**DATABASE MANAGEMENT SYSTEM**

**PROJECT**

**Attendance Management System**

**BY :**

**Vivek Ramkumar**

**2018103082**

**Introduction:**

Attendance Management System is a software developed for daily student attendance in schools, colleges and institutes. If facilitates to access the attendance information of a particular student in a particular class. The information is sorted by the operators, which will be provided by the teacher for a particular class. This system will also help in evaluating attendance eligibility criteria of a student.

Attendance Management System basically has two main modules:

**¬ Admin login** can add/delete teachers,students,grade. Admin can create attendance charts and reports of the students.

¬ **Teacher login** can add daily attendance, generating report. Attendance report can be taken by given details of student details, date, class

**Purpose:**

The purpose of developing attendance management system is to computerized the tradition way of taking attendance which is done on hand written registers. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy as the records are maintained in the hand written registers. Another purpose for developing this software is to generate the attendance report/chart of any student or a whole class between any given interval.

**SOFTWARE USED:**

**Operating system**: Windows 10

**Front-End Language:** HTML,CSS

**Back-End :** PHP,SQL

**DataBase Server:** MySQL

**DataBase Tools:** MySQL 8.0.19, MySQL Workbench 8.0 CE

**Tables:**







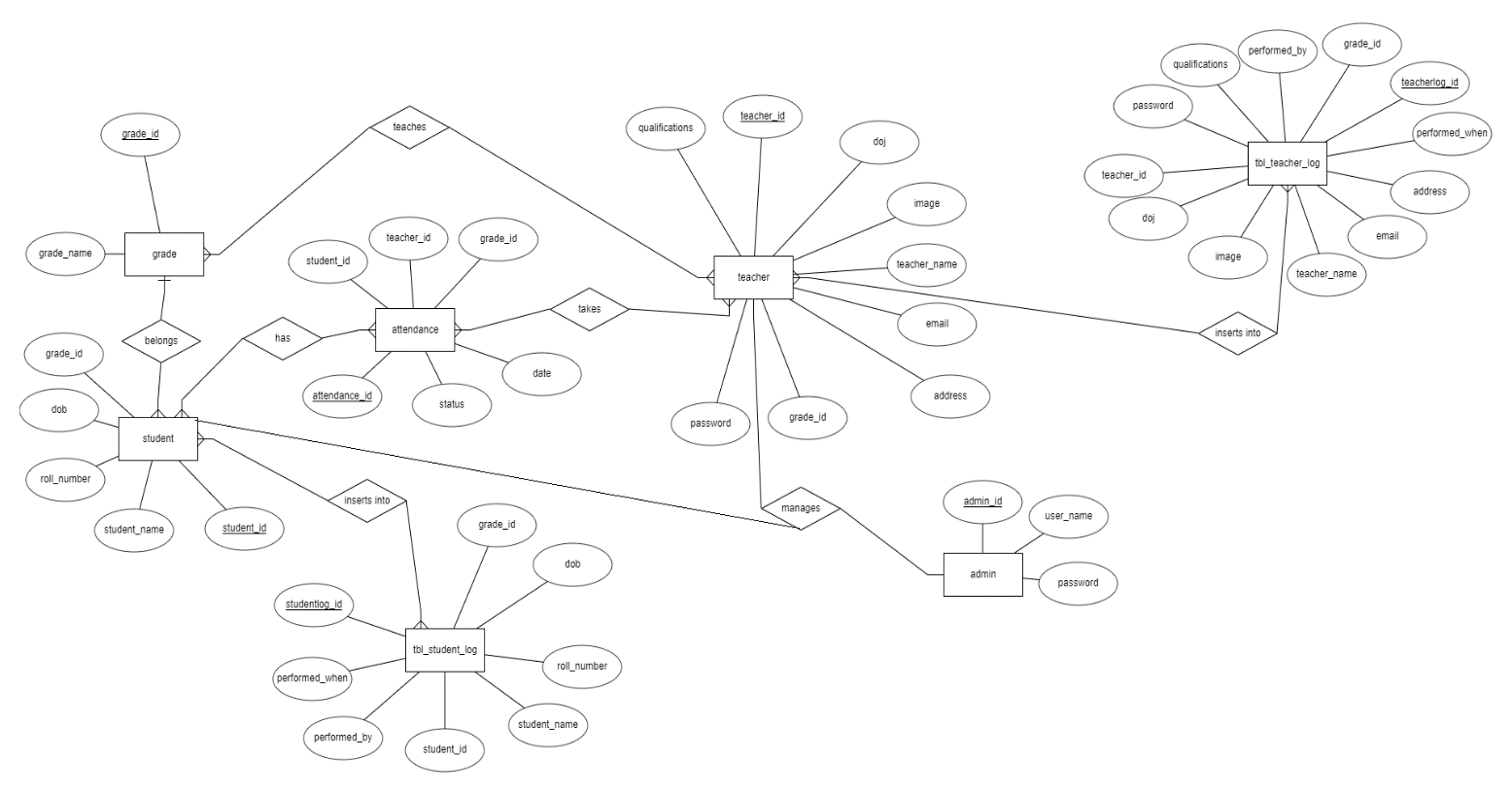




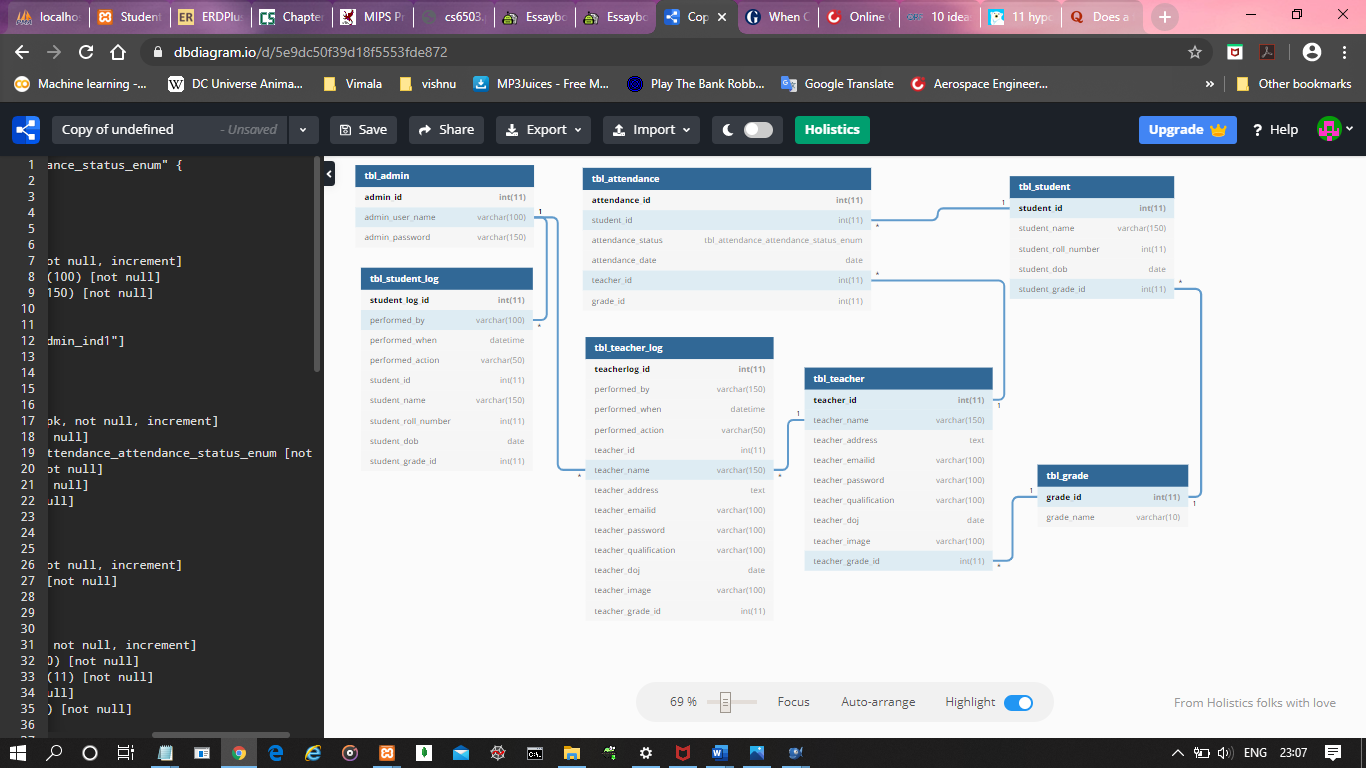




***ER DIAGRAM:***



***RELATIONAL MODEL:***



***TRIGGERS USED:***

**/\*Trigger functions for `tbl\_student\_log` \*/**

**1. If tbl\_student data is updated, trigger will insert into tbl\_student\_log**

DELIMITER $$

CREATE or replace DEFINER=CURRENT\_USER TRIGGER student\_log\_update

before update

ON tbl\_student

FOR EACH ROW

BEGIN

INSERT INTO tbl\_student\_log(performed\_by,performed\_when,performed\_action,student\_id,student\_name,student\_roll\_number,student\_dob,student\_grade\_id)

VALUES ('admin',now(),'UPDATE',new.student\_id,new.student\_name,new.student\_roll\_number,new.student\_dob,new.student\_grade\_id);

END;

$$

1. **If tbl\_student data is deleted, trigger will insert into tbl\_student\_log**

DELIMITER $$

CREATE or replace DEFINER=CURRENT\_USER TRIGGER student\_log\_delete

before delete

ON tbl\_student

FOR EACH ROW

BEGIN

INSERT INTO tbl\_student\_log (performed\_by,performed\_when,performed\_action,student\_id,student\_name,student\_roll\_number,student\_dob,student\_grade\_id)

VALUES ('admin',now(),'DELETE',old.student\_id,old.student\_name,old.student\_roll\_number,old.student\_dob,old.student\_grade\_id);

END;

$$

1. **If tbl\_student data is inserted, trigger will insert into tbl\_student\_log**

DELIMITER $$

CREATE or replace DEFINER=CURRENT\_USER TRIGGER student\_log\_insert

after insert

ON tbl\_student

FOR EACH ROW

BEGIN

INSERT INTO tbl\_student\_log (performed\_by,performed\_when,performed\_action,student\_id,student\_name,student\_roll\_number,student\_dob,

student\_grade\_id)

VALUES ('admin',now(),'INSERT',new.student\_id,new.student\_name,new.student\_roll\_number,new.student\_dob,

new.student\_grade\_id);

END;

$$

**/\*Trigger functions for `tbl\_teacher\_log` \*/**

1. **If tbl\_teacher data is updated, trigger will insert into tbl\_teacher\_log**

DELIMITER $$

CREATE or replace DEFINER=CURRENT\_USER TRIGGER teacher\_log\_update

before update

ON tbl\_teacher

FOR EACH ROW

BEGIN

INSERT INTO tbl\_teacher\_log (performed\_by,performed\_when,performed\_action,teacher\_id,teacher\_name,teacher\_address,teacher\_emailid,teacher\_password,teacher\_qualification,teacher\_doj,teacher\_image,teacher\_grade\_id)

VALUES (@username,now(),'UPDATE',new.teacher\_id,new.teacher\_name,new.teacher\_address,new.teacher\_emailid,new.teacher\_password,new.teacher\_qualification,new.teacher\_doj,new.teacher\_image,new.teacher\_grade\_id);

END;

$$

1. **If tbl\_teacher data is inserted, trigger will insert into tbl\_teacher\_log**

DELIMITER $$

CREATE or replace DEFINER=CURRENT\_USER TRIGGER teacher\_log\_insert

after insert

ON tbl\_teacher

FOR EACH ROW

BEGIN

INSERT INTO tbl\_teacher\_log (performed\_by,performed\_when,performed\_action,teacher\_id,teacher\_name,teacher\_address,teacher\_emailid,teacher\_password,teacher\_qualification,teacher\_doj,teacher\_image,teacher\_grade\_id)

VALUES ('admin',now(),'INSERT',new.teacher\_id,new.teacher\_name,new.teacher\_address,new.teacher\_emailid,new.teacher\_password,new.teacher\_qualification,new.teacher\_doj,new.teacher\_image,new.teacher\_grade\_id);

END;

$$

1. **If tbl\_teacher data is deleted, trigger will insert into tbl\_teacher\_log**

DELIMITER $$

CREATE or replace DEFINER=CURRENT\_USER TRIGGER teacher\_log\_delete

before delete

ON tbl\_teacher

FOR EACH ROW

BEGIN

INSERT INTO tbl\_teacher\_log (performed\_by,performed\_when,performed\_action,teacher\_id,teacher\_name,teacher\_address,teacher\_emailid,teacher\_password,teacher\_qualification,teacher\_doj,teacher\_image,teacher\_grade\_id)

VALUES ('admin',now(),'DELETE',old.teacher\_id,old.teacher\_name,old.teacher\_address,old.teacher\_emailid,old.teacher\_password,old.teacher\_qualification,old.teacher\_doj,old.teacher\_image,old.teacher\_grade\_id);

END;

$$

***VIEWS USED:***

1. **/\*View for`student\_join\_view` \*/**

create or replace view student\_join\_view as

SELECT \* FROM tbl\_student

INNER JOIN tbl\_grade

ON tbl\_grade.grade\_id = tbl\_student.student\_grade\_id ;

1. **/\*View for`attendance\_join\_view` \*/**

create or replace view attendance\_join\_view as

SELECT a.attendance\_id,a.student\_id,a.attendance\_status,a.attendance\_date,a.teacher\_id as v,a.grade\_id as p,b.student\_id as x,b.student\_name,b.student\_roll\_number,b.student\_dob,b.student\_grade\_id as y,c.grade\_id,c.grade\_name FROM tbl\_attendance a

