

# **HOSTEL MANAGEMENT SYSTEM**

## **TEAM MEMBERS**

**SHUBHAM PARASHAR-15BCE1318**

**OSHO AGYEYA-15BCE1326**

**SHRIANSH SRIVASTAVA-15BCE1314**

**KASHISH MIGLANI-15BCE1003**

**ANIMESH SENGUPTA-15BCE1198**

## **FINAL REVIEW DOCUMENT**



**SUBMITTED TO:-**

**GRACELINE MA'AM**

**CSE DEPARTMENT**

**VIT UNIVERSITY, CHENNAI CAMPUS**

## **ACKNOWLEDGEMENT**

**We would like to express our special thanks of gratitude to our teacher Graceline Jasmine Ma'am as well as our program chair Dr R Ganesan sir who gave us the golden opportunity to do this wonderful project on the topic Hostel Management System for college events which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.**

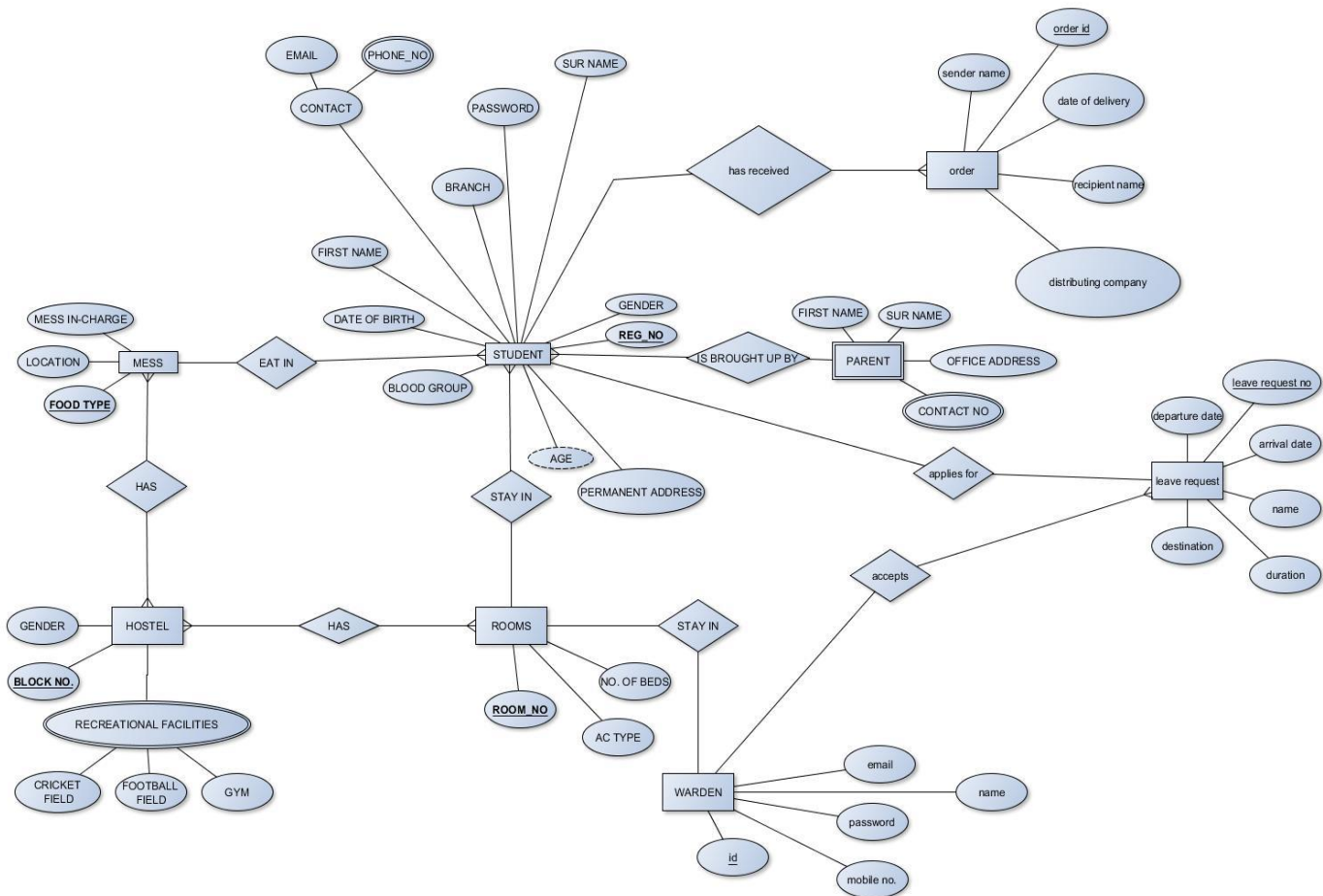
**Secondly we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.**

