GENERAL FEATURES –

1. **LOGIN PAGE**
2. **Edit Profile**
3. **See syllabus – for students – My syllabus**
4. **For teachers – see syllabus of his Department**
5. General notices

ADMIN –

1. Can add and delete new courses ( extremely large process ) [ alpha phase] [V2 maybe]
2. Can add and delete new subjects [ reference -> if the subject is still in use -> don’t allow deletion][ using reference key ]
3. To make this useful -> we have to give an option to choose papers,  
   so subject can be deleted and added.  
   [ LIMIT THIS – Don’t give this power -> lot of handling ]
4. Change subjects of semesters.
5. **Confirm awaiting students and waiting teachers**BUT HOW,
   * 1. A separate table for waiting students, -> easy to maintain/no redundancy/  
        saves query complexity
     2. One bit column to say if the student is yet confirmed or not,  
        each query will have to check if the students are confirmed or not,  
        one redundant bit forever.

Teachers –

1. **View self-details.**
2. **View student details – all students**
3. **Edit profile**
4. Can upload attendance

STUDENTS –

1. **View self-details**
2. **Register using Roll No and details (Course as well) [year as well] & wait for admin confirmation.**
3. can have up to 3 subjects

TABLES –

1. **Login table**- Login credentials – Email, password Role – [Admin, Teacher, Student]
2. **Teachers Details** – Name, contact, email, Department, Description , AND DEPARTMENT  
   As such NO table is required for department for now.
3. **~~DEPARTMENT~~** ~~– CS, physics, maths, electronics.  
   dept code, dept name, Major course, other courses-> text array~~

~~--- NOT VERY RELEVANT TABLE ---~~

1. TEMP TABLE \_\_ AWAITING MEMBERS TABLE- IF ANY [ STUDENTS, TEACHERS]
2. **STUDENT details** 
   1. FIELDS
      1. Name, DOB, Email, Contact no,
      2. Course – ONE department has only one course
      3. ~~Date of admission – to calculate year of the student.~~
      4. Semester no – to store the current semester no
      5. ROLL.NO – if student provides his rollono during registration,   
         it’ll save course/year/sem entry
      6. ALSO – if feasible - generate a stored procedure to delete students who goes beyond third year
3. **Courses** – addition and deletion of courses by admin only. - ---- V2.  
   course id, course name, department  
   STATIC TABLE In V1.
   1. When a course is deleted – delete all semesters.
   2. When a course is added – WARNING – adding of courses requires developer authority. BLAH BLAH LONG PROCESS.   
      give an option to configure subject and add 2 semesters, 3 subjects in each semester.
4. Maintain a DUAL with have current server time  
   if server time and date<system clock, update it to current system date
5. Semesterinfo - course + semester makes a batch
6. **Batchinfo –** [PK: batch id]   
   (1 initially) Course + 6 semester + 2 subjects.
   1. Course
   2. Semester
   3. Subject 1
   4. Attendance\_subject1
   5. Marks\_subject1
   6. … … .. upto subject 4
   7. Teacher id
7. **Subjects** –
   1. At least 30 subjects with subject codes and description.
   2. Subcode , subject name, syllabus [ link to text file ]
   3. ADMIN can ADD SUBJECTS
8. **Notices - Admin can upload notices**
9. **Modification record – to store last modified time of each table**

**Make a trigger, or a procedure, whatever to fill date  
filling it by user code is a little illogical**

* + - * **ADDED CURRENT TIMESTAMP AS DEFAULT**
      * **- check whether this timestamp is compatible with my date time**
      * **-- RECHECK**

1. **CREATE A**  FILES **TABLE**

**WHICH STORES FILE ID, FILE NAME AND EXACT PATH**

**AND THEN GIVE LINKS TO THAT TABLE FROM EVERYWHERE ELSE**

**LET THE COMMON TYPES BE –**

1. **Name- 64**
2. **Username- 32**
3. **Email 256**
4. **Password – 32**
5. **Priviledge – 32**
6. **Contact – 32**
7. **Doj – dob- date**
8. **Description – 1024**
9. **Department name- 128- course name**
10. **Course code – 8**
11. **Roll no – 16**

**## ALL FOREIGN KEYS ARE CASCADED OR SET NULL OR RESTRICT**

**If assumed meaning is wrong, update all cascading commands**

**FK s-**

alumni - not req

batchinfo - 1 done

course --------- 1 done

department ---- not req

files -------- not req

mod recoed ---- not req

ntoice --------- 1 done

sem info - done

student - 1 done

subject - not rerq

suinfb1 ---- 5 done

sub2 -------- 5 done

teacher ------------ 2 doen

user -------------- not req

MODULES –

1. ADMISSION
2. //ATTENDANCE
3. ACADEMIC - marks

FEATURES IN THE TESLA:

1. Admin Login / teacher login / student login
2. Student is in a course, have 4 subjects.  
   Subjects are pre-filled semester wise.
3. Teacher can choose a subject for a particular semester to teach which subject to teach, 2 subjects.  
   Once chosen, subjects can’t be changed.   
   choose subjects for this sem, 1 and 2 ->   
   can teach only 1 subject to one course class
4. We have a total of 6 semesters, at once, only even or odd semester is active.  
   Admin can activate a semester-   
   1,3,5 AND 2,4,6

LIMITING THE CURRENT MODEL –

1. Only 2 subjects per semester [ out of 4 ]
2. 1 course – CS [ multiple can be added ]
3. As only one course – so admin can’t add/ delete in v1
4. All students are passes in all subjects, backs are not handled.

NOTES:

1. Subject&Semester table is pre-filled and static
2. Strict boundary on date of admission,  
   15 June to 15 august – current year to currYear-3, If don’t remember date, choose your current year,  
   default admission date – 1 July will be auto-filled.

SCREENS:

1. Student --  
   all set, semester will be displayed,  
   then subjects accordingly-> then attendance and marks TAB accordingly.
2. Teachers –

Department – matched form teacher’s table, shown only those subjects and semesters. [ currently only CS ]  
which subjects do you wanna teach this semester – 2 subjects asked.  
year wise dropdown list of subjects

These subjects must not of those of as chosen by other teachers

1. ADMIN PANEL:

A tab for Date managaement

1. Finish the sem & start a new one  
   set sysclock to start of next sem and give notification that teachers should now choose their subjects.  
   All students must be promoted to next sem, how ??

This will be automatic as sem is obtained from DOA and sys clock comparison.  
Problem: Students cant be notified that they have been passed. –do

TODO:

1. Think about end of semester
2. Promotion of students and students pass out
3. ADMIN – can finish a semester by pressing button,  
   that’ll promote all the students to sem+1 and reset subject for teachers etc.
   1. Set restriction on button pressing,  
       can only be pressed if server time is ->  
      for even sem-> 15 june – 15 july

For odd sem -> 25 november - 25 December.

AS IT IS IN AUTOMATED SYSTEM, TRY TO REMOVE THIS BUTTON

MAKE IT AUTOMATIC

1. COMPLEX SHIT

A STORED SQL PROCEDURE –to auto update sem and reset surbjects based on server time. [maybe v2]

Problem with this architecture -> if server time goes to future [ sem updates ], if again comes back ?? , this case can be handled by logic in the stored procedure] [decrement sems according to time frame].

For now,it is easier to do the THE FINISH BUTTON concept of admin.  
IF students pass out, delte them.  
SO BETTER WILL BE USING internet time services. To set coordination.  
but db won’t be updated until connected to internet,  
INTRANET FEATURE GONE. -> so in and all -> save it for later implementation.

1. GENERATE DEPENDENCIES ->
   1. reference -> if the subject is still in use -> don’t allow deletion]
   2. In front of subject, show -> NO Teacher allotted yet
   3. When registering any new members, put his details in LOGIN table as well

TODO:

1. Generate a trigger to move students to alumni table
2. Also think about procedures to clean up database – after new sem
3. Identify functional dependencies and references - DONEE
4. Implement references - DONE
5. Optimize table to Normal FORM – DONE

TRIGGERS USED:

1. **TRIGGER FOR SEM UPDATE AND FLUSH SUBJECTS ON DATE jump – MOST COMPELX –**

Maybe create a stored procedure and call at the end of every semester-

Static timings for semester start and end needs to be defined. BE CAREFUL.

1. TRIGGER FOR MODIFICATION RECORD – when any table is updated- DONE
2. Only useful tables for FLUSHING are:
   1. Subject1 & Subject2
   2. In these only, all the attendance, and teacher related semester data is kept
   3. So storing modification of these two tables will be enough
3. ~~Only storing last addition date is enough.~~

NOT enough- AS ADDITION WILL TAKE ONLY INITIALLY, THEN ALL TIMES, IT’LL BE FLUSHED AND REFILLED. TO BE ON SAFER SIDE, I am holding time of , INSERT & UPADTE

* 1. Case 1: I assign subjects now-> date modified set to now  
     then JUMP to next semester  
     both tables cleared, before adding I can verify that today’s date is in future of the last\_modified\_date.
  2. Case 2: I assign subjects now, And JUMP to past,  
     we’ll see that current < last mod date,  
     so we leave the tables as it is.

