022	86400	86400	22-NOV-21
Amanullah	4	4	Spring	2022	86400	86400	25-NOV-21
Zisnara	5	5	Spring	2022	86400	86400	19-NOV-21

5 rows returned in 0.01 seconds [CSV Export](#)

Registration Table:

```
create table registration
(
    course_code varchar(10) not null primary key,
    course_name varchar(50) not null,
    course_prerequisite_code varchar(10),
);
```

Insertion:

```
insert into registration values('PHY1102', 'Physics-1 Lab', null);
insert into registration values('ENG1101', 'English-1', null);
insert into registration values('CSC1101', 'ICS', null);
insert into registration values('CSC1102', 'C++', null);
insert into registration values('CSC1103', 'C++ Lab', null);
insert into registration values('CSC1204', 'Discrete Math', 'CSC1102');
insert into registration values('MAT1205', 'Math-2', 'MAT1102');
insert into registration values('CSC1205', 'Java', 'CSC1102');
insert into registration values('MAT1102', 'Math-1', null);
insert into registration values('PHY1101', 'Physics-1', null);
```

```
create table registration
(
course_code varchar(10) not null primary key,
course_name varchar(50) not null,
course_prerequisite_code varchar(10)
);
```

Results Explain Describe Saved SQL History

Table created.

0.22 seconds

```
insert into registration values('PHY1102', 'Physics-1 Lab', null);
insert into registration values('ENG1101', 'English-1', null);
insert into registration values('CSC1101', 'ICS', null);
insert into registration values('CSC1102', 'C++', null);
insert into registration values('CSC1103', 'C++ Lab', null);
insert into registration values('CSC1204', 'Discrete Math', 'CSC1102');
insert into registration values('MAT1205', 'Math-2', 'MAT1102');
insert into registration values('CSC1205', 'Java', 'CSC1102');
insert into registration values('MAT1102', 'Math-1', null);
insert into registration values('PHY1101', 'Physics-1', null);
select*
from registration
```

Results Explain Describe Saved SQL History

COURSE_CODE	COURSE_NAME	COURSE_PREREQUISITE_CODE
PHY1102	Physics-1 Lab	-
ENG1101	English-1	-
CSC1101	ICS	-
CSC1102	C++	-
CSC1103	C++ Lab	-
CSC1204	Discrete Math	CSC1102
MAT1205	Math-2	MAT1102
CSC1205	Java	CSC1102
MAT1102	Math-1	-
PHY1101	Physics-1	-

10 rows returned in 0.06 seconds [CSV Export](#)

CSRT Table:

```
create table CSRT (
    serial int constraint pk_CSRT primary key,
    course_code int references all_course(course_code), course_name      varchar(50),
    teacher_id   int references all_teachers(ID), teacher_name varchar(50),
    student_id   int not null references all_student(id), student_name varchar(50),
    marks int
);
create sequence cst
    increment by 1
    start with 1
    nocycle
    nocache;
```

Insertion:

insert into CSRT values

(1, 1, 'economy', 1, 'Nazmul Hossain', 1, 'waskique', 90);

insert into CSRT values

(2, 2, 'physics', 2, 'Najia Alfaz', 2, 'shahriar zaman', 80);

insert into CSRT values

(3, 3, 'computer graphics', 3, 'kuwsur imran', 3, 'sumsun nahar', 85);

insert into CSRT values

(4, 4, 'software engineering', 4, 'tanjil amin', 4, 'amanullah', 72);

insert into CSRT values

(5, 5, 'data base', 5, 'asif iqbal', 5, 'zisnara', 96);

```

create table CSRT (
    serial int constraint pk_CSRT primary key,
    course_code int references all_course(course_code), course_name varchar(50),
    teacher_id int references all_teachers(ID), teacher_name varchar(50),
    student_id int not null references all_student(id), student_name varchar(50),
    marks int
);

```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Table created.