## **ABSTRACT**

The online hostel management system works on the backbone of database and tables. The website has been developed to do away with old and obsolete system of website and bring a new website with new, better and up-to date functionalities such as search by clause, leave form acceptance, online order receiving, dashboard for messages etc.

The website uses the back end and database to fetch results.

## ER – MODEL



The ER-model has been included for the understanding of the tables and their dependencies. The relations have been clearly mentioned and the dependencies and relations are given.

The student as an entity does the following-:

Eats in MESS

Stays in ROOM

Receives ORDER

Applies for LEAVE REQUEST

Similarly, all the functionalities of each entity have been mentioned in the E-R diagram.

The various attributes of the entities has also been mentioned clearly to understand each table and their column in the actual database.

The other entities that are included are-:

- Mess
- Room
- Order
- Hostel

For each of the given entity the attributes and relations have been mentioned.

## MODULES/FUNCTIONALITIES

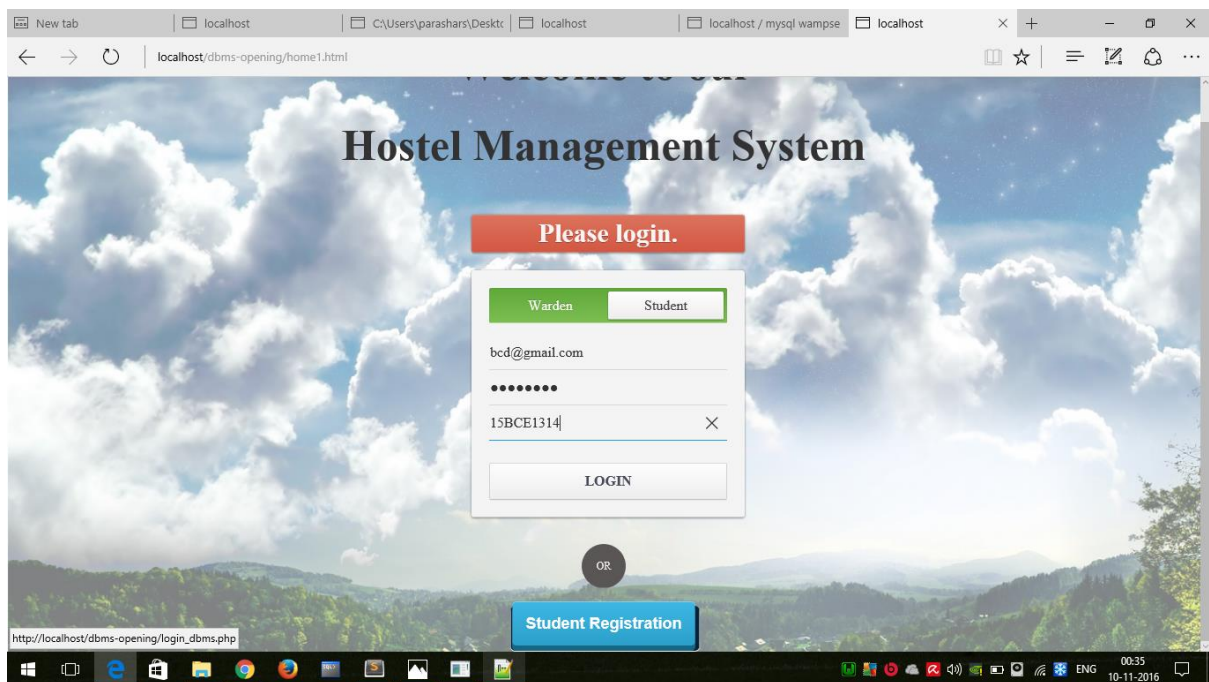
### Module-1

The first module of our website comprises of Welcome page user sign-up and login. The user sign-up and login have been validated using JavaScript. The sign-up enables a user to be a member of the website and from then onwards the user can login and access the various functionalities of the website.

