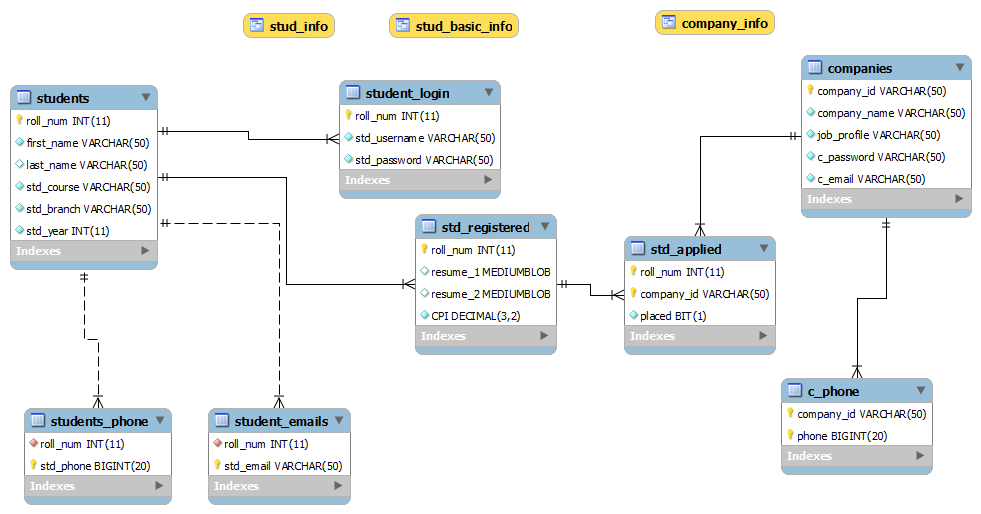
PLACEMENT PORTAL

GROUP #17



This Doc includes

1. Database creation queries

2. Queries used in PHP files

3. ER Diagram

4. Some Screenshots

DROP DATABASE placement\_portal;

CREATE DATABASE placement\_portal;

USE placement\_portal;

CREATE TABLE students(

roll\_num INT NOT NULL PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50),

std\_course VARCHAR(50) NOT NULL,

std\_branch VARCHAR(50) NOT NULL,

std\_year INT NOT NULL );

INSERT INTO students VALUES (11012322,'Naveen','Chaudhary','Btech','Mathematics',2015);

INSERT INTO students VALUES (11012327,'Prashant','Sankhla','Btech','Civil',2015);

INSERT INTO students VALUES (11012342,'Ravindra','Gahlot','Btech','Biotechnology',2015);

INSERT INTO students VALUES (11012319,'Lokesh','Meena','Btech','CSE',2015);

INSERT INTO students VALUES (11012312,'Parik','Gupta','Btech','Design',2015);

CREATE TABLE std\_registered (

roll\_num INT NOT NULL PRIMARY KEY ,

FOREIGN KEY (roll\_num) REFERENCES students(roll\_num) ON DELETE CASCADE ON UPDATE CASCADE,

resume\_1 MEDIUMBLOB,

resume\_2 MEDIUMBLOB,

CPI decimal(3,2) NOT NULL,

CHECK (CPI >= 5.00));

INSERT INTO std\_registered(roll\_num,CPI) VALUES (11012322,6.52);

INSERT INTO std\_registered(roll\_num,CPI) VALUES (11012327,9.72);

INSERT INTO std\_registered(roll\_num,CPI) VALUES (11012319,5.99);

INSERT INTO std\_registered(roll\_num,CPI) VALUES (11012312,8.26);

CREATE TABLE student\_login (

roll\_num INT NOT NULL PRIMARY KEY ,

FOREIGN KEY (roll\_num) REFERENCES students (roll\_num) ON DELETE CASCADE ON UPDATE CASCADE,

std\_username VARCHAR(50) NOT NULL ,

std\_password VARCHAR(50) NOT NULL);

INSERT INTO student\_login VALUES (11012322,'n.chaudhary','123');

INSERT INTO student\_login VALUES (11012327,'p.sankhla','123');

INSERT INTO student\_login VALUES (11012342,'r.gahlot','123');

INSERT INTO student\_login VALUES (11012319,'l.meena','123');

INSERT INTO student\_login VALUES (11012312,'g.parik','123');

CREATE TABLE student\_emails (

roll\_num INT NOT NULL ,

std\_email VARCHAR(50) NOT NULL ,

PRIMARY KEY (std\_email),

FOREIGN KEY (roll\_num ) REFERENCES students(roll\_num) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO student\_emails VALUES (11012322,'n.chaudhary@iitg.ernet.in');

INSERT INTO student\_emails VALUES (11012327,'p.sankhla@iitg.ernet.in');

INSERT INTO student\_emails VALUES (11012342,'r.gahlot@iitg.ernet.in');

INSERT INTO student\_emails VALUES (11012319,'l.meena@iitg.ernet.in ' );

INSERT INTO student\_emails VALUES (11012312,'g.parik@iitg.ernet.in');

CREATE TABLE students\_phone(

roll\_num INT NOT NULL,

std\_phone BIGINT NOT NULL,

PRIMARY KEY (std\_phone),

FOREIGN KEY (roll\_num ) REFERENCES students(roll\_num) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO students\_phone VALUES (11012322,97067227);

INSERT INTO students\_phone VALUES (11012322,97067364);

INSERT INTO students\_phone VALUES (11012327,98014562);

INSERT INTO students\_phone VALUES (11012342,97064521);

INSERT INTO students\_phone VALUES (11012319,80114961);

INSERT INTO students\_phone VALUES (11012312,80116982);

CREATE TABLE companies (

company\_id VARCHAR(50) NOT NULL PRIMARY KEY,

company\_name VARCHAR(50) NOT NULL,

job\_profile VARCHAR(50) NOT NULL,

c\_password VARCHAR(50) NOT NULL,

c\_email VARCHAR(50) NOT NULL );

INSERT INTO companies VALUES ('goo\_dev','google','developer','234','goo@hotmail.com');

INSERT INTO companies VALUES ('goo\_mar','google','marketing','234','goo@hotmail.com');

INSERT INTO companies VALUES ('fb\_dev','facebook','developer','234','fb@hotmail.com');