0.09 seconds

```

insert into CSRT values
(1, 1, 'economy', 1, 'Nazmul Hossain', 1, 'waskiure', 90);
insert into CSRT values
(2, 2, 'physics', 2, 'Najia Alfaz', 2, 'shahriar zaman', 80);
insert into CSRT values
(3, 3, 'computer graphics', 3, 'kuwsur imran', 3, 'sumsun nahar', 85);
insert into CSRT values
(4, 4, 'software engineering', 4, 'tanjil amin', 4, 'amanullah', 72);
insert into CSRT values
(5, 5, 'data base', 5, 'asif iqbal', 5, 'zisnara', 96);

```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

SERIAL	COURSE_CODE	COURSE_NAME	TEACHER_ID	TEACHER_NAME	STUDENT_ID	STUDENT_NAME	MARKS
1	1	economy	1	Nazmul Hossain	1	waskiure	90
2	2	physics	2	Najia Alfaz	2	shahriar zaman	80
3	3	computer graphics	3	kuwsur imran	3	sumsun nahar	85
4	4	software engineering	4	tanjil amin	4	amanullah	72
5	5	data base	5	asif iqbal	5	zisnara	96

5 rows returned in 0.00 seconds [CSV Export](#)

Ankurita, Firefox 24

Query Writing (Write down the question and also the answer)

Single-row function

1. Display student name,id and program for program eee.

```
SELECT name,id,program  
      FROM all_student  
     WHERE LOWER(program) = 'eee';
```

The screenshot shows a MySQL command-line interface. The user has entered the following SQL query:

```
SELECT name,id,program FROM all_student  
WHERE LOWER(program) = 'eee';
```

Below the query, the results are displayed in a table:

NAME	ID	PROGRAM
Waskique	1	EEE
sumsur nahar	3	EEE

Text at the bottom of the interface indicates "2 rows returned in 0.14 seconds" and "CSV Export".

2. Find the length for name nazia alfaz from teacher table

```
select length(name) from all_teachers where name='nazia alfaz';
```

The screenshot shows a MySQL command-line interface. The user has entered the following SQL query:

```
SELECT name,id,program FROM all_student  
WHERE LOWER(program) = 'eee';  
  
select length(name) from all_teachers where name='nazia alfaz';
```

Below the query, the results are displayed in a table:

LENGTH(NAME)
11

Text at the bottom of the interface indicates "1 rows returned in 0.01 seconds" and "CSV Export".

3. concat name,program as nameprogram from all_student

select concat(name,program) as nameprogram from all_student

```
SELECT name,id,program FROM      all_student
WHERE  LOWER(program) = 'eee';

select length(name) from all_teachers where name='nazia alfaz';

select concat(name,program) as nameprogram from all_student
```

Results Explain Describe Saved SQL History

NAMEPROGRAM
WaskiqueEEE
Shahriar ZamanCSE
sunsur naharEEE
AMANULLAHCSE
ZisnaraCSE

5 rows returned in 0.02 seconds CSV Export

Group-row function

1. Find maximum salary for teacher which domain is csc

select max(balance) from all_teachers where domain like'CSC%'

2. How many row for cse program in all_student table

select count(*) from all_student where program='CSE'

3. Find the minimum and maximum hire_date from all_teachers table

select min(hire_date),max(hire_date) from all_teachers

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:2702549379414260::NO:::. The SQL command entered is:

```
select max(balance) from all_teachers where domain like'CSC%'
```

The results section shows a single row:

MAX(BALANCE)
50000

1 rows returned in 0.01 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:2702549379414260::NO:::. The SQL commands entered are:

```
select max(balance) from all_teachers where domain like'CSC%'  
select count(*) from all_student where program='CSE'
```

The results section shows two rows:

COUNT(*)
3

1 rows returned in 0.02 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

SQL Commands +

127.0.0.1:8080/apex/f?p=4500:1003:2702549379414260::NO::