The sign-up and login form have been styled using CSS.

### Welcome Page and Login

The home page has been styled using CSS and allows the user to login and register for the website.



## **Sign Up Page**

The sign-up page has been dynamically designed using HTML and CSS and has user validation with it. The sign-up is available for user as well as admin.

**Enter Student Details:**

**\*\*All fields are mandatory**

First Name:

Last Name:

College:

E-mail:

Password:

Retype Password:

What was the first word you said?

## **Module-2**

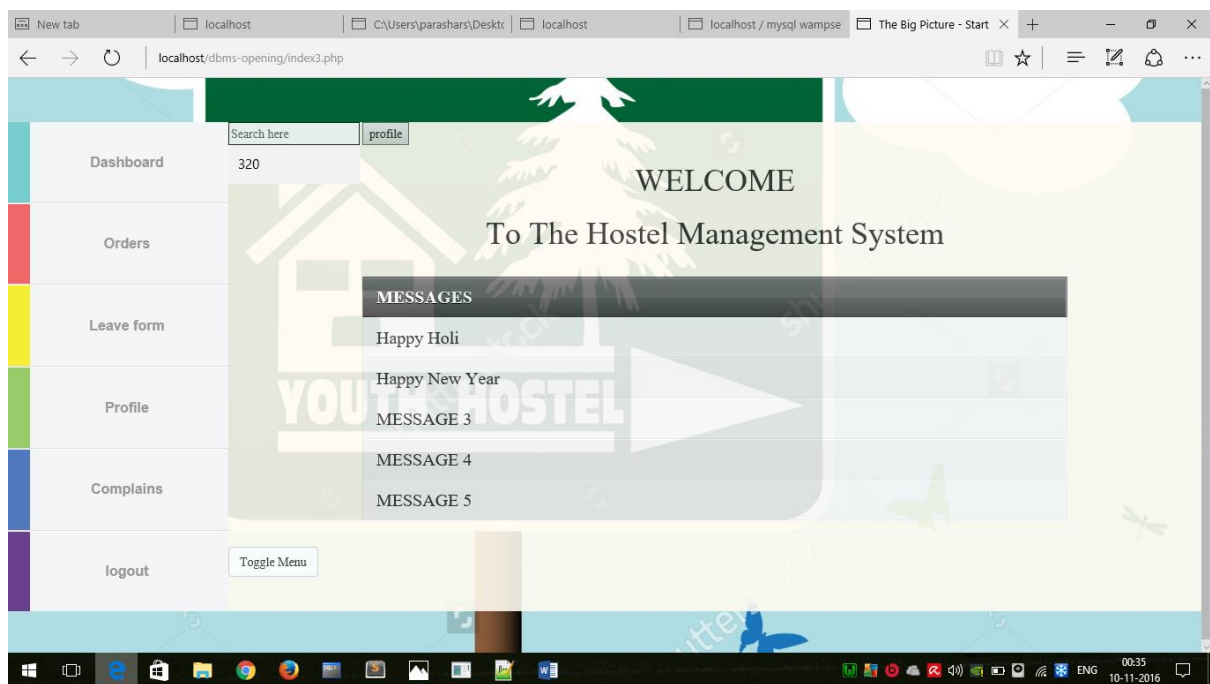
The module comprises of the inner part of the website which is the home page, admin homepage, the overall functionality, the menu bar and user profile.

The user gets to interact with the website through this module and can view the functionalities of the website. It allows the user to view events by college or date and also lets the user rate them and know about them.

The users can also view their profile.

## Home page

The home page comprises of a dynamically designed filter and a dashboard featuring all the major events of the hostel that have been covered in the website. The filter can change the dashboard according to the college selected. The user can also click on the event button to view the desired thing.