INNER JOIN tbl\_student b

ON b.student\_id = a.student\_id

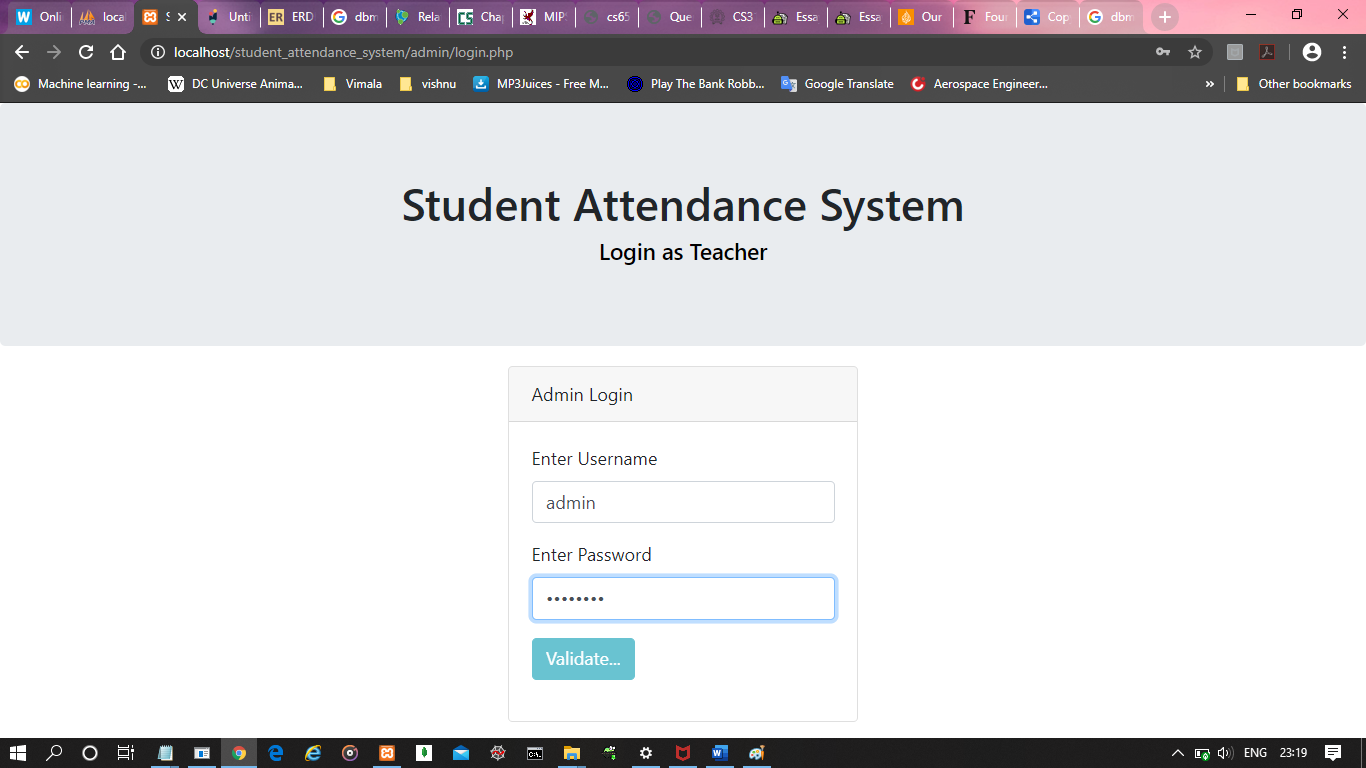
INNER JOIN tbl\_grade c

ON c.grade\_id = b.student\_grade\_id ;

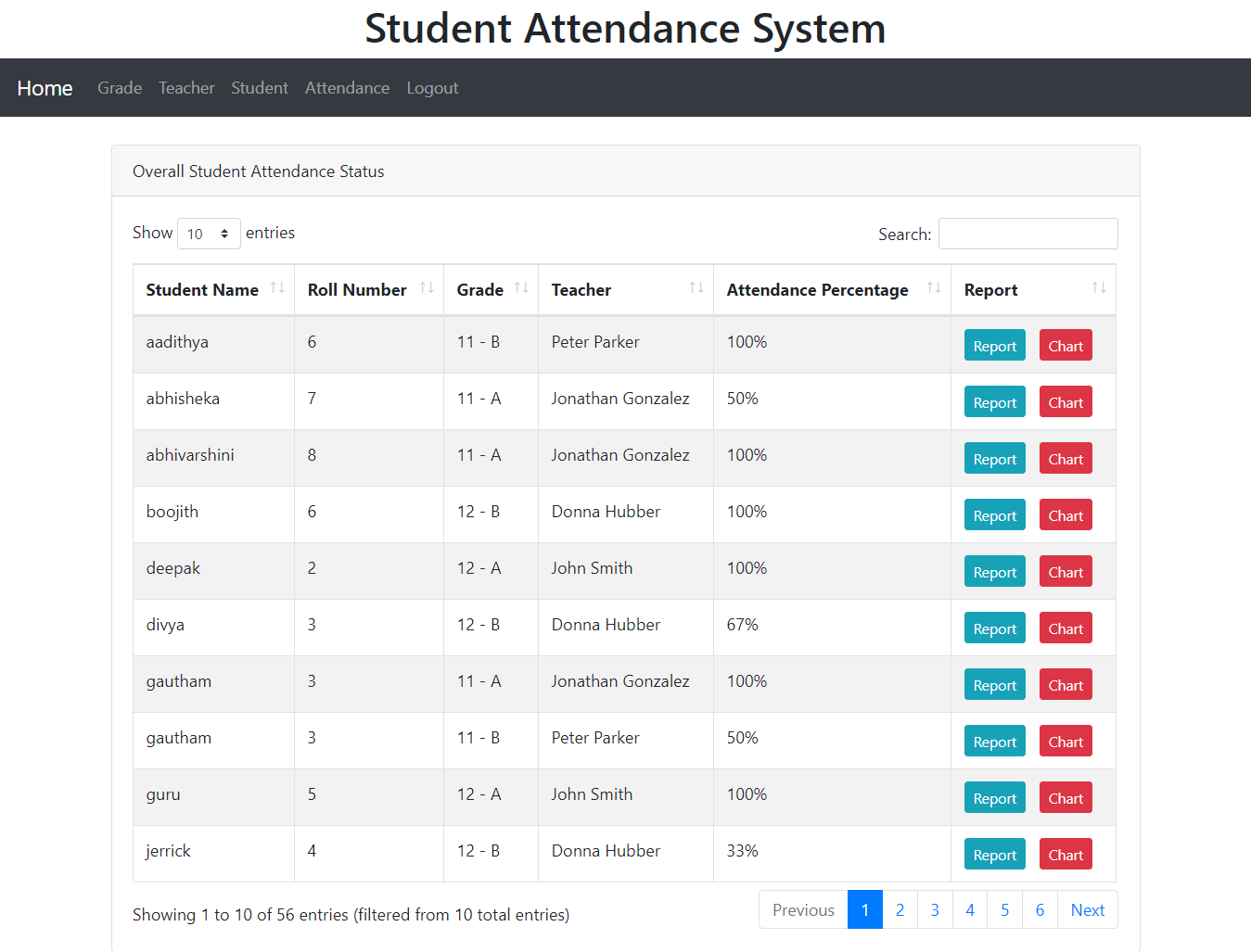
***SCREENSHOTS:***

***ADMIN LOGIN:***

**Login page:**



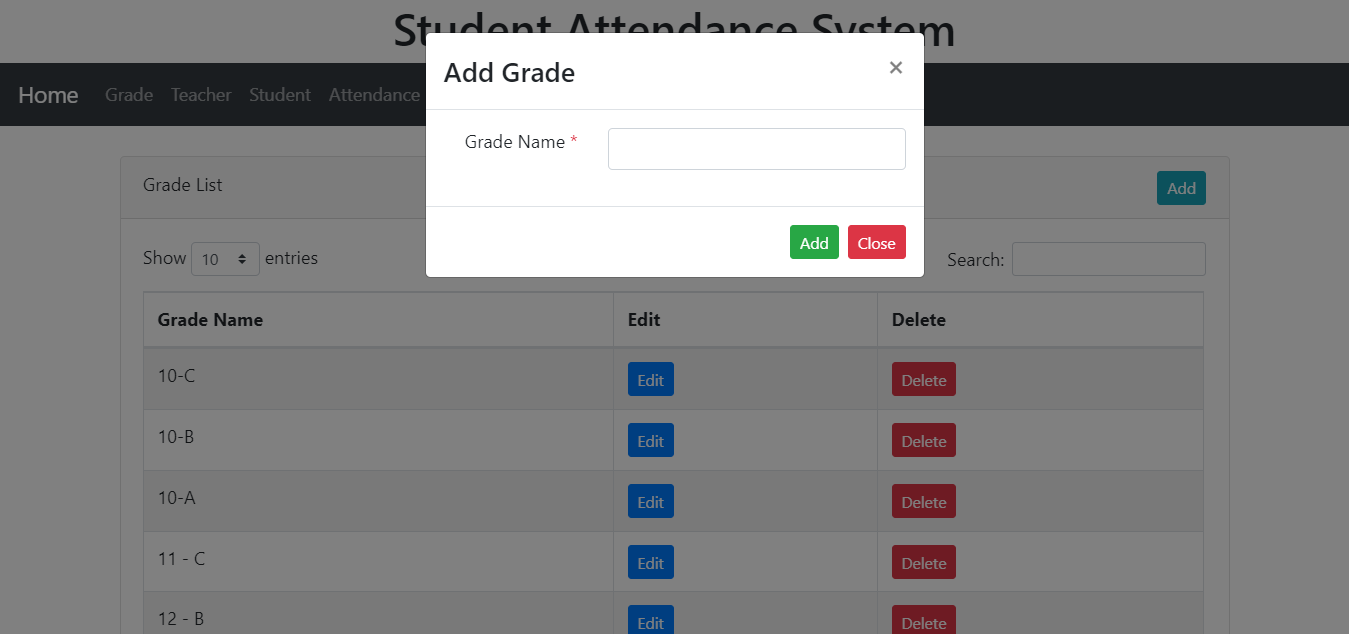
**Attendance status of all the students:**