ORACLE Database Express Edition

User SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
select max(balance) from all_teachers where domain like'CSC%'  
select count(*) from all_student where program='CSE'  
select min(hire_date),max(hire_date) from all_teachers
```

Results Explain Describe Saved SQL History

MIN(HIRE_DATE)	MAX(HIRE_DATE)
02-JUN-21	17-NOV-22

1 rows returned in 0.02 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

MACHINE

12:45 AM 12/27/2022

The screenshot shows the Oracle Database Express Edition SQL Commands interface. At the top, there's a toolbar with various icons. Below it is a navigation bar with 'User SYSTEM' and 'Home > SQL > SQL Commands'. The main area contains three SQL statements: 'select max(balance) from all_teachers where domain like'CSC%'' (with a result of 1 row), 'select count(*) from all_student where program='CSE'' (with a result of 1 row), and 'select min(hire_date),max(hire_date) from all_teachers' (with a result of 1 row). Below the SQL area is a 'Results' tab, followed by 'Explain', 'Describe', 'Saved SQL', and 'History'. A table displays the results of the third query. At the bottom, there's a language setting 'Language: en-us', a copyright notice for Oracle, and a Windows taskbar with various application icons. The date and time '12/27/2022 12:45 AM' are also visible.

Subquery

1. Which teachers have a salary greater than 'asif iqbal' salary?

```
select name,balance from all_teachers where balance>(select balance from all_teachers where name='asif iqbal')
```

The screenshot shows a MySQL command-line interface. At the top, there is a SQL query window containing the subquery. Below it is a results table with two columns: NAME and BALANCE. The results table contains five rows of data. At the bottom, there is some status information and a CSV export link.

NAME	BALANCE
Nazmul Hossain	30000
kuwsur imron	30000
nazia alfaz	50000
tanjil amin	50000

4 rows returned in 0.20 seconds [CSV Export](#)

2. which student have a cgpa less than 'waskiquire' cgpa?

```
select name,cgpa from all_student where cgpa<(select cgpa from all_student where name='Waskiquire')
```

The screenshot shows a MySQL command-line interface. At the top, there are two separate SQL queries. Below them is a results table with two columns: NAME and CGPA. The results table contains two rows of data. At the bottom, there is some status information and a CSV export link.

NAME	CGPA
sumsur nahar	3.25
AMANULLAH	2.98

2 rows returned in 0.06 seconds [CSV Export](#)

3. display name who join after nazia alfaz

```
select name from all_teachers where hire_date > (select hire_date from all_teachers where name='nazia alfaz')
```

The screenshot shows a MySQL query editor window. At the top, there is a toolbar with buttons for Autocommit, Display, and Save. Below the toolbar, the SQL query is displayed:

```
select name, balance from all_teachers where balance > (select balance from all_teachers where name='asif iqbal')
select name, cgpa from all_student where cgpa < (select cgpa from all_student where name='Waskiure')
select name from all_teachers where hire_date > (select hire_date from all_teachers where name='nazia alfaz')
```

Below the query, the results are shown in a table format:

NAME
Nazmul Hossain
kuvsur imron

At the bottom of the results pane, it says "2 rows returned in 0.01 seconds" and has a "CSV Export" link.

Joining

1. dispaly name,id,payable_amount of each student

```
select s.name,s.id,p.payable_amount from all_student s ,payment p where s.id=p.s_id
```

2. Write a query to display the name, payment_serial, semester and payment_date for all student.

```
select s.name,p.payment_serial,p.semester,p.payment_date from all_student s ,payment p where s.id=p.s_id
```

Synonym

1. Create a shortened name for the all_student view

```
CREATE SYNONYM student1  
For all_student;
```

The screenshot shows a database interface with a dark theme. At the top, there is a code editor containing the SQL command to create a synonym. Below the code editor is a toolbar with several buttons: 'Results' (which is underlined in white), 'Explain', 'Describe', 'Saved SQL', and 'History'. The main area below the toolbar displays the results of the query. It shows the message 'Synonym created.' and a execution time of '0.18 seconds'.