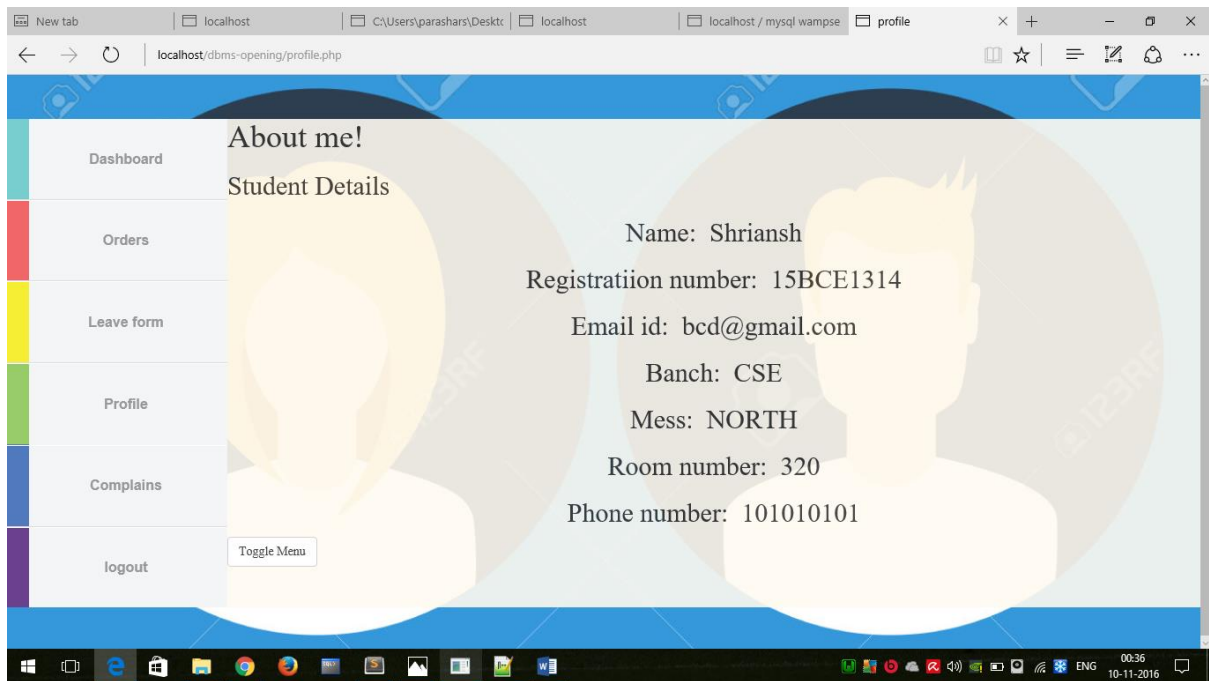


## User Profile

- Users can view their profile by clicking on the view profile button and in his profile they can view the following things-
  - A. First Name
  - B. Last Name
  - C. Room number
  - D. Mess type