INSERT INTO companies VALUES ('ms\_dev','microsoft','developer','234','ms@hotmail.com');

CREATE TABLE std\_applied(

roll\_num INT,

company\_id VARCHAR(50),

placed BIT NOT NULL,

PRIMARY KEY (company\_id,roll\_num),

FOREIGN KEY (company\_id ) REFERENCES companies(company\_id ) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (roll\_num) REFERENCES std\_registered(roll\_num) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO std\_applied VALUES (11012312,'goo\_dev',0);

INSERT INTO std\_applied VALUES (11012312,'goo\_mar',0);

INSERT INTO std\_applied VALUES (11012319,'goo\_mar',0);

INSERT INTO std\_applied VALUES (11012319,'ms\_dev',1);

INSERT INTO std\_applied VALUES (11012322,'fb\_dev',0);

INSERT INTO std\_applied VALUES (11012327,'goo\_dev',0);

INSERT INTO std\_applied VALUES (11012327,'ms\_dev',0);

CREATE TABLE c\_phone(

company\_id VARCHAR(50),

phone BIGINT NOT NULL,

PRIMARY KEY (company\_id,phone),

FOREIGN KEY (company\_id ) REFERENCES companies(company\_id ) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO c\_phone VALUES ('goo\_dev',99576869);

INSERT INTO c\_phone VALUES ('goo\_mar',99576869);

INSERT INTO c\_phone VALUES ('goo\_mar',99574536);

INSERT INTO c\_phone VALUES ('fb\_dev',90856869);

INSERT INTO c\_phone VALUES ('ms\_dev',96549436);

INSERT INTO c\_phone VALUES ('ms\_dev',98653265);

CREATE VIEW stud\_info AS SELECT students.roll\_num, students.first\_name, students.last\_name, students.std\_course, students.std\_branch, students.std\_year, student\_login.std\_username, student\_login.std\_password,std\_registered.resume\_1,std\_registered.resume\_2

FROM students INNER JOIN student\_login ON students.roll\_num=student\_login.roll\_num INNER JOIN std\_registered ON students.roll\_num=std\_registered.roll\_num;

CREATE VIEW company\_info AS SELECT companies.company\_id, companies.company\_name, companies.job\_profile, companies.c\_password, companies.c\_email

FROM companies ;

CREATE VIEW stud\_basic\_info AS SELECT std\_registered.roll\_num, students.first\_name, students.last\_name ,student\_login.std\_username

FROM std\_registered INNER JOIN students ON std\_registered.roll\_num=students.roll\_num INNER JOIN student\_login ON std\_registered.roll\_num=student\_login.roll\_num;

DROP USER 'student\_manager'@'localhost';

CREATE USER 'student\_manager'@'localhost' IDENTIFIED BY 'pass1';

GRANT SELECT, INSERT,DELETE,UPDATE ON placement\_portal.stud\_info TO student\_manager@localhost IDENTIFIED BY 'pass1';

DROP USER 'company\_manager'@'localhost';

CREATE USER 'company\_manager'@'localhost' IDENTIFIED BY 'pass2';

GRANT SELECT, INSERT,DELETE,UPDATE ON placement\_portal.company\_info TO company\_manager@localhost IDENTIFIED BY 'pass2';

***Functional Dependencies and Normal Forms:***

roll\_num : RN company\_id : CID

first\_name : FN company\_name : CN

last\_name : LN c\_password : CP

std\_username : SU c\_email : CE

std\_password : SP phone : CPH

std\_phone : SPH job\_profile : JP std\_course : SC

std\_branch : SB placed : PD

std\_email : SE

std\_year : SY

resume\_1 : R1

resume\_2 : R2

CPI : C

1. Students Table :

RN ---> RN FN LN SC SY SB

Since there is a PRIMARY KEY, no partial or transitive dependencies and no overlapping candidate keys

Therefore it is in BCNF

2. Registered Students Table :

RN ---> RN R1 R2 C

Since there is a PRIMARY KEY, no partial or transitive dependencies and no overlapping candidate keys

Therefore it is in BCNF

3. Students Application Table :

RN CID ---> RN CID PD

Since there is a PRIMARY KEY, no partial or transitive dependencies and no overlapping candidate keys

Therefore it is in BCNF

4. Companies Table :

CID ---> CID CN CP CE JP

Since there is a PRIMARY KEY, no partial or transitive dependencies and no overlapping candidate keys

Therefore it is in BCNF

5. Company Phone Table :

CPH CID--->CPH CID

Since there is a PRIMARY KEY, no partial or transitive dependencies and no overlapping candidate keys

Therefore it is in BCNF

6. Student Login Table

RN ---> RN SU SP

SU ---> RN SU SP

Since there is a primary key, therefore It is in 1NF

Since there are no partial dependencies, therefore it is in 2NF

But since there is a transitive dependency (RN--->SU , SU--->SP) , therefore it is not in 3NF

7. Student Phone Table

SPH ---> SPH RN

Since there is a PRIMARY KEY, no partial or transitive dependencies and no overlapping candidate keys

Therefore it is in BCNF

7. Student Email Table

SE ---> SE RN

Since there is a PRIMARY KEY, no partial or transitive dependencies and no overlapping candidate keys

Therefore it is in BCNF

Queries that we used for fetching and updating the database :

We first created views using inner join then we used these views in the queries.

§ CREATE VIEW **stud\_info** AS SELECT students.roll\_num, students.first\_name, students.last\_name, students.std\_course, students.std\_branch, students.std\_year, student\_login.std\_username, student\_login.std\_password,std\_registered.resume\_1,std\_registered.resume\_2

FROM students INNER JOIN student\_login ON students.roll\_num=student\_login.roll\_num INNER JOIN std\_registered ON students.roll\_num=std\_registered.roll\_num;

§ CREATE VIEW **company\_info** AS SELECT companies.company\_id, companies.company\_name, companies.job\_profile, companies.c\_password, companies.c\_email FROM companies ;

§ CREATE VIEW **stud\_basic\_info** AS SELECT std\_registered.roll\_num, students.first\_name, students.last\_name ,student\_login.std\_username

FROM std\_registered INNER JOIN students ON std\_registered.roll\_num=students.roll\_num INNER JOIN student\_login ON std\_registered.roll\_num=student\_login.roll\_num;