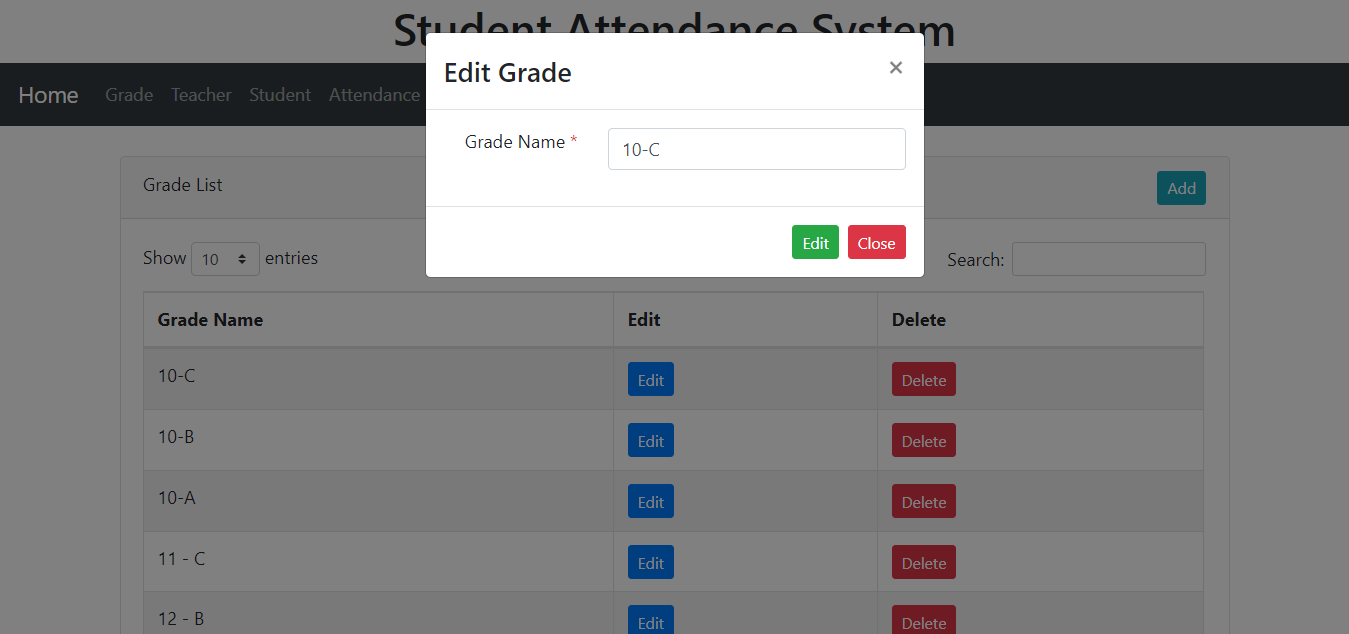
**Grade lists:**



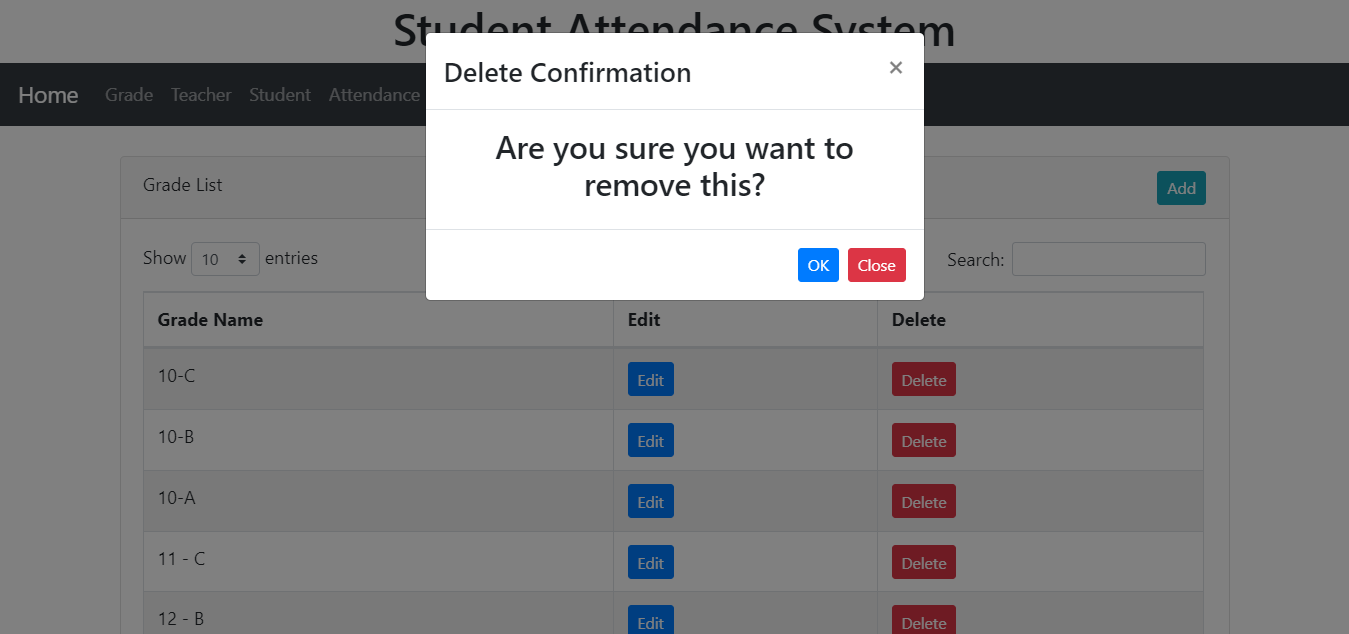
**Add grade:**



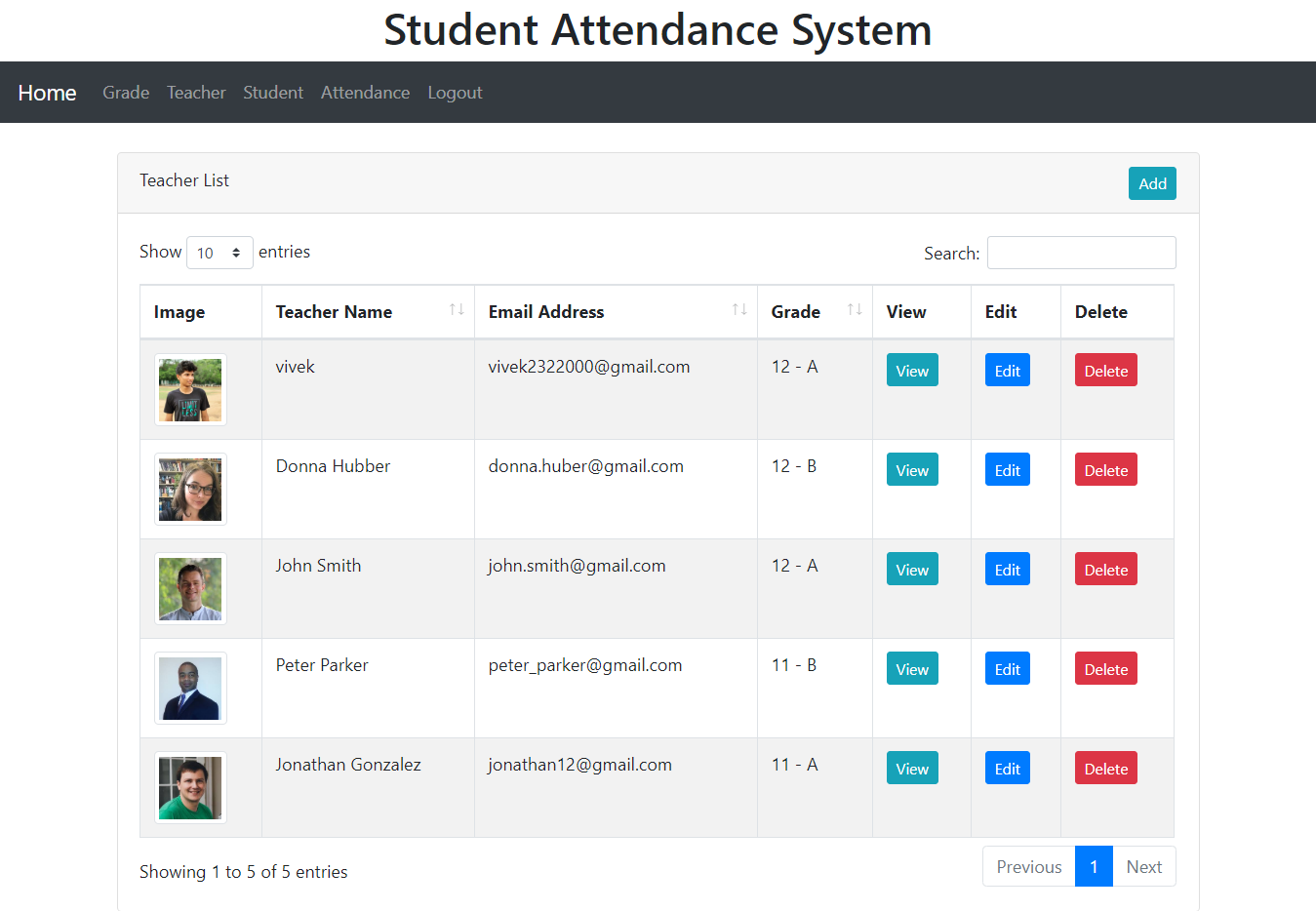
**Edit Grade:**



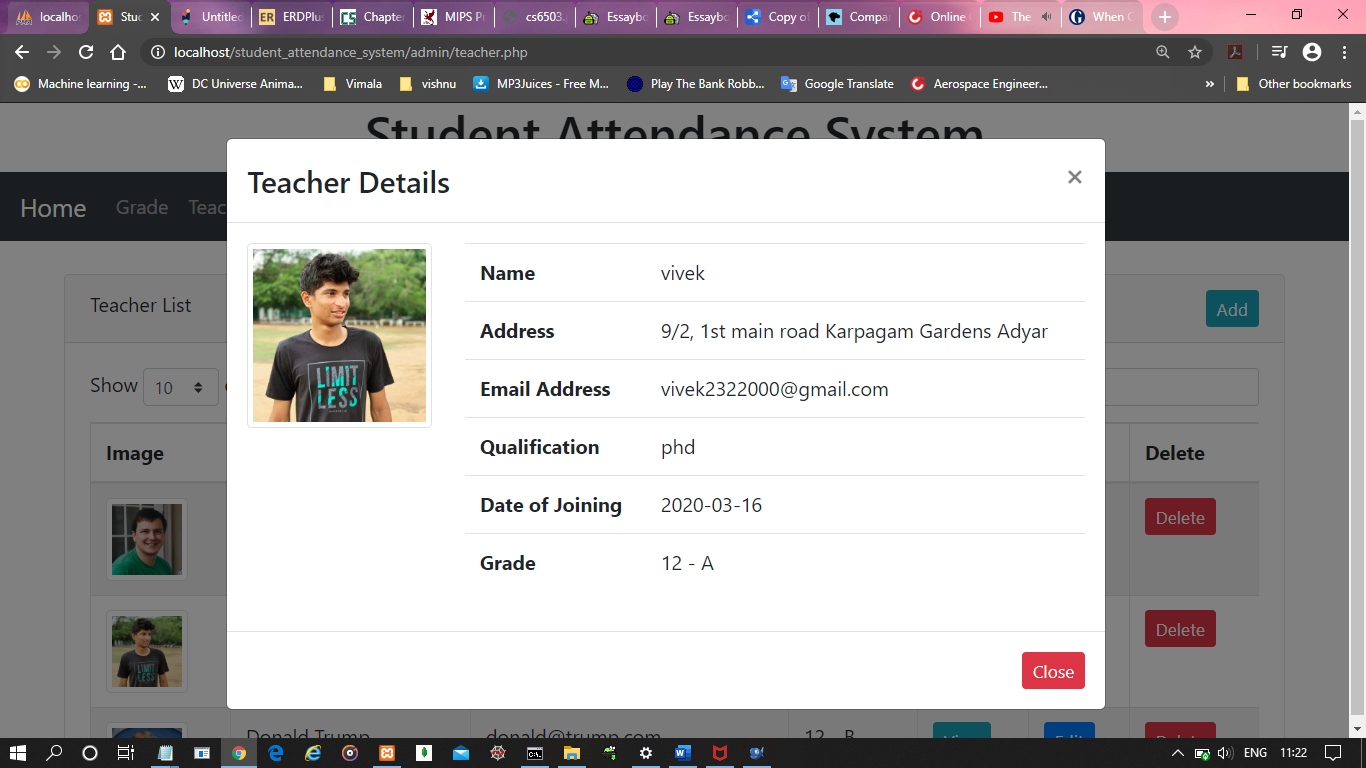
**Delete Grade:**



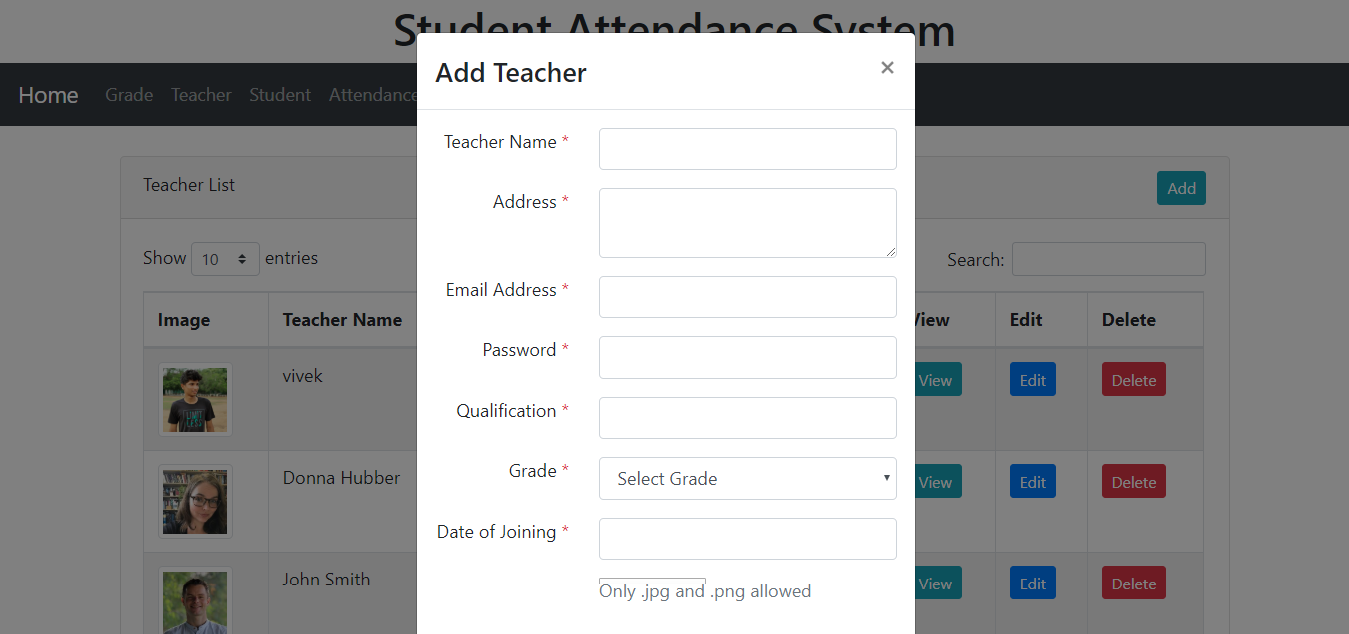
**Teacher List:**



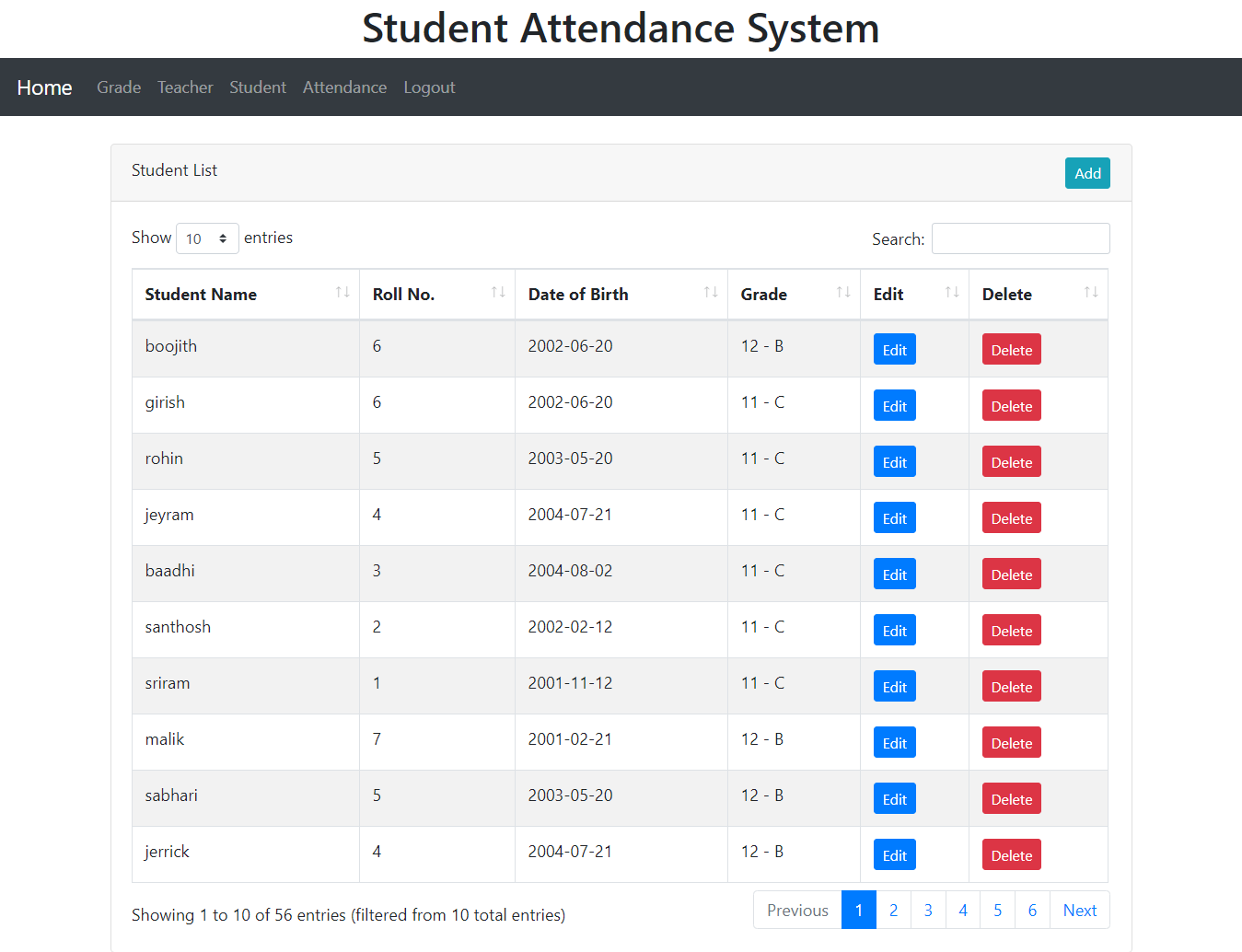