## E. Room Type

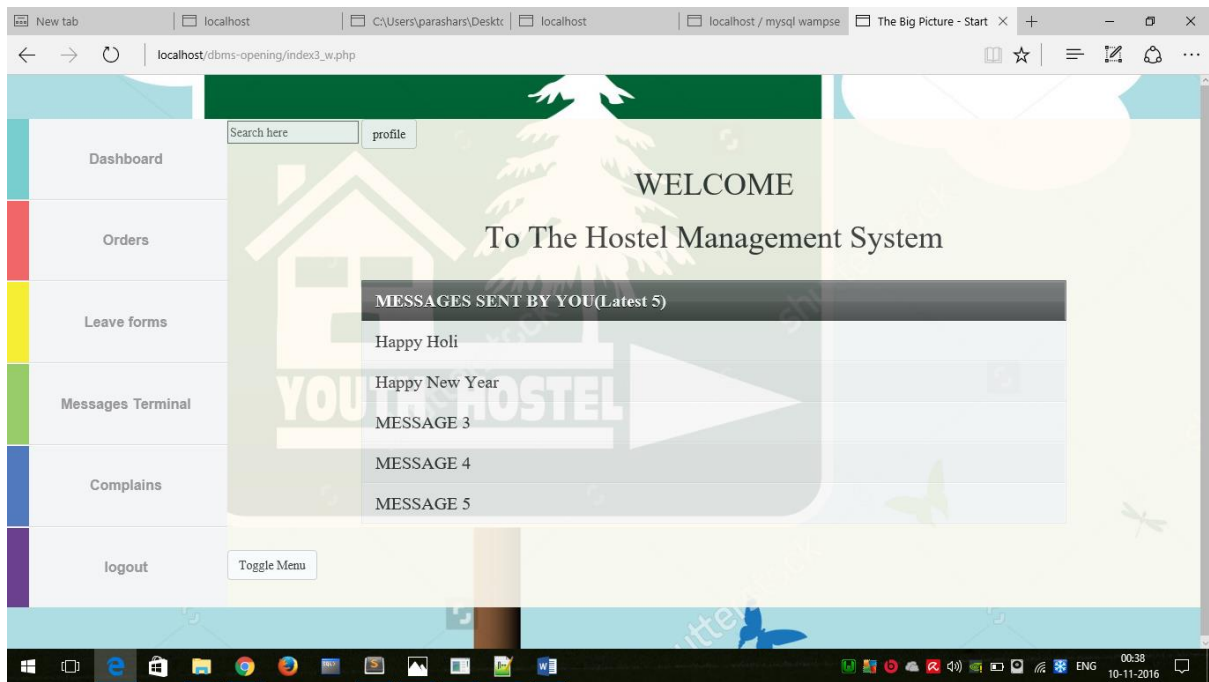


## Admin Homepage

This page comprises of all the admin functionality that is

- He/she can search accept leave forms, search students by query, see complaints, online orders acceptance.

At last there is the logout button.