```
CREATE SYNONYM student1 For all_student;
```

Results Explain Describe Saved SQL History

Synonym created.
0.18 seconds

2. Create a shortened name for the all_teachers view

```
CREATE SYNONYM  
teacher For all_teachers;
```

The screenshot shows a database interface with a dark theme. At the top, there is a code editor containing two SQL commands to create synonyms. Below the code editor is a toolbar with several buttons: 'Results' (which is underlined in white), 'Explain', 'Describe', 'Saved SQL', and 'History'. The main area below the toolbar displays the results of the query. It shows the message 'Synonym created.' and a execution time of '0.04 seconds'.

```
CREATE SYNONYM student1 For all_student;  
CREATE SYNONYM teacher For all_teachers;
```

Results Explain Describe Saved SQL History

Synonym created.
0.04 seconds

3 . Create a shortened name for the all_course view

```
CREATE SYNONYM course  
For all_course;
```

```
CREATE SYNONYM student1 For all_student;  
CREATE SYNONYM teacher For all_teachers;  
CREATE SYNONYM course For all_course;
```

Results Explain Describe Saved SQL History

Synonym created.

0.02 seconds

PL/SQL

Function

1.Create a function that returns the total number of student.

```
CREATE OR REPLACE FUNCTION totalstudent
```

```
RETURN number AS
```

```
total number(2) := 0;
```

```
BEGIN
```

```
    SELECT count(*) into total
```

```
    FROM all_student;
```

```
    RETURN total;
```

```
END;
```

```
/
```

```
DECLARE
```

```
    c number(2);
```

```
BEGIN
```

```
    c := totalstudent();
```

```
    dbms_output.put_line('Total no. of student: ' || c);
```

```
END; /
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
CREATE OR REPLACE FUNCTION totalstudent
RETURN number AS
total number(2) := 0;
BEGIN
    SELECT count(*) into total
    FROM all_student;
    RETURN total;
END;
/
select * from dept;
DECLARE
    c number(2);
BEGIN
    c := totalstudent();
    dbms_output.put_line('Total no. of student: ' || c);
END;
/
```

The results section shows:

```
Total no. of student: 5
Statement processed.
```

Execution time: 0.05 seconds

2.Create a function that returns the total number of teacher.

```
CREATE OR REPLACE FUNCTION totalteacher
```

```
RETURN number AS
```

```
    total number(2) := 0;
```

```
BEGIN
```

```
    SELECT count(*) into total
```

```
    FROM all_teachers;
```

```
    RETURN total;
```

```
END;
```

```
/
```

```
DECLARE
```

```
    c number(2);
```

```
BEGIN
```

```
    c := totalteacher();
```

```
    dbms_output.put_line('Total no. of teacher: ' || c);
```

```
END;
```

```
/
```

```
DECLARE
    c number(2);
BEGIN
    c := totalteacher();
    dbms_output.put_line('Total no. of teacher: ' || c);
END;
/
```

Total no. of teacher: 5
Statement processed.
0.01 seconds

3.Create a function that returns the total number of suspended student.

```
CREATE OR REPLACE FUNCTION totalstudent
```

```
RETURN number AS
```

```
    total number(2) := 0;
```

```
BEGIN
```

```
    SELECT count(*) into total
```

```
    FROM osa where action='Suspended';
```

```
    RETURN total;
```

```
END;
```

```
/
```

```
DECLARE
```

```
    c number(2);
```

```
BEGIN
```

```
    c := totalstudent();
```

```
    dbms_output.put_line('Total no. of suspended student: ' || c);
```

```
END;
```

```
/
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window displays the following PL/SQL block:

```
CREATE OR REPLACE FUNCTION totalstudent
RETURN number AS
    total number(2) := 0;
BEGIN
    SELECT count(*) into total
    FROM osa where action='Suspended';
    RETURN total;
END;
/
DECLARE
    c number(2);
BEGIN
    c := totalstudent();
    dbms_output.put_line('Total no. of suspended student: ' || c);
END;
/
```

Below the command window, the results are shown:

```
Total no. of suspended student: 5
Statement processed.

0.03 seconds
```