**Teacher Details:**



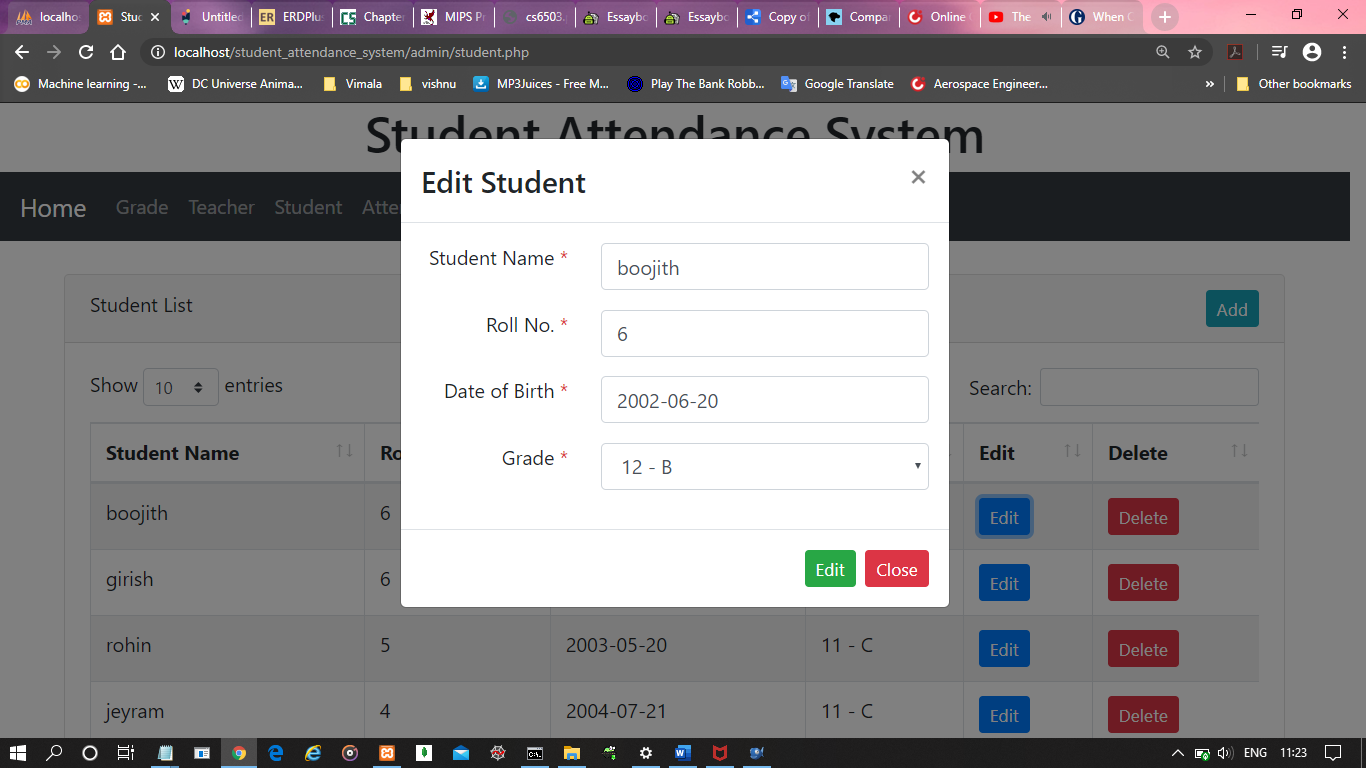
**Add Teacher:**



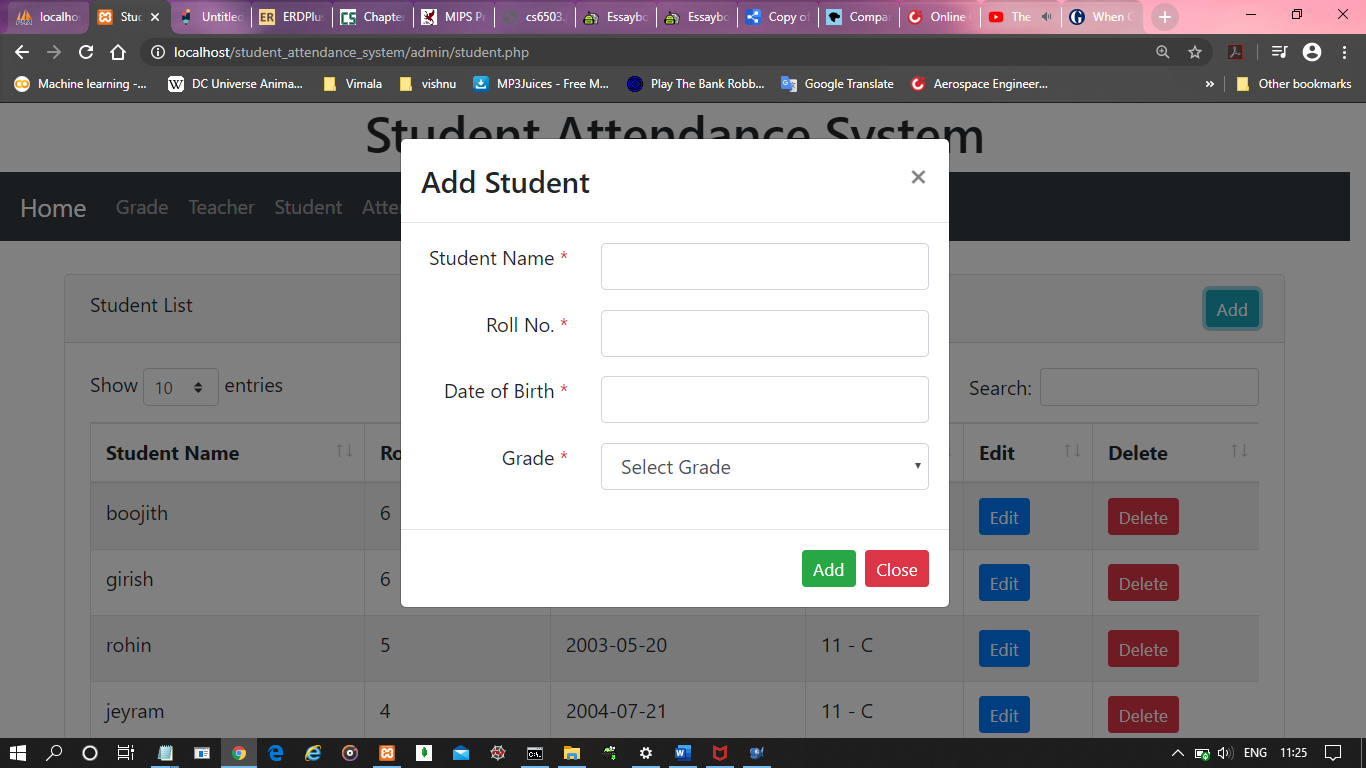
**Student List:**



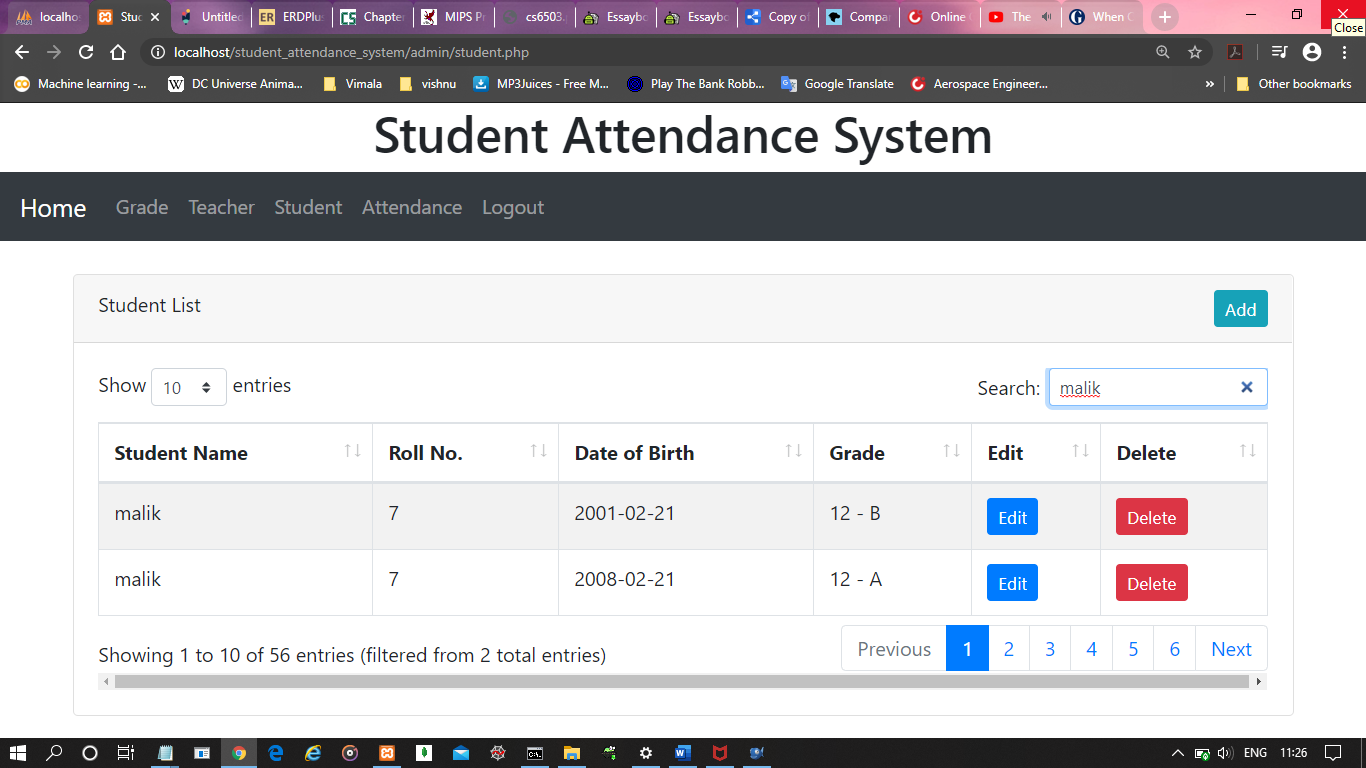
**Edit Student:**



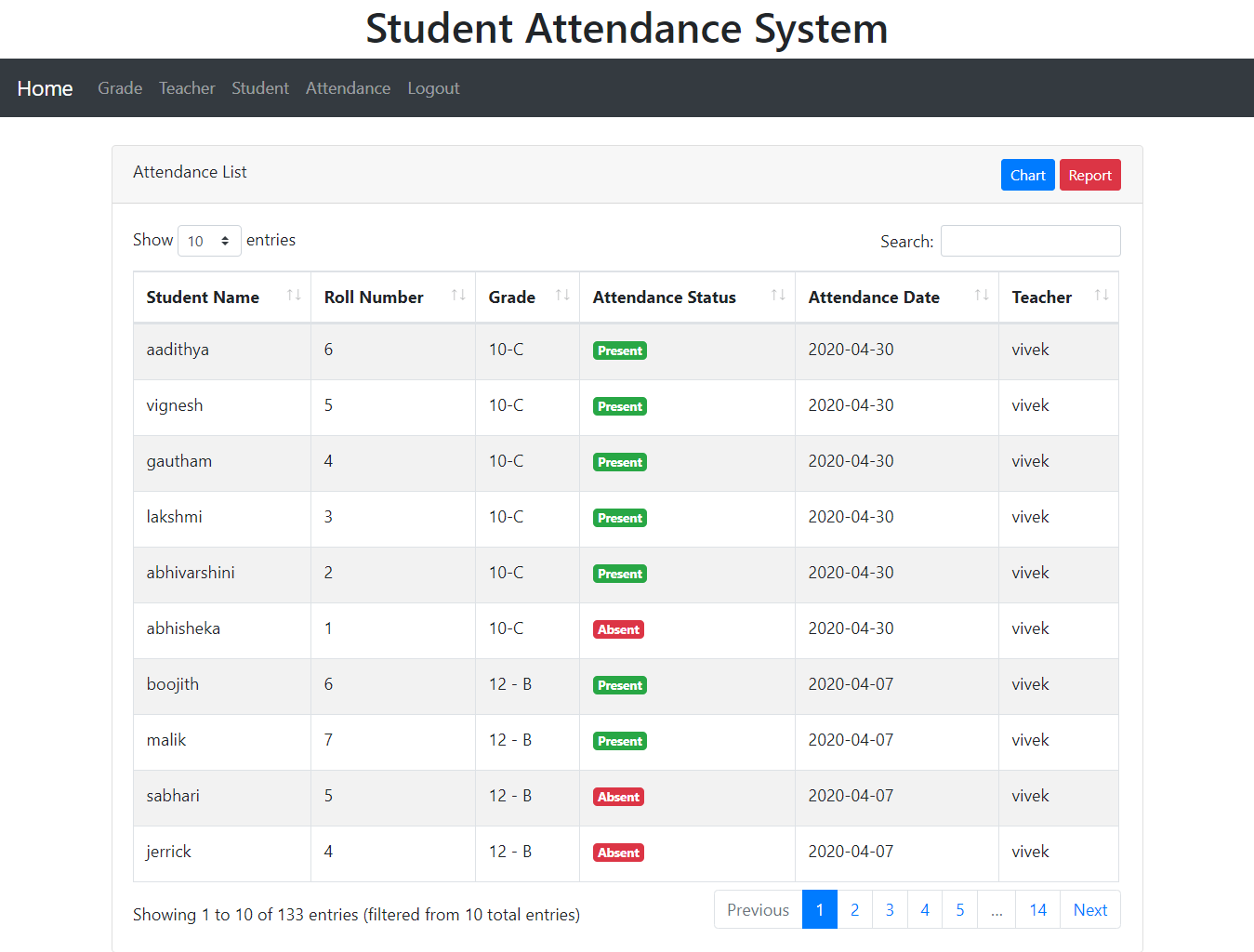
**Add Student:**



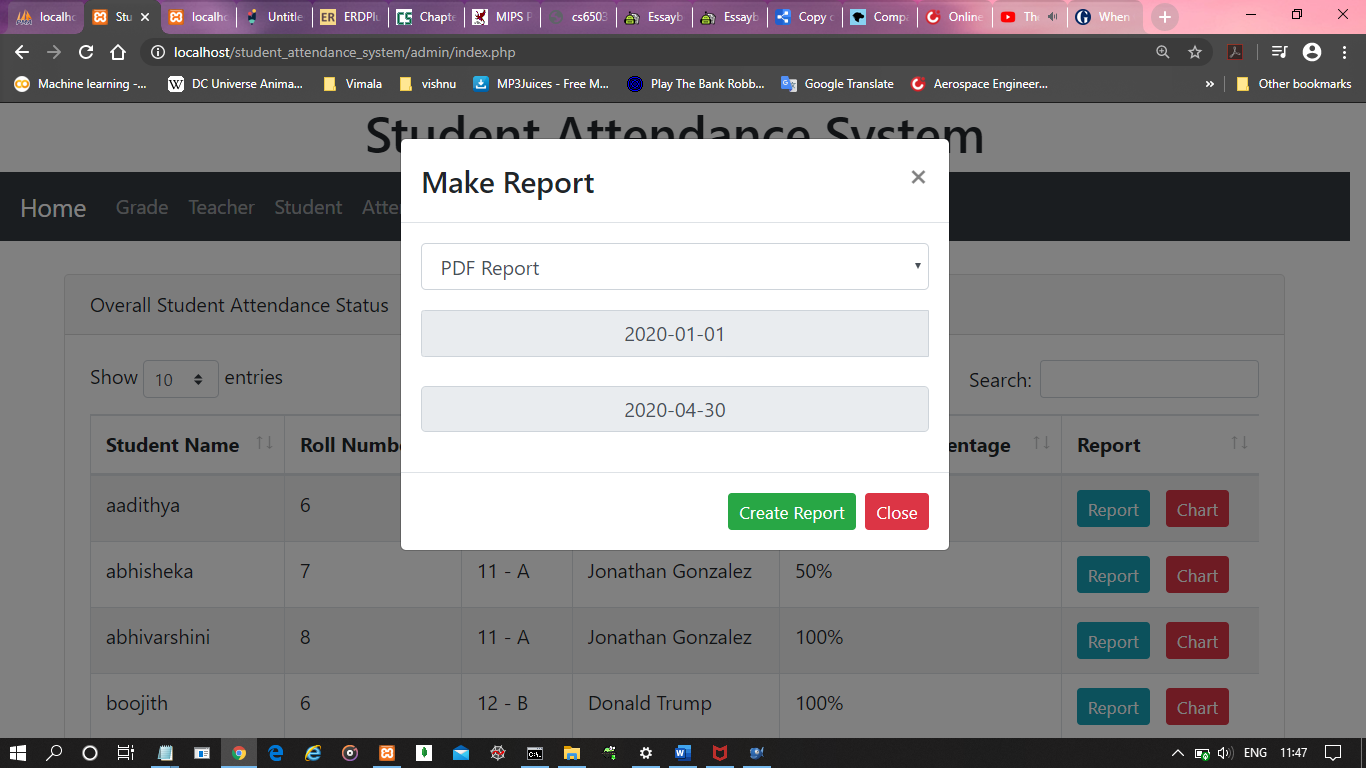
**Search for any student by name, roll no, date of birth, grade:**