### **MODULE-3**

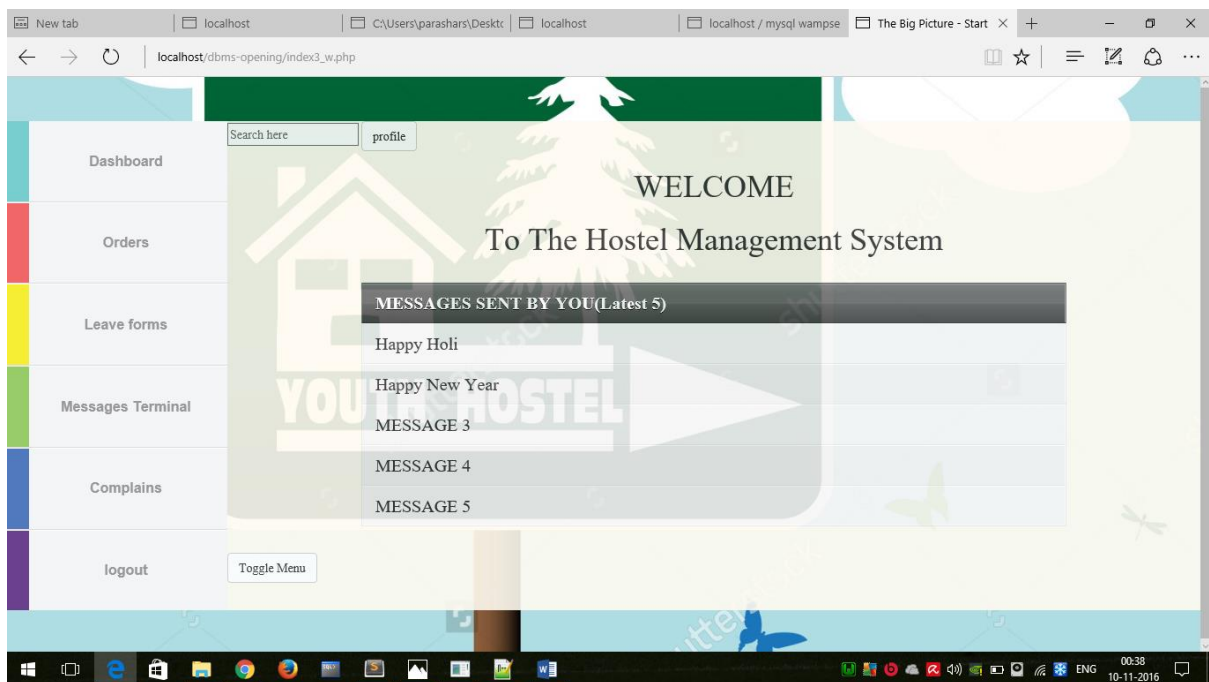
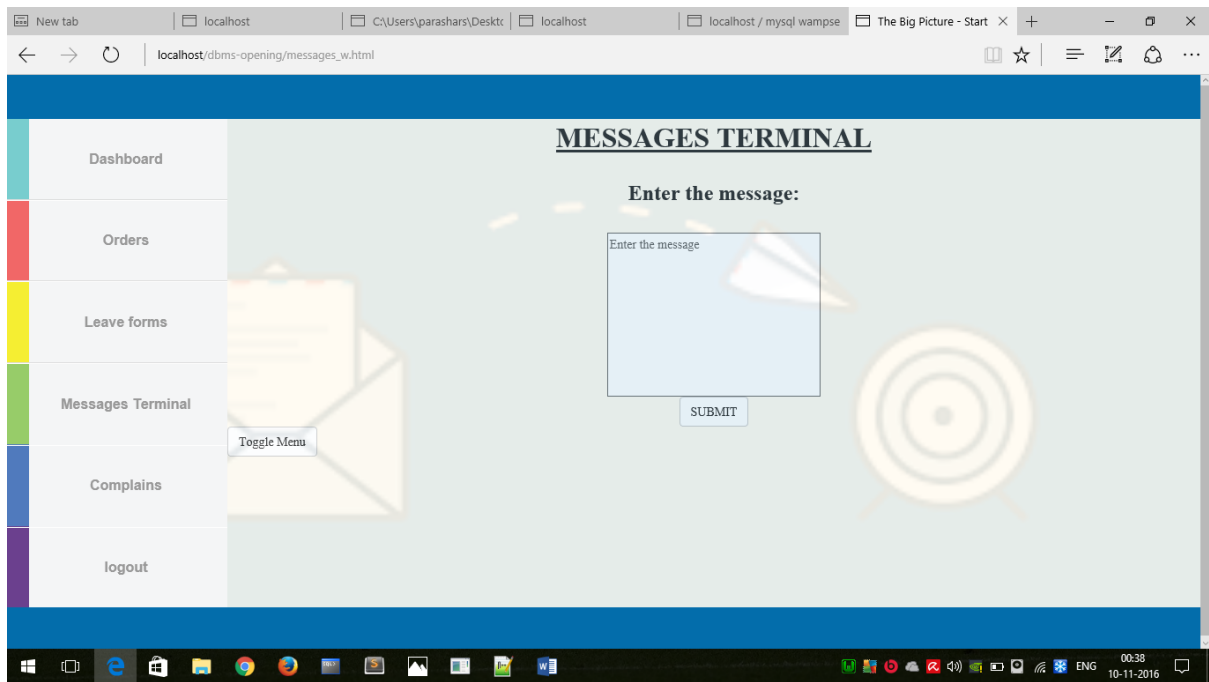
The third module comprises of the inner important functionalities of the website such as searching a student by query, dashboard updating, hostel complaints, viewing status of online orders, applying for leave forms etc.

It also includes the overall working of the website and the basic functionality of the website.

#### **Dashboard/Messages**

The basic portal for the user to see and view all the functionalities of the website, and to also see the messages the student may have got. The dashboard is basically the backbone of the home page.

The messages are also fetched from the database and are presented to the user.