Record:

1.Create a record that can output the id, name, cgpa, and admission date of the student whose Id is 4

declare

stu_rec all_student%rowtype;

begin

select * into stu_rec from all_student

where id=4;

dbms_output.put_line(stu_rec.id||' '| stu_rec.name||' '| stu_rec.cgpa||' '| stu_rec.admission_date);

end

/

The screenshot shows the Oracle Database Express Edition interface. The URL in the address bar is 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO:::. The user is SYS. The SQL Commands page is displayed, showing the following PL/SQL code in the editor:

```
declare
  stu_rec all_student%rowtype;
begin
  select * into stu_rec from all_student
  where id=4;
  dbms_output.put_line(stu_rec.id||' '| stu_rec.name||' '| stu_rec.cgpa||' '| stu_rec.admission_date);
end
/
```

The code is executed successfully, with the output:

```
4 AMANULLAH 2.98 19-DEC-21
Statement processed.
```

Execution time: 0.00 seconds

2.Create a record that can output the id, name, domain, and hire date of the teacher whose Id is 2

```
declare
teacher_rec all_teachers%rowtype;
begin
select * into teacher_rec from all_teachers
where id=2;
dbms_output.put_line(teacher_rec.id||' '||teacher_rec.name||' '||teacher_rec.domain||
'||teacher_rec.hire_date);
end
/
```



The screenshot shows the Oracle Database Express Edition interface. The URL in the address bar is 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::. The title bar says "ORACLE Database Express Edition". The user is "User SYS". The navigation bar shows "Home > SQL > SQL Commands". The SQL editor contains the following PL/SQL code:

```
declare
teacher_rec all_teachers%rowtype;
begin
select * into teacher_rec from all_teachers
where id=2;
dbms_output.put_line(teacher_rec.id||' '||teacher_rec.name||' '||teacher_rec.domain||' '||teacher_rec.hire_date);
end
/
```

Below the editor, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Results" tab is selected. The output pane shows the results of the query:

```
2 nazia alfaz CSC 05-JAN-22
Statement processed.

0.01 seconds
```

3.Create a record that can output the id, name, domain, and hire date of the course whose Id is 2

```
declare
b all_course%rowtype;
begin
select * into b from all_course
where course_code=2;
dbms_output.put_line(b.course_code||' '||b.course_name||' '||b.credit_count);
end
/
```

The screenshot shows the Oracle Database Express Edition interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::. The user is SYS. The SQL Commands page is displayed. The code entered is:

```
declare
b all_course%rowtype;
begin
select * into b from all_course
where course_code=2;
dbms_output.put_line(b.course_code||' '||b.course_name||' '||b.credit_count);
end
/
```

The results show the output of the dbms_output.put_line statement:

```
2 physics 3
```

Statement processed.

0.02 seconds

Package:

- 1.Create a package that contains a procedure which can display the name,cgpa whose id is passed as its parameter**

```
CREATE OR REPLACE PACKAGE BODY stu_pack AS  
  PROCEDURE display_name(e all_student.id%TYPE) IS  
    name all_student.id%TYPE;  
  BEGIN  
    SELECT name INTO name  
    FROM all_student  
    WHERE id=e;  
    dbms_output.put_line('student Name: '|| name);  
  END display_name;  
  
  PROCEDURE display_cgpa(e1 all_student.id%TYPE) IS  
    cgpa all_student.id%TYPE;  
  BEGIN  
    SELECT cgpa INTO cgpa  
    FROM all_student  
    WHERE id=e1;  
    dbms_output.put_line('student cgpa: '|| cgpa);  
  END display_cgpa;  
  
END stu_pack;  
/
```

```
CREATE OR REPLACE PACKAGE stu_pack AS  
  PROCEDURE display_name(e all_student.id%type);  
  PROCEDURE display_cgpa(e1 all_student.id%type);  
END stu_pack;
```

```

begin
stu_pack.display_name('2');
stu_pack.display_cgpa('2');