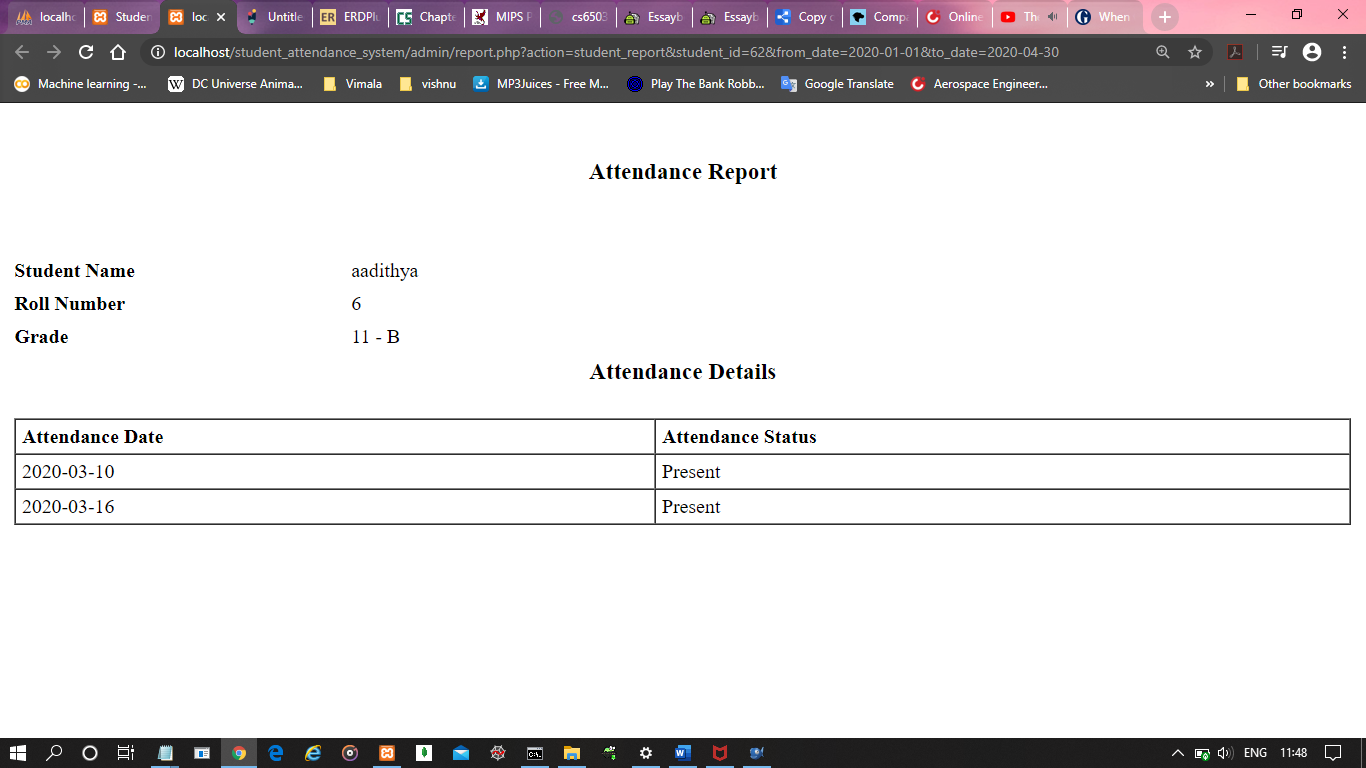


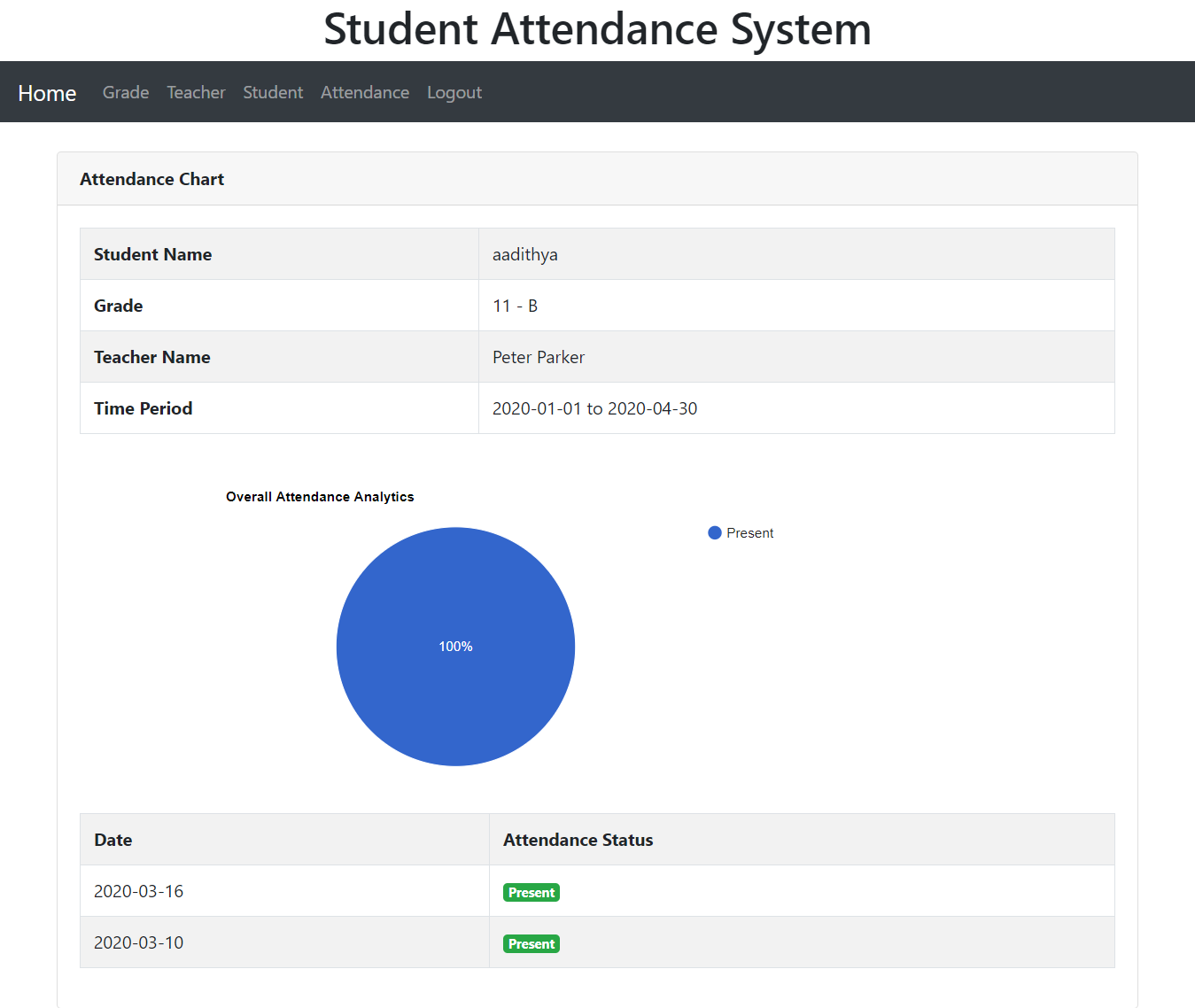
**Attendance List of Students:**



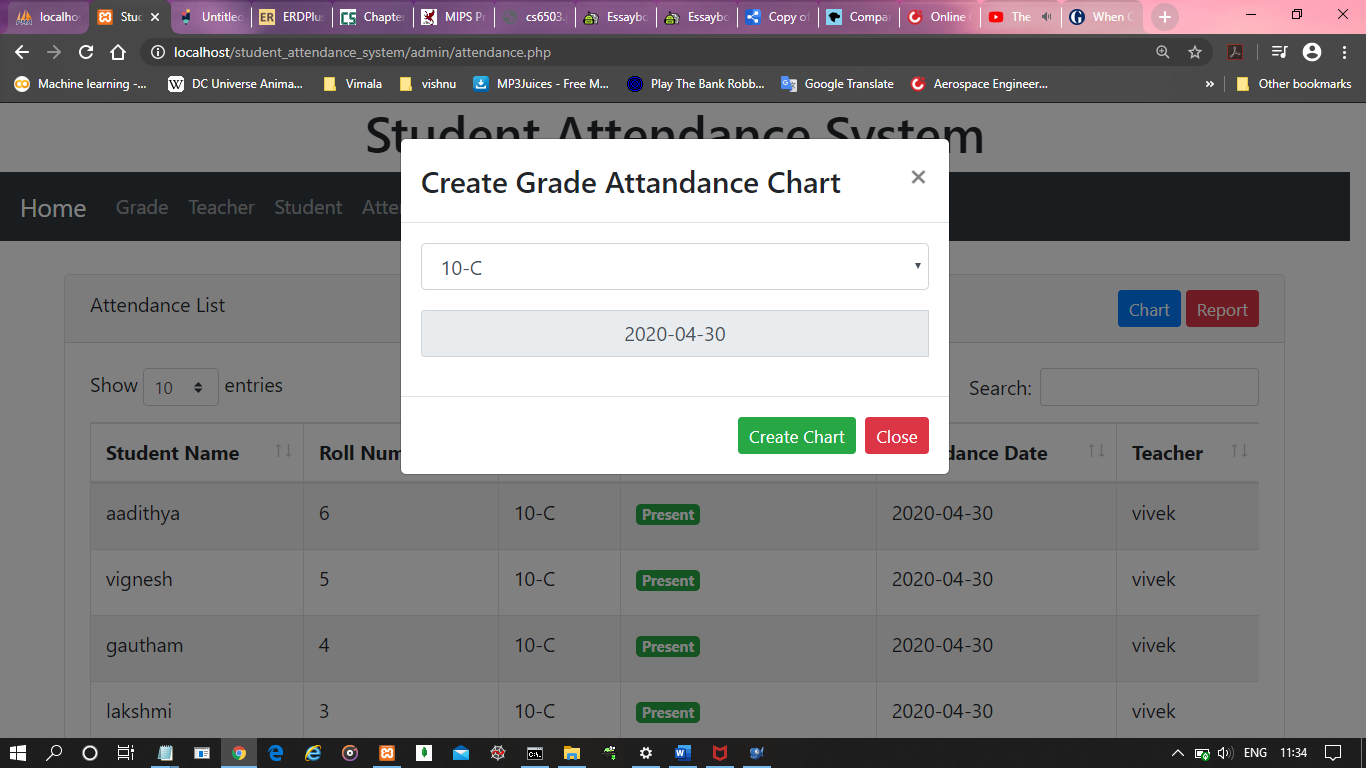
**Create Chart/Report for any particular student between particular intervals:**