## Search by Query

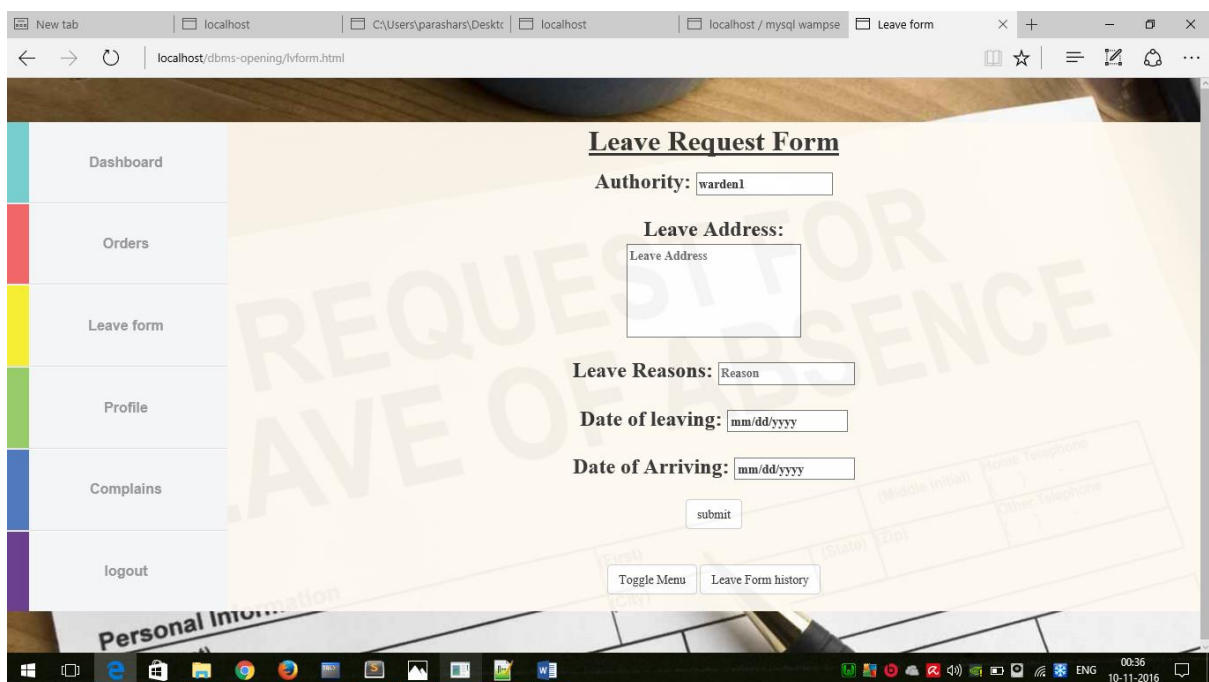
It is based purely on database where a person can search hostel members by clause such as registration number, name, mess type etc. It uses database fetching on condition functionality.