1. Companies.php

$sql = "select \* from companies ";

To print the list companies with their profiles

2. Company\_insert.php

$sql = "insert into companies(company\_id,company\_name,job\_profile,c\_password,c\_email) values('".$CompanyID."','".$CompanyName."','".$JobProfile."','".$Password."','".$Email."')";

$sql2 = "insert into c\_phone(company\_id,phone) values('".$CompanyID."','".$Mobile."')";

To insert the data of the new registering company into companies and c\_phone respectively

3. Login.php

$sql = "select \* from company\_info where company\_id='".$UserName."' and c\_password='".$Password."'";

$sql = "select \* from stud\_info where std\_username='".$UserName."' and std\_password='".$Password."'";

For login of companies and students respectively

4. New\_reg\_stud\_Insert.php

$sql = "SELECT \* from student\_login where std\_username='$UserName'";

$sql = "insert into std\_registered(roll\_num,resume\_1,resume\_2,CPI) values('".$RollNum."','".$path1."','".$path2."','".$Cpi."')";

To insert the data of the new registering student

5. Company -> Profile.php

$sql = "select \* from company\_info where company\_id ='".$ID."' ";

$sql = "select \* from c\_phone where company\_id ='".$ID."' ";

To display company profile

6. Company -> EditProfile.php

$sql = "select \* from company\_info where company\_id ='".$ID."' ";

$sql = "select \* from c\_phone where company\_id ='".$ID."' ";

$sql = "select \* from company\_info where company\_id ='".$ID."' ";

For editing company info

7. Company ->UpdateProfile.php

$sql = "Update companies set c\_email='".$txtEmail."',job\_profile='".$txtArea."',c\_password='".$txtPassword."'

Where company\_id='".$txtId."'";

To update the company profile

8. Student -> Profile.php

$sql = "select \* from stud\_info where roll\_num='".$ID."' ";

$sql = "select \* from students\_phone where roll\_num ='".$ID."' ";

$sql = "select \* from student\_emails where roll\_num ='".$ID."' ";

To display the student profile

9. Student ->EditProfile.php

$sql = "select \* from students where roll\_num ='".$ID."' ";

$sql = "select \* from students\_phone where roll\_num ='".$ID."' ";

$sql = "select \* from student\_emails where roll\_num ='".$ID."' ";

$sql = "select \* from std\_registered where roll\_num ='".$ID."' ";

$sql = "select \* from student\_login where roll\_num ='".$ID."' ";

To edit student info

10. Student -> register\_ajax.php

$sql="select roll\_num from stud\_basic\_info where std\_username='$add[1]'";

$sql="insert into std\_applied (roll\_num,company\_id) values('$roll','$add[0]') ";

To register a student for a company

11. Student -> Show\_detail.php

$sql="select roll\_num from stud\_basic\_info where std\_username='$add[1]'";

$sql=" select company\_id from std\_applied where roll\_num=$roll and company\_id='$add[0]'";