```

end

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```

CREATE OR REPLACE PACKAGE stu_pack AS
  PROCEDURE display_name(e all_student.id%TYPE);
  PROCEDURE display_cgpa(e1 all_student.id%TYPE);
END stu_pack;

```

The results show the package was created successfully.

Package created.

0.20 seconds

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```

CREATE OR REPLACE PACKAGE BODY stu_pack AS
  PROCEDURE display_name(e all_student.id%TYPE) IS
    name all_student.id%TYPE;
  BEGIN
    SELECT name INTO name
    FROM all_student
    WHERE id=e;
    dbms_output.put_line('student Name: '|| name);
  END display_name;
  PROCEDURE display_cgpa(e1 all_student.id%TYPE) IS
    cgpa all_student.id%TYPE;
  BEGIN
    SELECT cgpa INTO cgpa
    FROM all_student
    WHERE id=e1;
    dbms_output.put_line('student cgpa: '|| cgpa);
  END display_cgpa;

```

The results show the package body was created successfully.

Package Body created.

0.12 seconds

2.Create a package that contains a procedure which can display the name, domain whose id is passed as its parameter from teacher table

```
CREATE OR REPLACE PACKAGE BODY tea_pack AS  
  PROCEDURE display_name(e all_teachers.id%TYPE) IS  
    name all_teachers.name%TYPE;  
  BEGIN  
    SELECT name INTO name  
    FROM all_teachers  
    WHERE id=e;  
    dbms_output.put_line('teachers Name: '|| name);  
  END display_name;  
  
  PROCEDURE display_domain(e1 all_teachers.id%TYPE) IS  
    domain all_teachers.domain%TYPE;  
  BEGIN  
    SELECT domain INTO domain  
    FROM all_teachers  
    WHERE id=e1;  
    dbms_output.put_line('teachers domain: '|| domain);  
  END display_domain;  
  
END tea_pack;  
/  
CREATE OR REPLACE PACKAGE tea_pack AS  
  PROCEDURE display_name(e all_teachers.id%type);  
  PROCEDURE display_domain(e1 all_teachers.id%type);  
  
END tea_pack;
```

```

select * from all_teachers
begin
tea_pack.display_name('2');
tea_pack.display_domain('2');
end

```

← ⌂ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > SQL Commands

Autocommit Display 15

```

END stu_pack;

begin
stu_pack.display_name('2');
stu_pack.display_cgpa('2');
end

```

Results Explain Describe Saved SQL History

```

student Name: Shahriar Zaman
student cgpa: 3.84

Statement processed.

0.06 seconds

```

← ⌂ ⓘ 127.0.0.1:8080/apex/f?p=4500:1003:1606051144458873::NO::

ORACLE® Database Express Edition

User: SYS

Home > SQL > SQL Commands

Autocommit Display 15

```

CREATE OR REPLACE PACKAGE tea_pack AS
  PROCEDURE display_name(e all_teachers.id%type);
  PROCEDURE display_domain(e1 all_teachers.id%type);
END tea_pack;
select * from all_teachers
begin
tea_pack.display_name('2');
tea_pack.display_domain('2');
end

```

Results Explain Describe Saved SQL History

```

teachers Name: nazia alfaz
teachers domain: CSC

Statement processed.

0.09 seconds

```

Trigger:

1. Display a trigger that give reminder when banlance is 0.

create or replace trigger expenditure_recorder

before update on all_teachers

for each row

begin

update this_semester_report set expenditure=expenditure+(:new.balance-:old.balance);

end;

```
CREATE OR REPLACE TRIGGER checkBalance
BEFORE INSERT OR UPDATE OF balance ON all_teachers
FOR EACH ROW
WHEN (new.balance<0)
BEGIN
raise application_error(-20000, 'Balance can not be Zero');
END;

Results Explain Describe Saved SQL History

Trigger created.

0.27 seconds
```

```
CREATE OR REPLACE TRIGGER checkBalance
BEFORE INSERT OR UPDATE OF balance ON all_teachers
FOR EACH ROW
WHEN (new.balance<0)
BEGIN
raise application_error(-20000, 'Balance can not be Zero');
END;
update all_teachers set balance=-20000;

Results Explain Describe Saved SQL History

ORA-20000: Balance can not be Zero
ORA-06512: at "SYSTEM.CHECKBALANCE", line 2
ORA-04088: error during execution of trigger 'SYSTEM.CHECKBALANCE'
1. update all_teachers set balance=-20000;

0.02 seconds
```