1. A TRIGGER MADE FOR ALUMNI on semester update- DONE and delete this old student from student table – write a stored procedure for this, which can be called from java bean.  
   stored procedure made for this is: **remove\_old\_Students()**
2. **ONE MORE TRIGGER IS REQUIRED**

stored procedure made now just call it from admin panel

1. When a COURSE is created, make 6 semesters for it.
2. For all those 6 semesters, create batch id
3. For each batch id, make an entry in batch info table
4. For each of this batch id, make entry in all subject tables. As each batch must teach some subjects at least.
5. STORED PROCEDURE - **flush\_after\_Semester()**

SUGGESTED TRIGGER IS:

CREATE DEFINER=`root`@`localhost` TRIGGER `tesla`.`batchinfo\_AFTER\_INSERT` AFTER INSERT ON `batchinfo` FOR EACH ROW

BEGIN

DECLARE subSTRING varchar(10) default 'subject';

DECLARE i int default 1;

DECLARE t bool default true;

DECLARE subjectTABLE varchar(10);

declare myquery varchar(500);

SET subjectTABLE= CONCAT(subSTRING,cast(i as char));

WHILE(t) DO

IF EXISTS( SELECT \* FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_SCHEMA = 'tesla' AND TABLE\_NAME = subjectTABLE)

THEN

SET myquery = 'INSERT INTO '+subjectTABLE+' (batch\_id) VALUES(NEW.batch\_id)';

execute myquery;

SET i=i+1;

ELSE

SET t=false;

END IF;

END WHILE;

END

**For this point 4: Note that**

There is no good workaround for the absense of Dynamic SQL in MySQL functions, just klunky cludges. Some things still remain downright impossible to cludge, such as using a dynamically-calculated field name or table name in a SQL query. Yes, once in a while there is a need for doing this sort of thing!And don't try cheat by putting the Dynamic SQL in a stored procedure and wrapping in a function or trigger, as the question poser tried - MySQL is too clever and will give you the usual obscure error message. Believe me, I have been around all the houses.

Coming from an Oracle PL/SQL and MS SQL Server background, I sorely miss the richness that PL/SQL and (to a small extent) T-SQL offers for writing procedural SQL.

**SO, we cannot make entry of batch id in ALL subject tables at once.**

**We need to duplicate lines for each subject table.**

**-- DONE**

**FOR now,** I have added two lines to make entry into sub1 and sub2 tables

IF MORE SUBJECTS ARE ADDED IN FUTURE,  
 RE-MODIFY THIS TRIGGER.

**NOTE THAT:**

1. A department can exist without course
2. A course create must have 6 semester
3. **when cascaded triggers are not fired in mysql, so take care of that, if problem occurs**
4. CREATE DEFINER=`root`@`localhost` TRIGGER `tesla`.`student\_AFTER\_UPDATE` AFTER UPDATE ON `student` FOR EACH ROW

BEGIN

IF(NEW.semester>3)

THEN

INSERT INTO alumni(alumni\_id,name,dob,email,contact,rollno)

select NEW.student\_id,NEW.name,NEW.dob,NEW.email,NEW.contact,NEW.rollno;

END IF;

END

# ALSO CREATE SOME VIEWS TO SOLVE THE TASK OF MULTIPLE QUERIES

Queries:

1. Display subject 2 of BSC hons 1 sem

NOT SUPPORTED:

1. Upgradation of course to btech and cbcs etc

RESTRICTIONS TO IMPOSE:

1. A TEACHER can teach only 2 subjects in one sem and they should be to two diff semesters

THINGS TO FIGURE OUT:

1. AT a given time, only even OR odd sem is active, so different batch id’s really necessary?
2. Batch id’s must be unique according to each year batch, but that’ll be just BATCH\_YEAR
3. Admin can add and remove subjects to batches  
   Theory+ lab- **DONE**
4. AFTER EACH SEM, Flush this batchsub table.**- DONE**

HOW WILL I IDENTIFY 2014 BATCH OF cs HONS

SOL – Doa WILL BE 2014 AND ROLL NO WILL HAVE hCS

Pages in dashboard –

1. Profile
2. Logout
3. Dashboard
4. Attendance
5. Result
6. Syllabus
7. Notices

Queries required –

to display available subject to teachers -

SELECT CONCAT(subject\_code,subject\_name) FROM tesla.subject where subject\_id IN

( SELECT subject\_id FROM tesla.batchsub where teacher\_id is null) ;

to check if subjects has been already assigned to teacher or not

SELECT \* FROM tesla.batchsub where teacher\_id =

(SELECT teacher\_id FROM tesla.teacher where email = (SELECT email FROM tesla.user where username = 'ankit') );