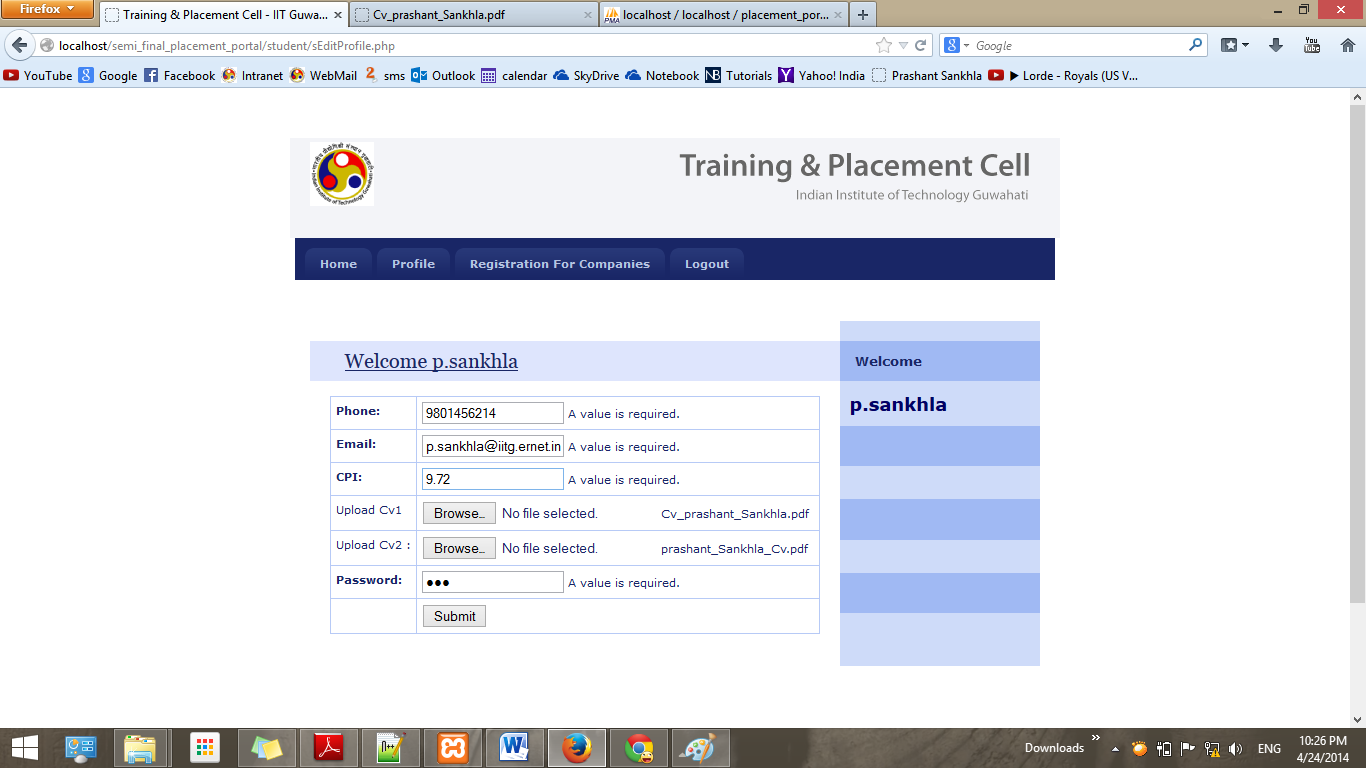
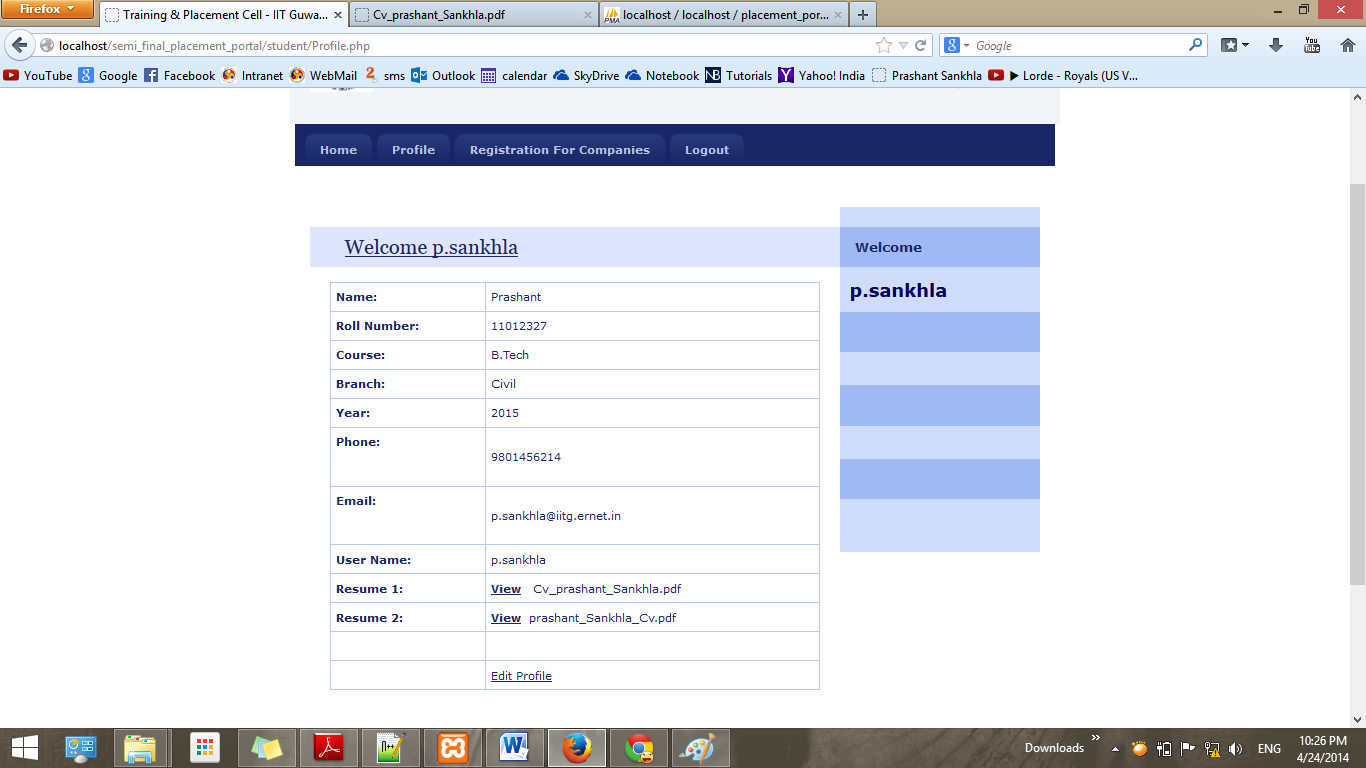
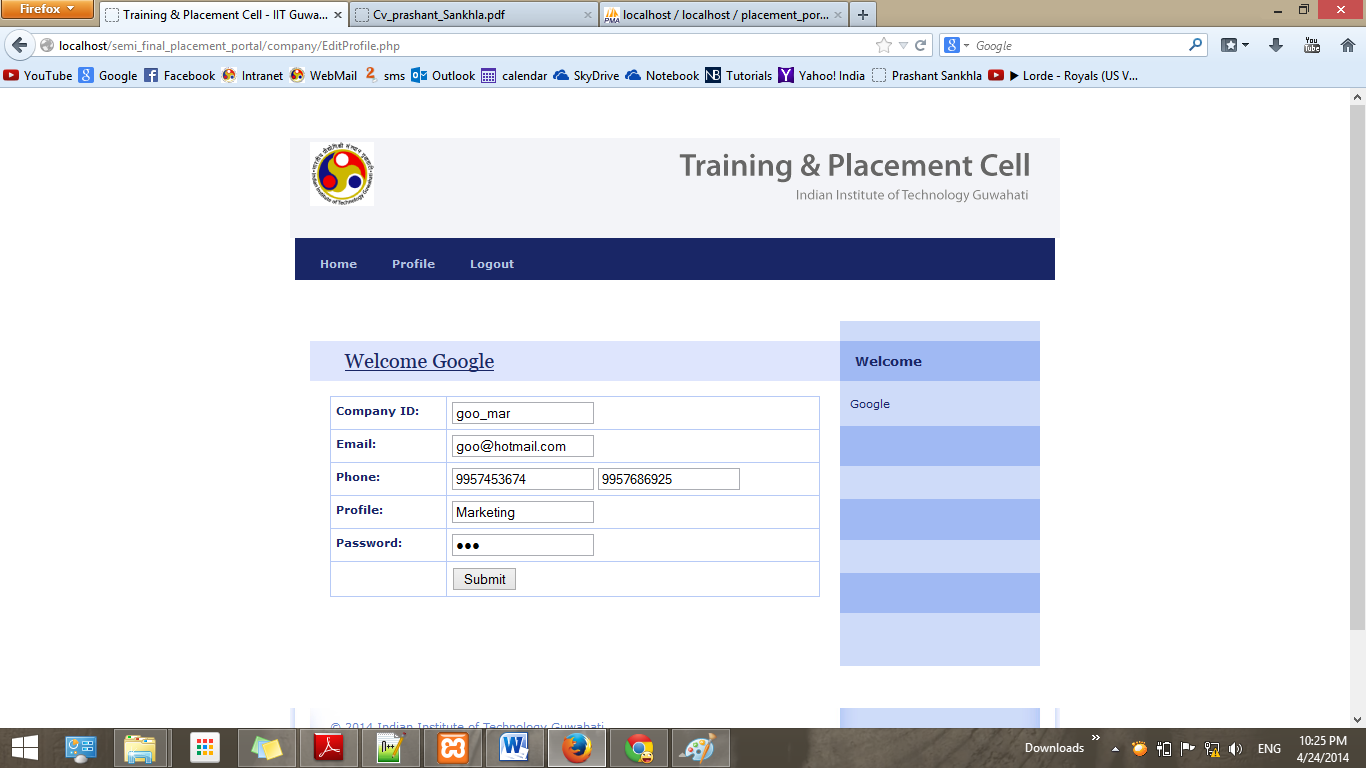