2. CGPA is between 0 to 4

create or replace trigger marks_value_fixer

before update of marks on csrt

for each row

begin

if :new.marks>100 then

:new.marks:=100;

elsif :new.marks<0 then

:new.marks:=0;

end if;

end;

```
CREATE OR REPLACE TRIGGER check_credits_completed
BEFORE INSERT OR UPDATE OF credits_completed ON all_student
FOR EACH ROW
WHEN (new.credits_completed <0 OR new.credits_completed>148)
BEGIN
raise_application_error(-20002, 'Credits completed has to be from 0 to 148');
END;
```

Results Explain Describe Saved SQL History

Trigger created.

0.02 seconds

```
CREATE OR REPLACE TRIGGER checkcgpa
BEFORE INSERT OR UPDATE OF cgpa ON all_student
FOR EACH ROW
WHEN (new.cgpa<0 OR new.cgpa>4)
BEGIN
raise_application_error(-20001, 'CGPA has to be from 0 to 4');
END;
update all_student set cgpa=5;
```

Results Explain Describe Saved SQL History

```
ORA-20001: CGPA has to be from 0 to 4
ORA-06512: at "SYSTEM.CHECKCGPA", line 2
ORA-04088: error during execution of trigger 'SYSTEM.CHECKCGPA'
1. update all_student set cgpa=5;
```

0.02 seconds

3. If credit complete is lower or higher than 0 and 148.

create or replace trigger result_log

after update on csrt

begin

```
insert into task_log values((select id from loggedinTable),'Updating  
results',sysdate,SYSTIMESTAMP);
```

end;

The screenshot shows the SQL code for creating a trigger named 'checkcgpa'. The trigger is defined as follows:

```
CREATE OR REPLACE TRIGGER checkcgpa  
BEFORE INSERT OR UPDATE OF cgpa ON all_student  
FOR EACH ROW  
WHEN (new.cgpa<0 OR new.cgpa>4)  
BEGIN  
raise application_error(-20001, 'CGPA has to be from 0 to 4');  
END;
```

Below the code, the results pane shows the message "Trigger created." and a execution time of "0.02 seconds".

The screenshot shows the SQL code for creating a trigger named 'check_credits_completed'. The trigger is defined as follows:

```
CREATE OR REPLACE TRIGGER check_credits_completed  
BEFORE INSERT OR UPDATE OF credits_completed ON all_student  
FOR EACH ROW  
WHEN (new.credits_completed <0 OR new.credits_completed>148)  
BEGIN  
raise application_error(-20002, 'Credits completed has to be from 0 to 148');  
END;  
update all_student set credits_completed = 200;
```

Below the code, the results pane shows an error message in a callout box:

```
ORA-20002: Credits completed has to be from 0 to 148  
ORA-06512: at "SYSTEM.CHECK_CREDITS_COMPLETED", line 2  
ORA-04088: error during execution of trigger 'SYSTEM.CHECK_CREDITS_COMPLETED'  
1. update all_student set credits_completed = 200;
```

At the bottom, it shows an execution time of "0.00 seconds".

Conclusion: In our project, there might be some minor problem. We will try to configure them if needed and we will provide more correct database for the software that we are going to done. In final term we implemented the PL/SQL in the project which we have implemented in our mid term. Here, we cannot provide a fully connected user interface for some sort of reason but we provide a interface which is also looks like same which we planed. A project always need to be upgrade. If change needed we can do that. There will be more changes in our project by following our upcoming plans if needed. We try to make more data to entry in our project and give a perfect output.