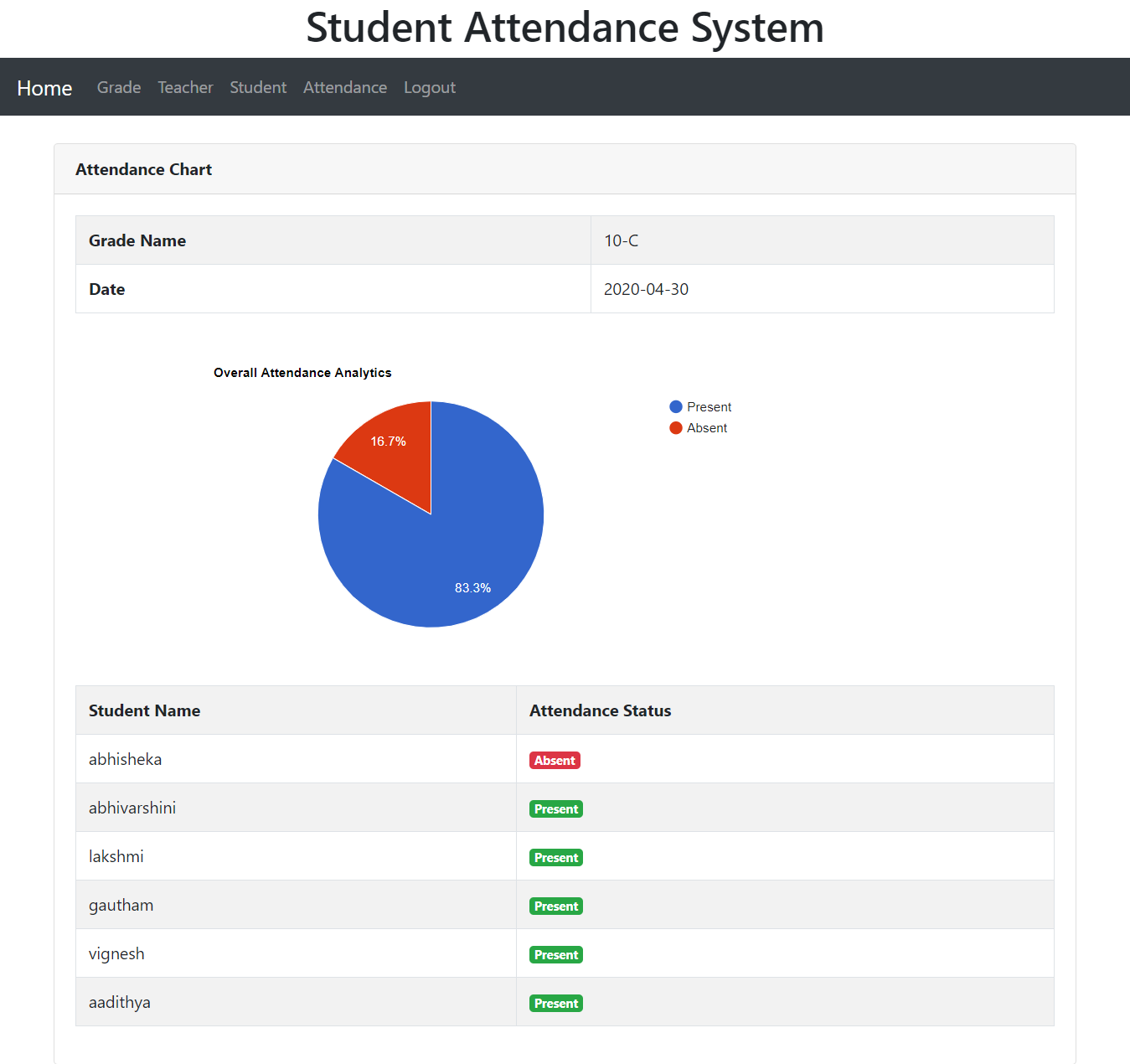


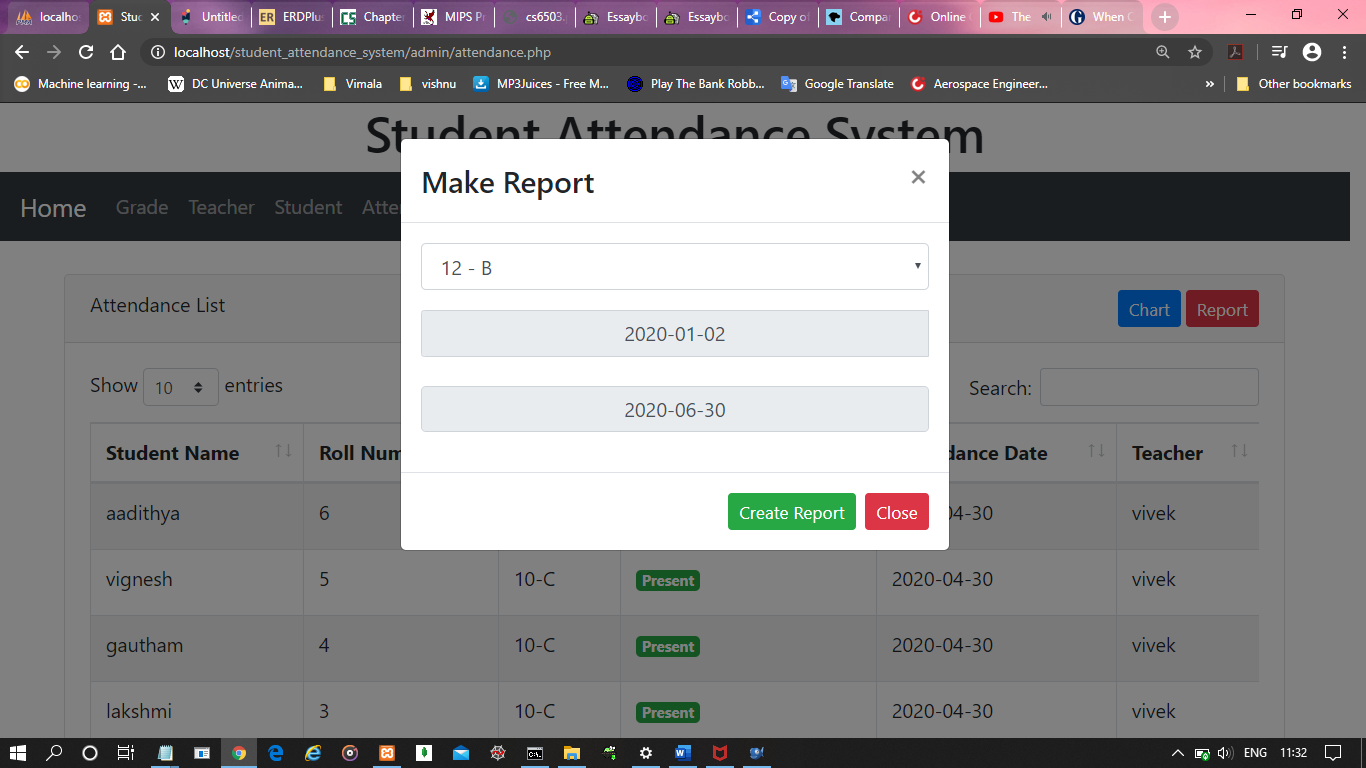


**Create Attendance Chart of an entire Grade for a particular Date:**

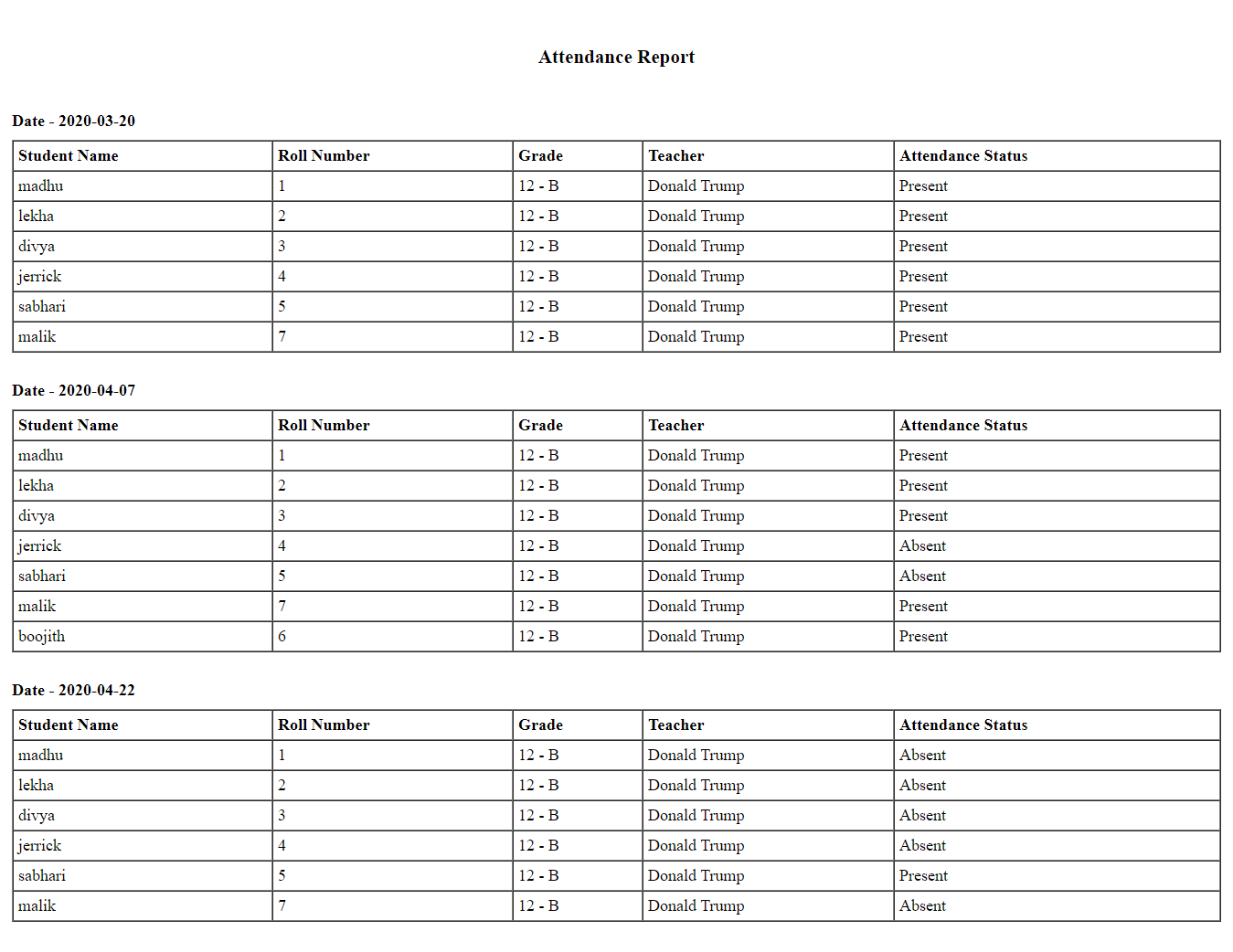


**Create Attendance Chart of an entire Grade for a particular Date:**



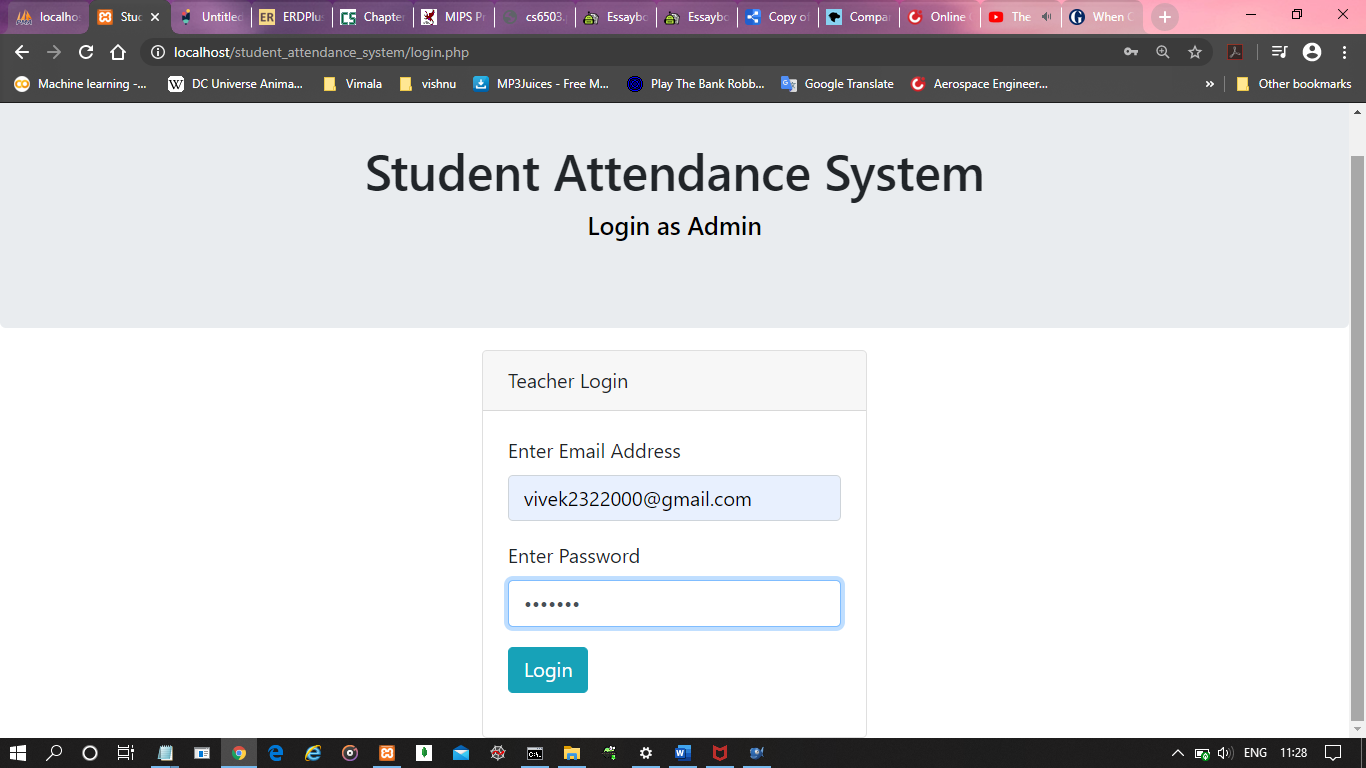
**Make Attendance Report between any intervals for an entire Grade:**