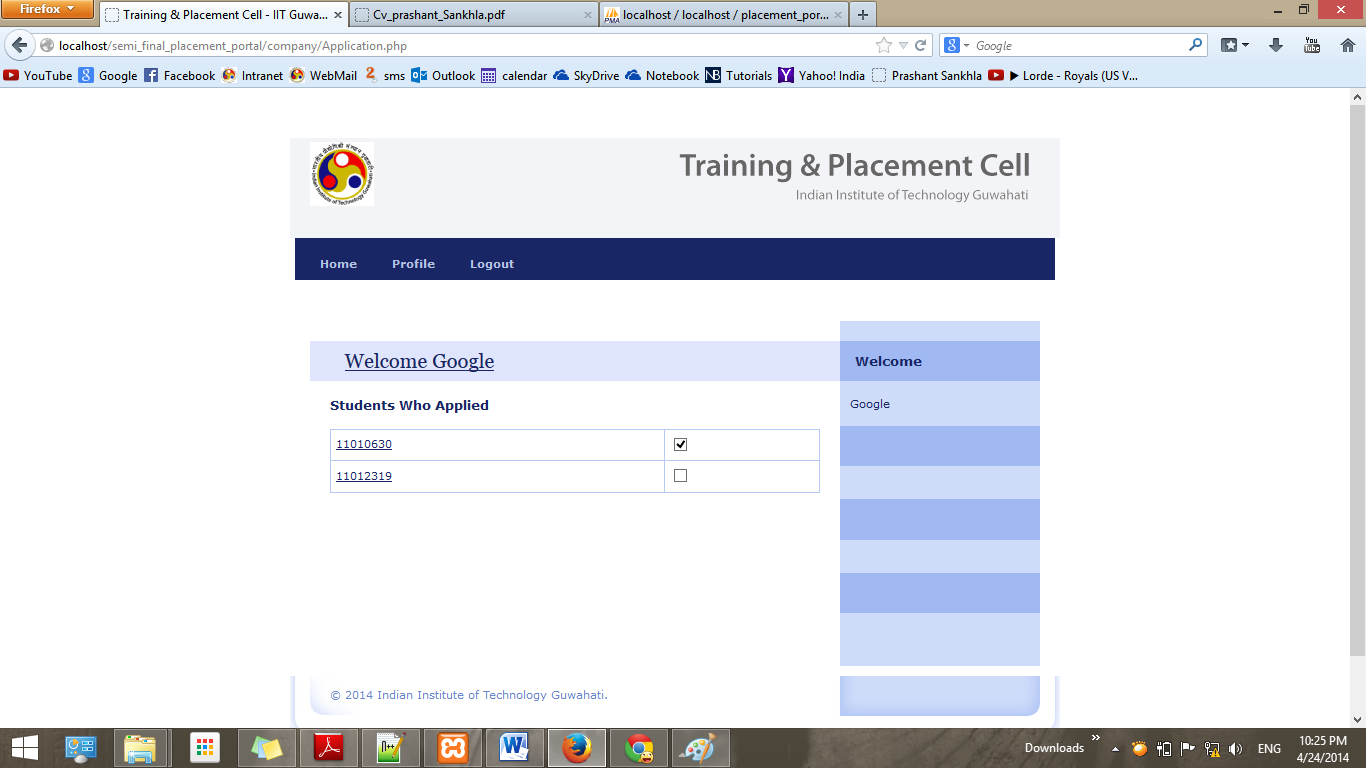
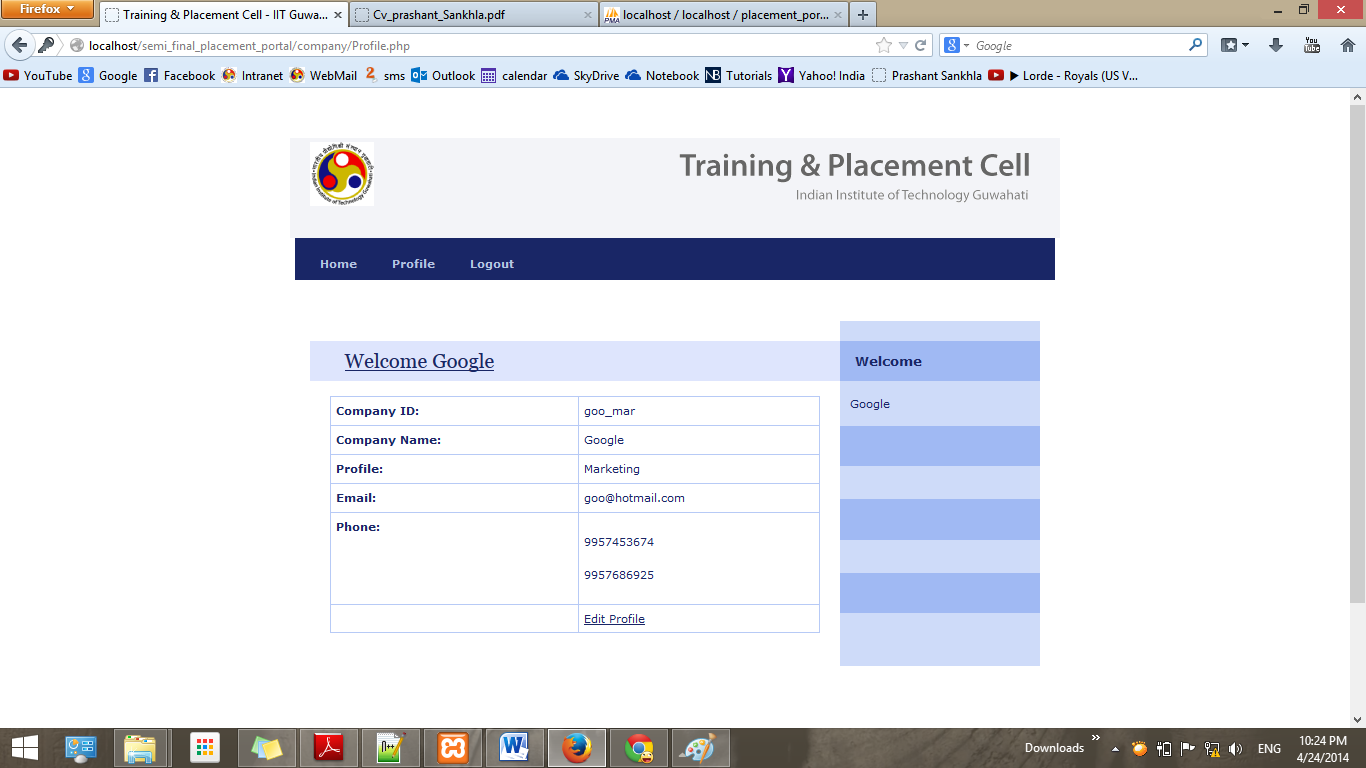
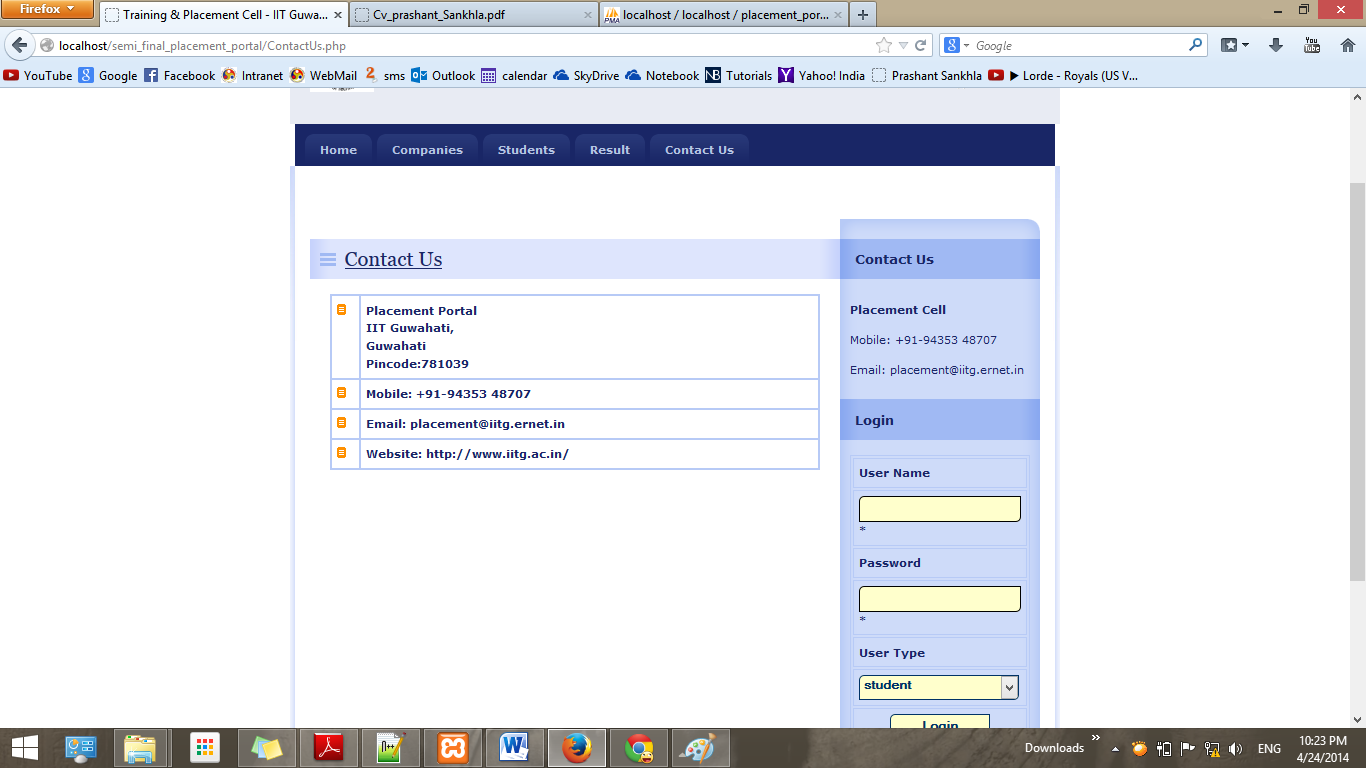
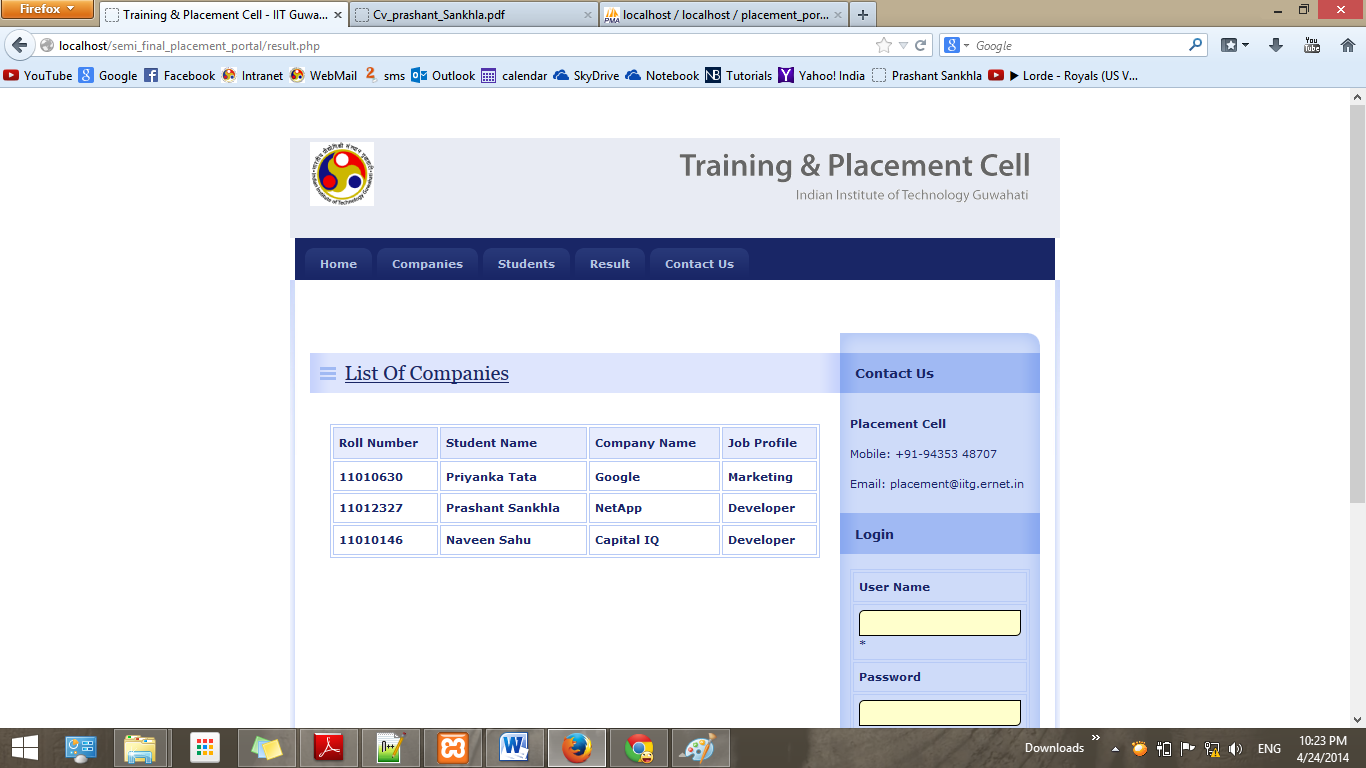