**Make Attendance Report between any intervals for an entire Grade:**



***TEACHER LOGIN:***

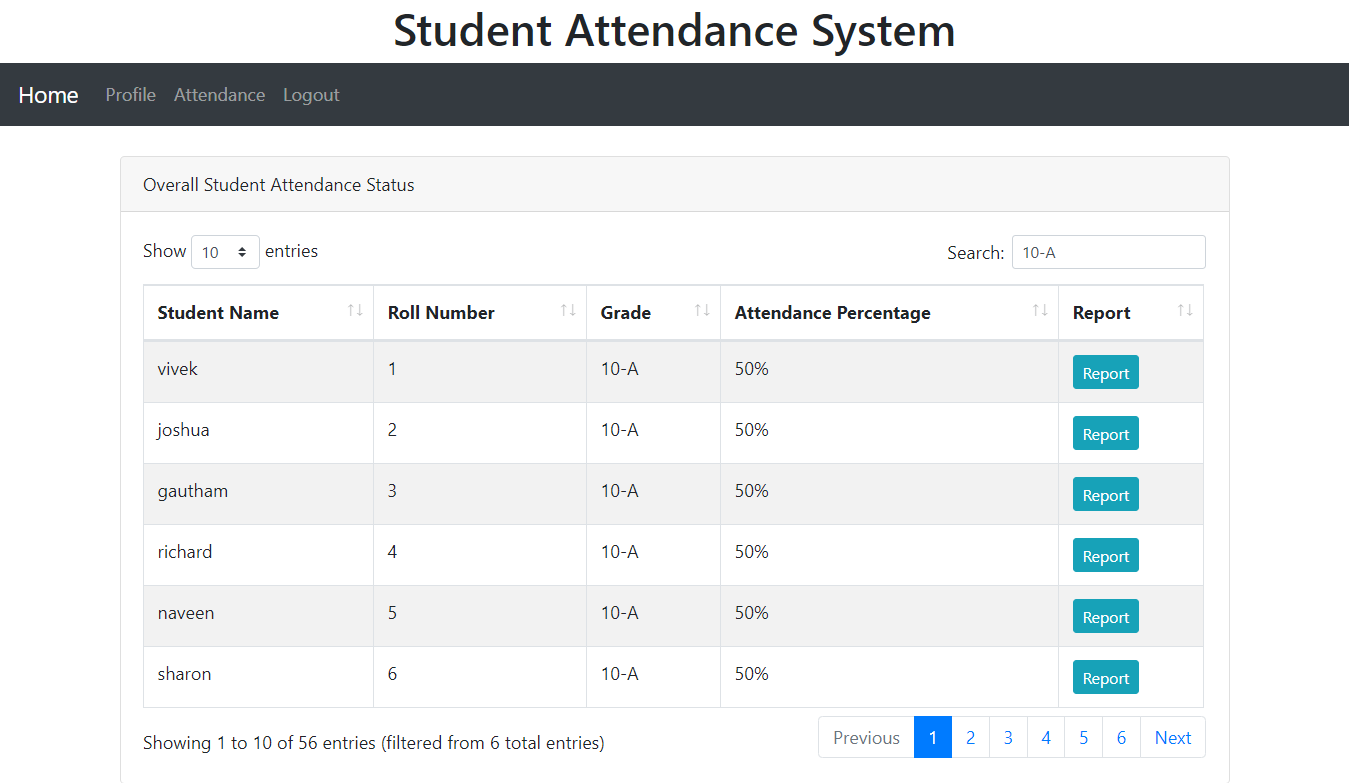
**Login Page:**

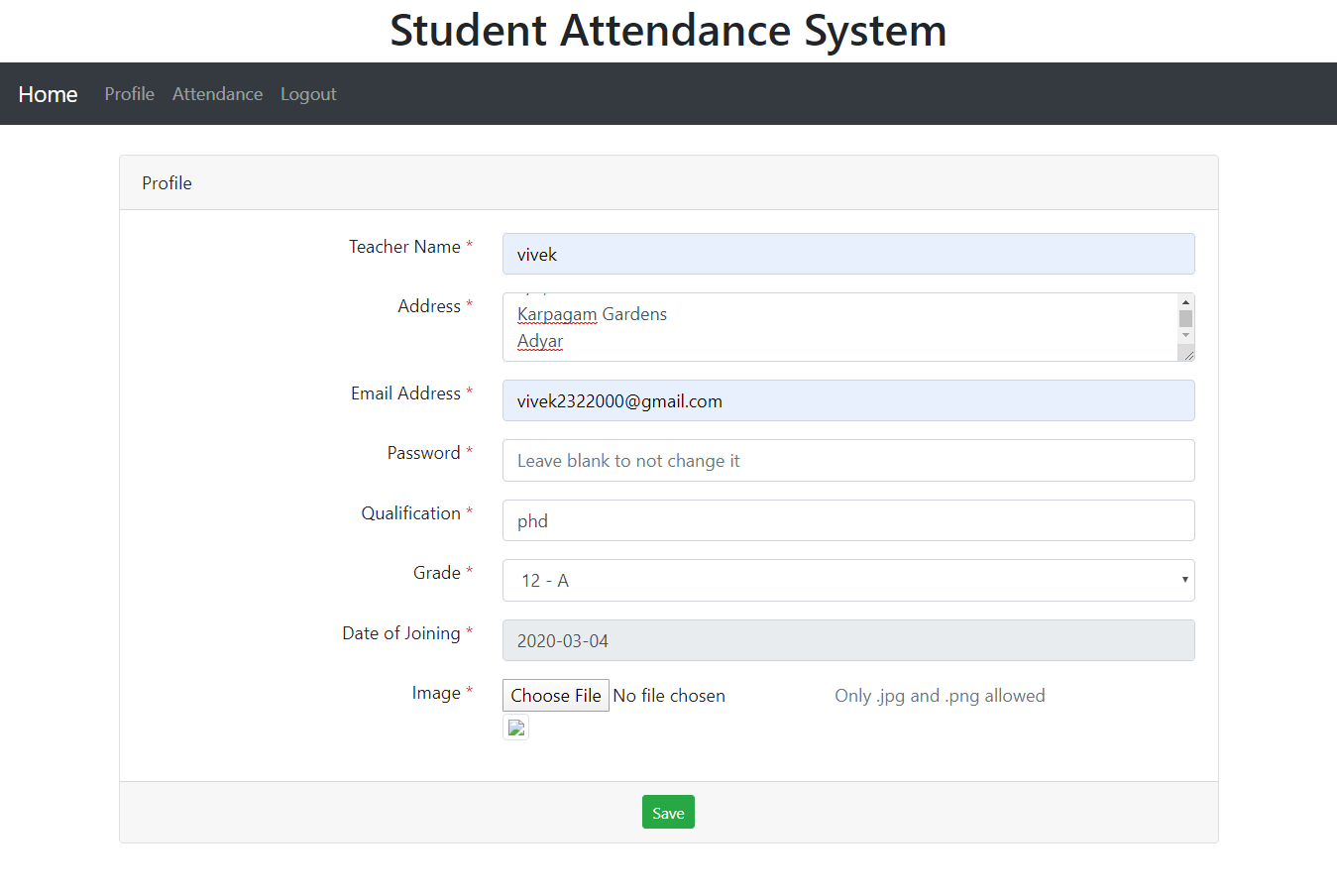


**Overall Student Attendance of Students:**

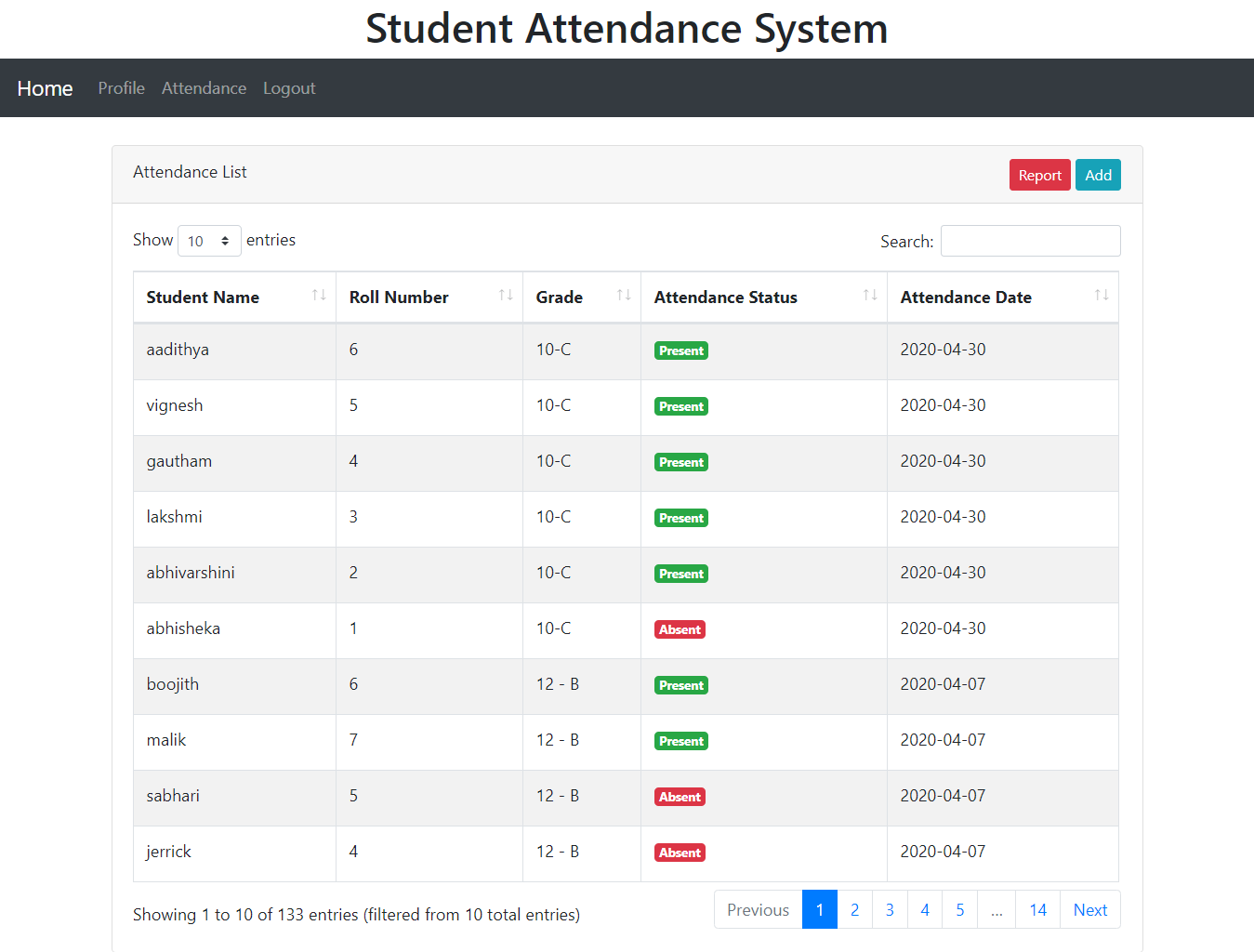


**Search for any student by name, roll no, grade:**

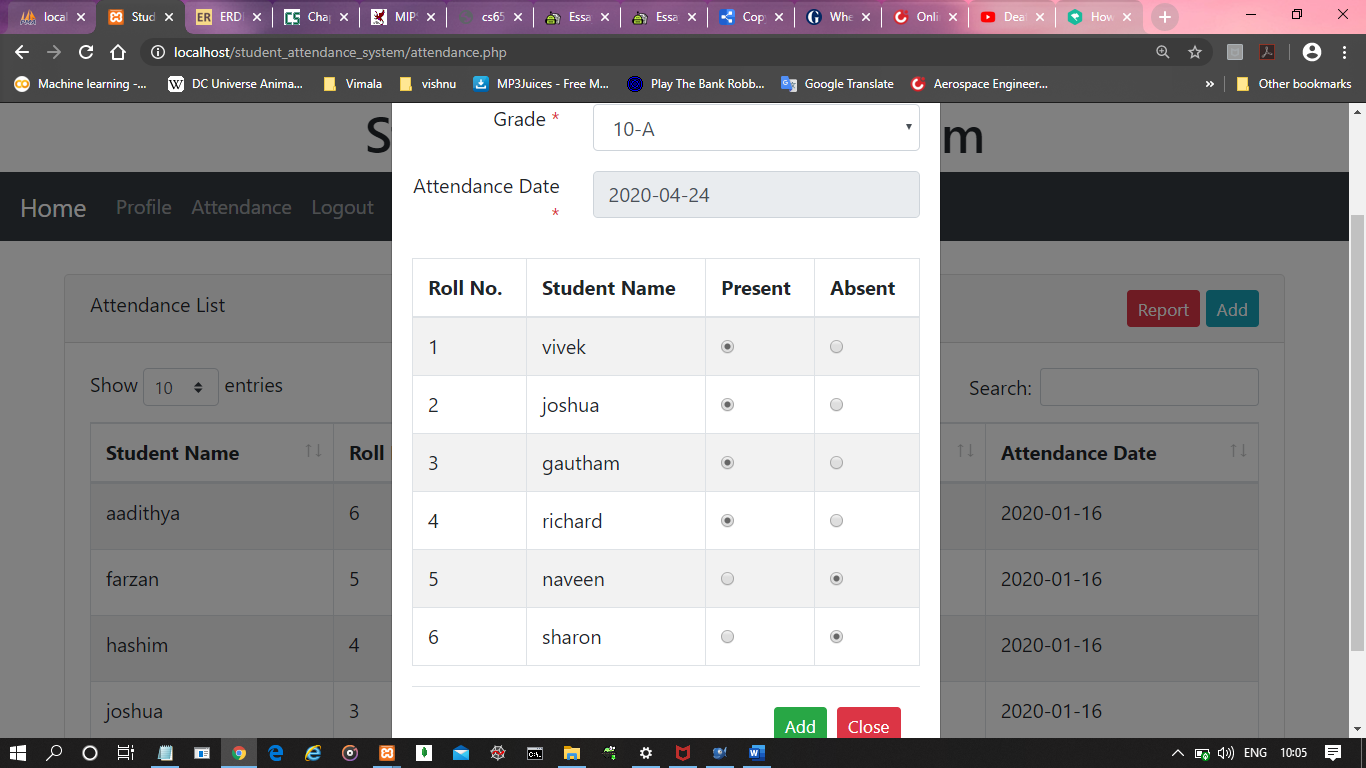


**Edit any Student’s Profile**:

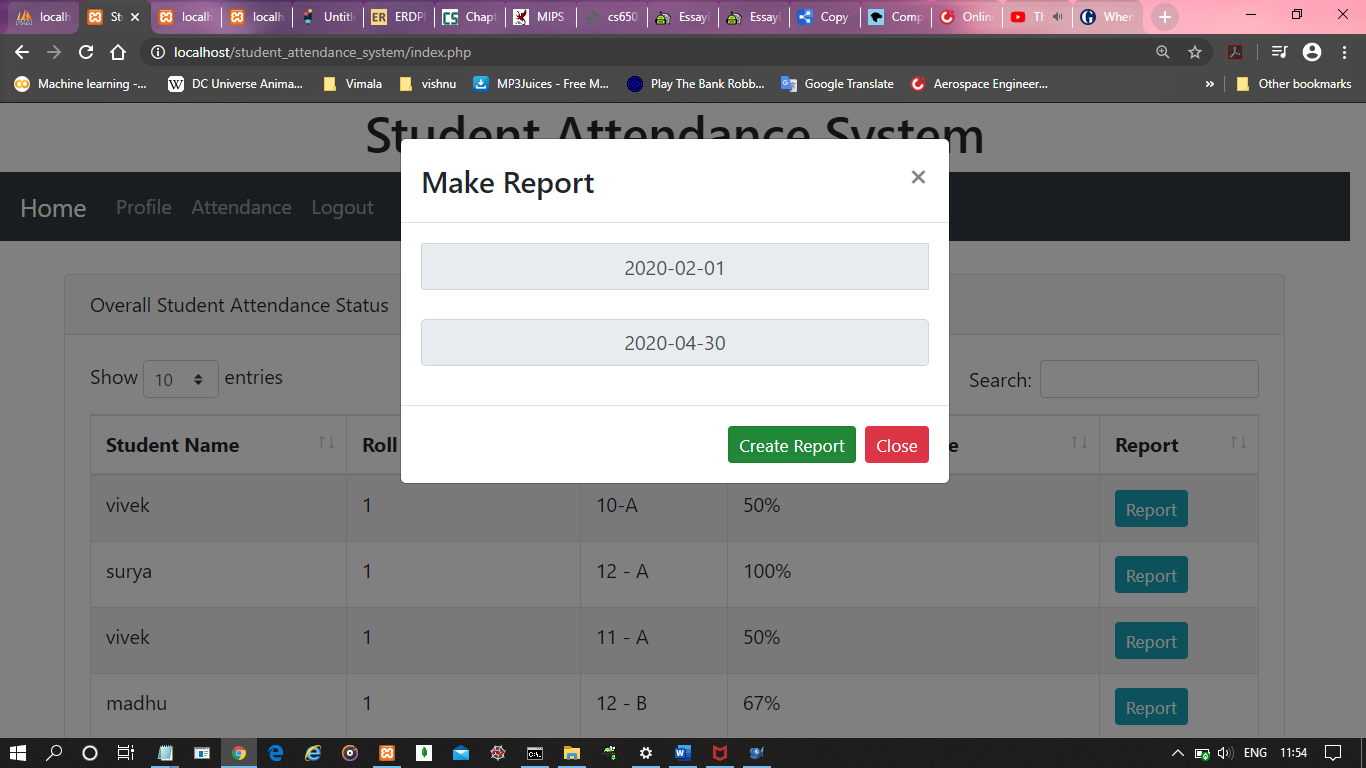
**Attendance List of Students:**



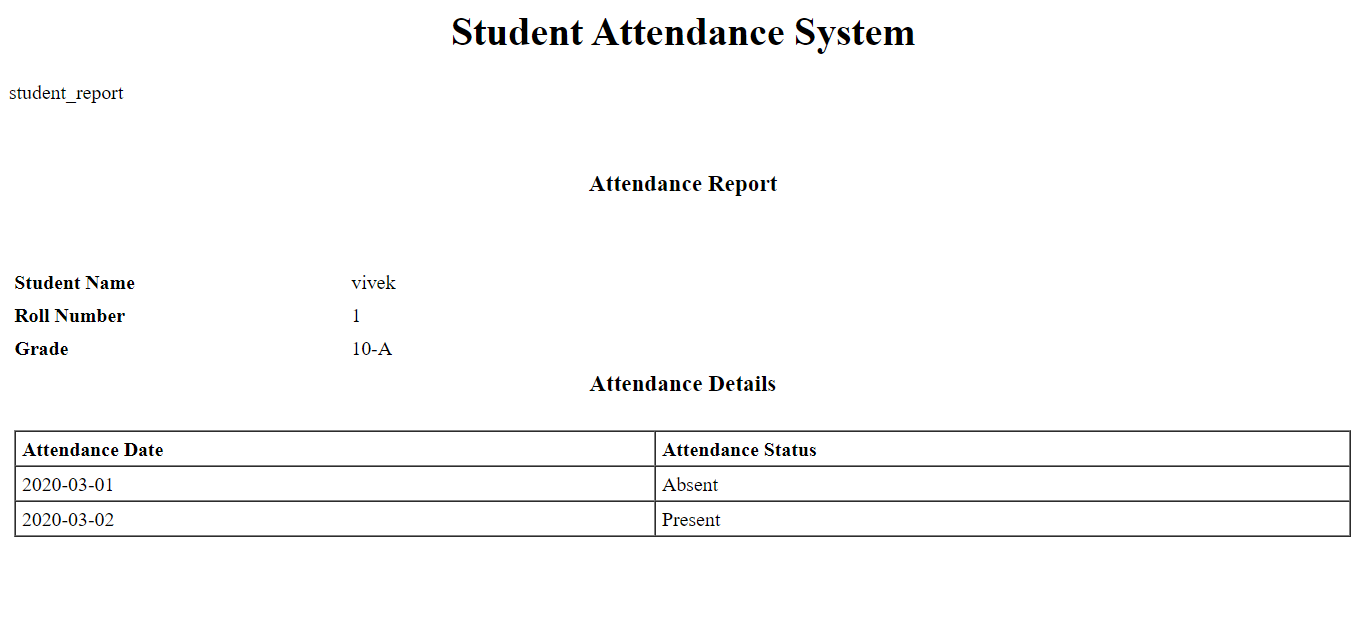
**Add Attendance for any grade:**



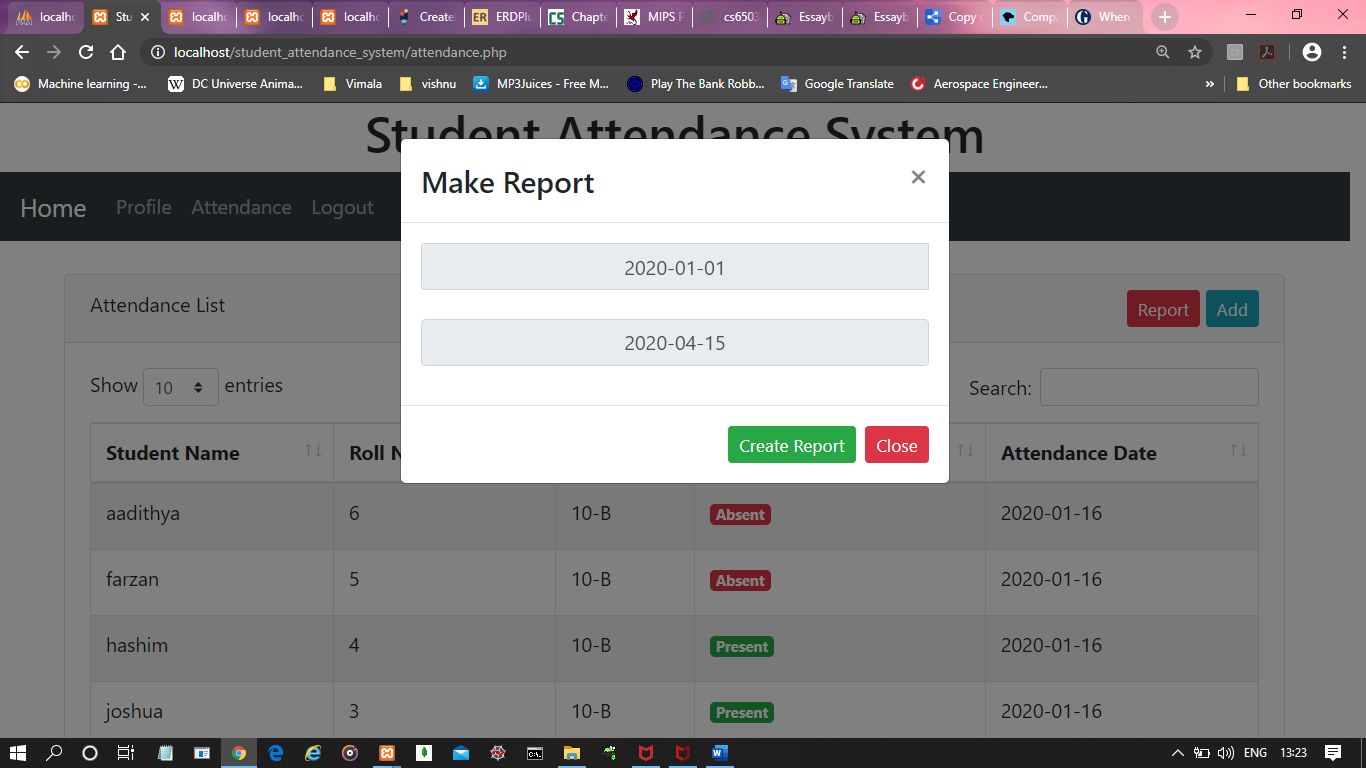
**Make Attendance Report for any particular student between Intervals:**



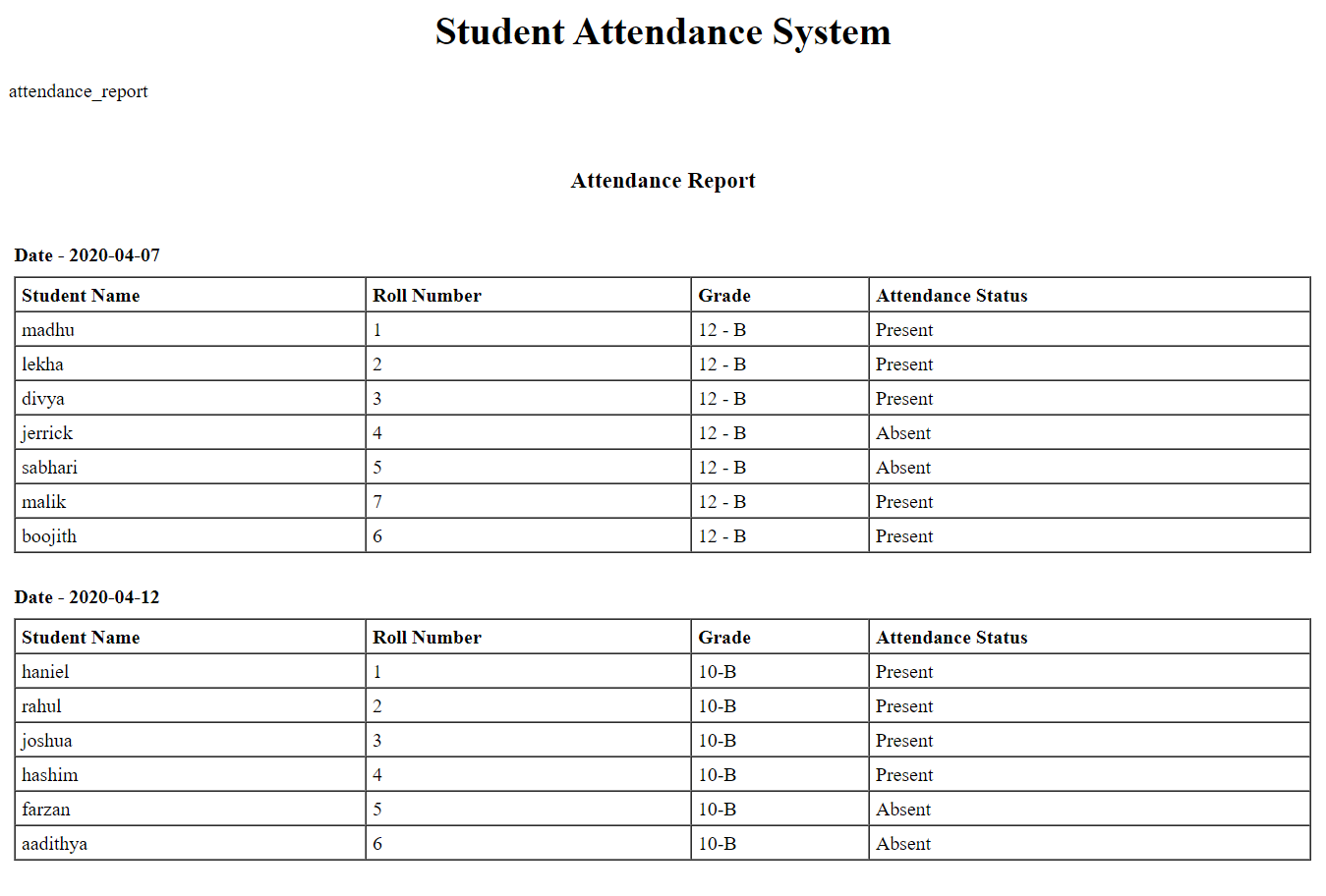
**Attendance Report for any particular student between Intervals:**



**Make Attendance Report of all Classes between any Interval:**

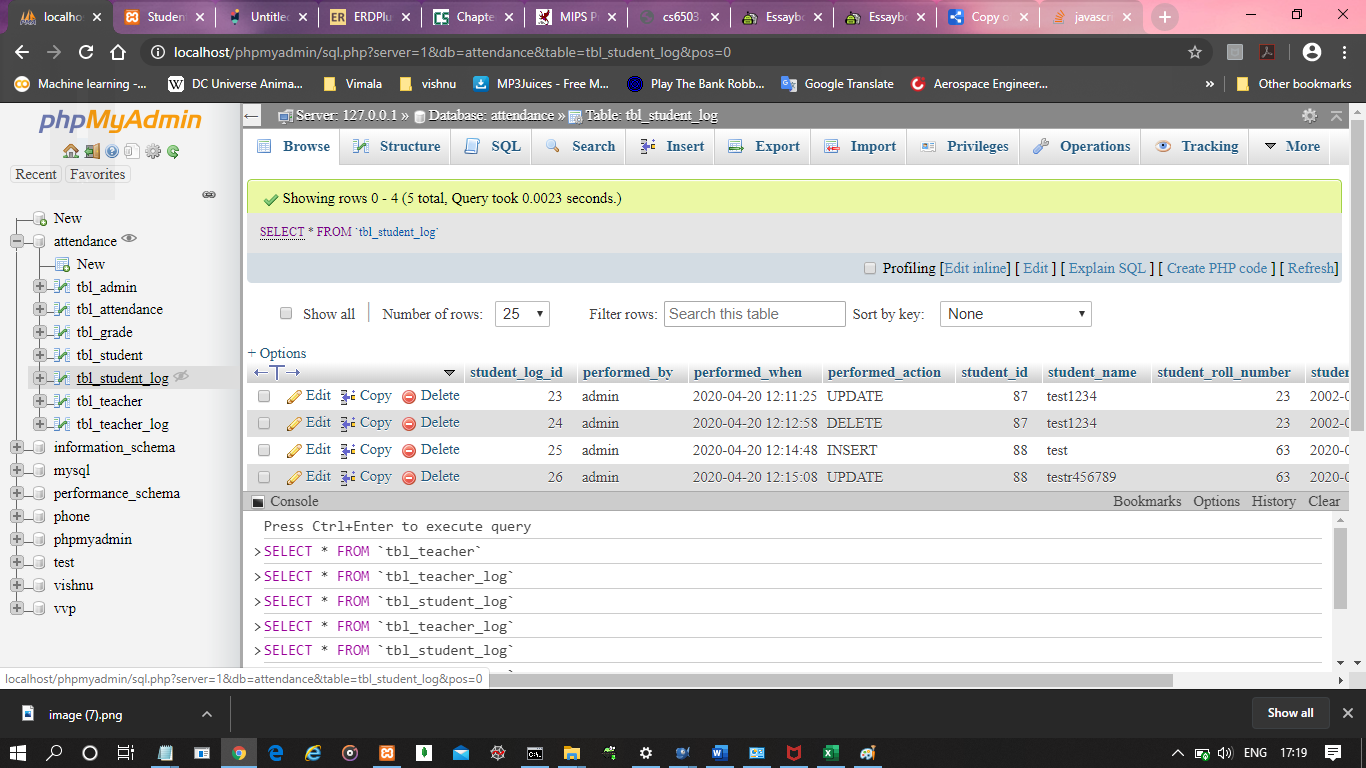


**Attendance Report of all Classes between any Interval:**



***TRIGGERS USED :***

**Data Inserted into tbl\_student\_log:**



**Data Inserted into tbl\_teacher\_log:**