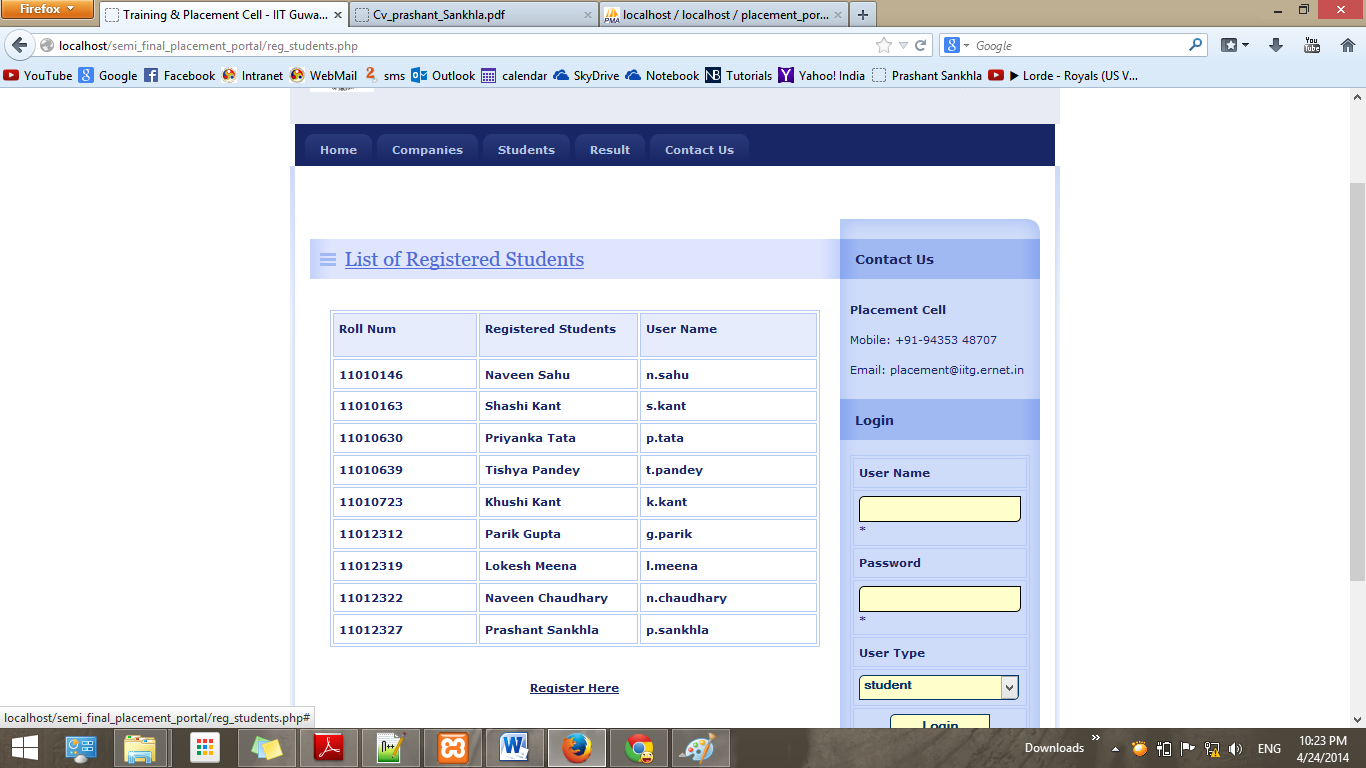
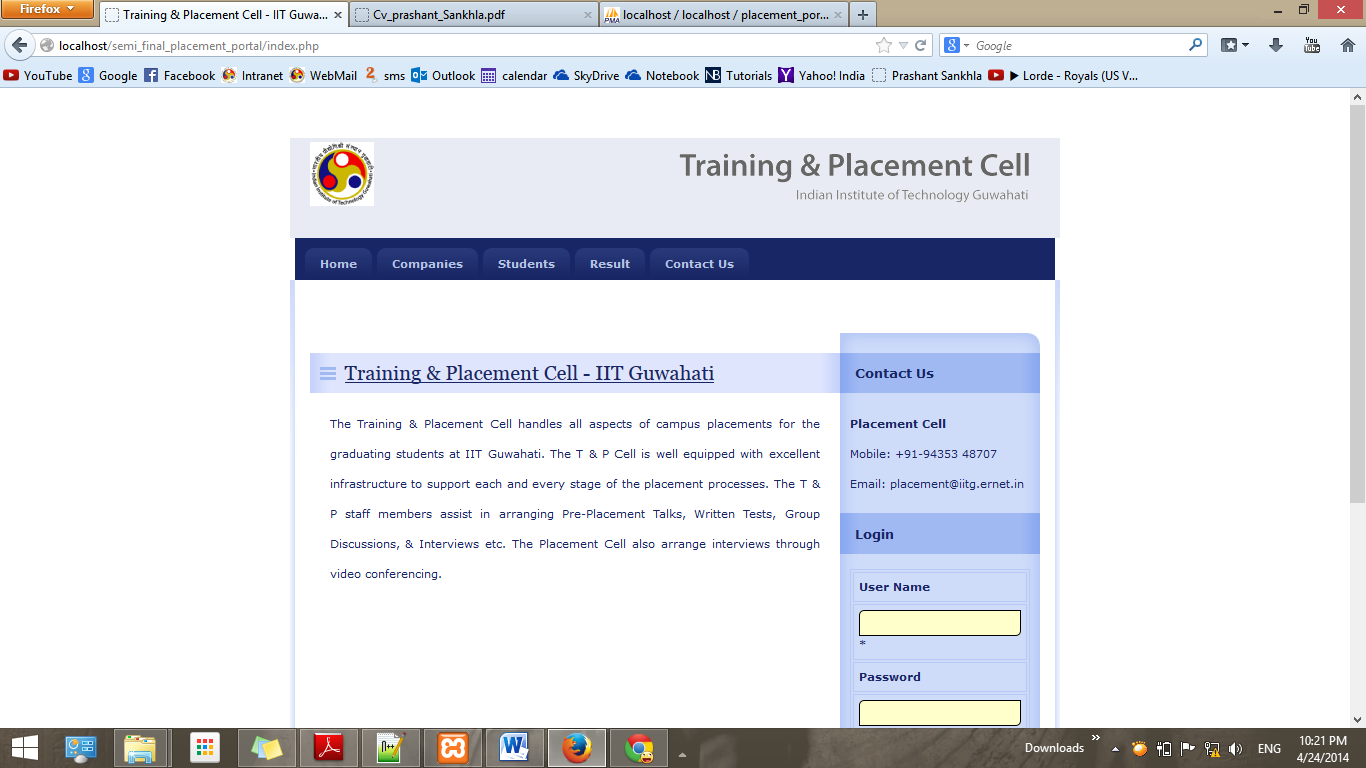
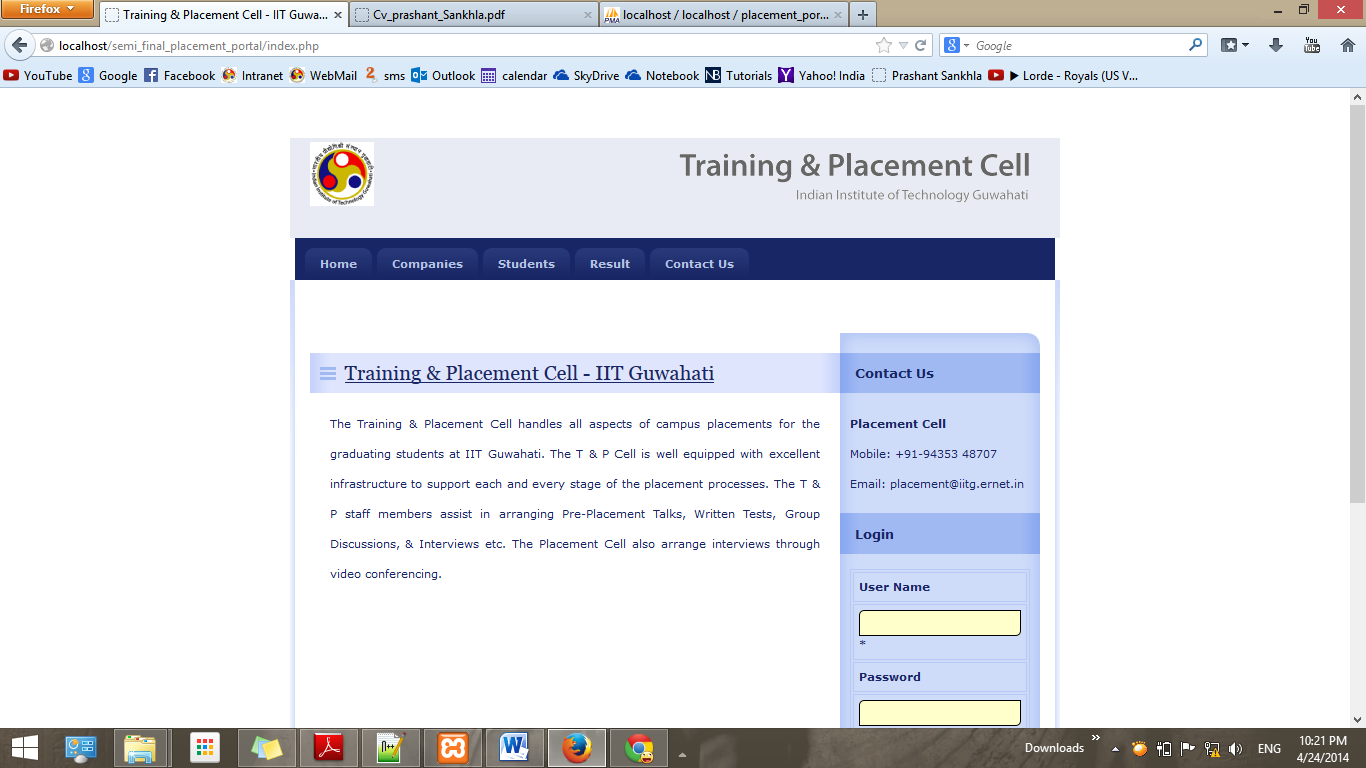
12. Student -> Unregister\_ajax.php

$sql=" select roll\_num from stud\_basic\_info where std\_username='$add[1]'";

$sql="delete from std\_applied where roll\_num=$roll and company\_id='$add[0]' " ;

To unregister a student from a company

***Some Screen Shots :***

******