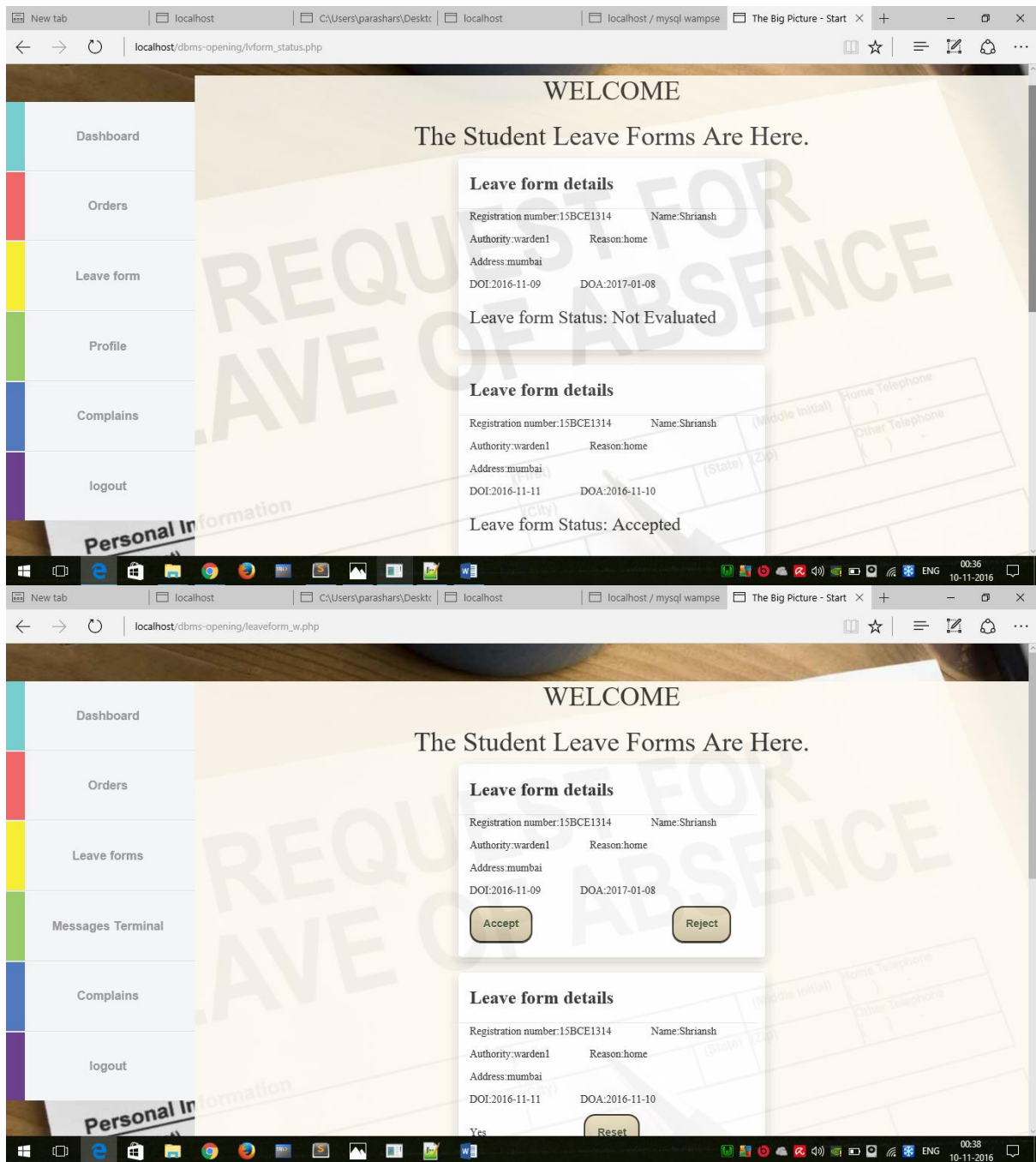


## Leave forms

The leave form has been included so that the student can apply for it online and the warden can get the applied form and accept or reject it.

It works on the table functionality only, where the student's applied response is stored in a table and the same table result is fetched to the warden home page so that he/she can accept or reject the form.





## Hostel Complaints

The hostel complaints have been added so that students can give their complaints of the hostel of various kinds to the warden and the warden can take required action by accepting or rejecting the complaint.

The complaint table is again brought into picture and the result is fetched to the warden home page.

It eases the complaint procedure of the students.

Register Complaints here

Name:

Registration Number:

Room Number:

Complaint type:

Describe Your Complaint.

submit

WELCOME

The Student Complaints are here.

Reg_No	Room	Name	Complain Type	Comp
15BCE1314	320	Shriansh	Washroom	tatti gandhi
15BCE1314	320	Shriansh	AC type	not cooling
15BCE1314	320	Shriansh	Water Problems	no water
15BCE1314	320	Shriansh	Cleaning	Not cleaning

Toggle Menu



## Online Orders

The online order functionality has been added so that the student could know the status about his online orders on the website itself instead of physically going and checking again and again.

The online order table is updated each time the user has an order for him/her.

The screenshot shows a web browser window with the URL `localhost/dbms-opening/orders_w.html`. The page features a sidebar menu on the left with options: Dashboard, Orders, Leave forms, Messages Terminal (highlighted in green), Complains, and logout. The main content area is titled **ORDERS** and contains a form with the following fields: Student Name, Registration Number, Date Of Arrival (with a placeholder `mm/dd/yyyy`), Address, and RoomNumber. Below the form is a 'Submit order' button. At the bottom of the form area, there are two buttons: 'Toggle Menu' and 'Order History'. The browser's taskbar at the bottom shows various application icons and the system clock indicating 00:39 on 10-11-2016.

The screenshot shows a web browser window with the URL `localhost/dbms-opening/order_accept.php`. The page features the same sidebar menu as the previous screenshot. The main content area is titled **Your Orders** and displays a confirmation message: **Order Collected**. Below this message is a table with the following details:

Registration number:15BCE1314	Name:Shriansh
Address:Pune	Room:320
DOA:2016-11-09	

Below the table is a 'Toggle Menu' button. The browser's taskbar at the bottom shows various application icons and the system clock indicating 00:36 on 10-11-2016.

## Logout

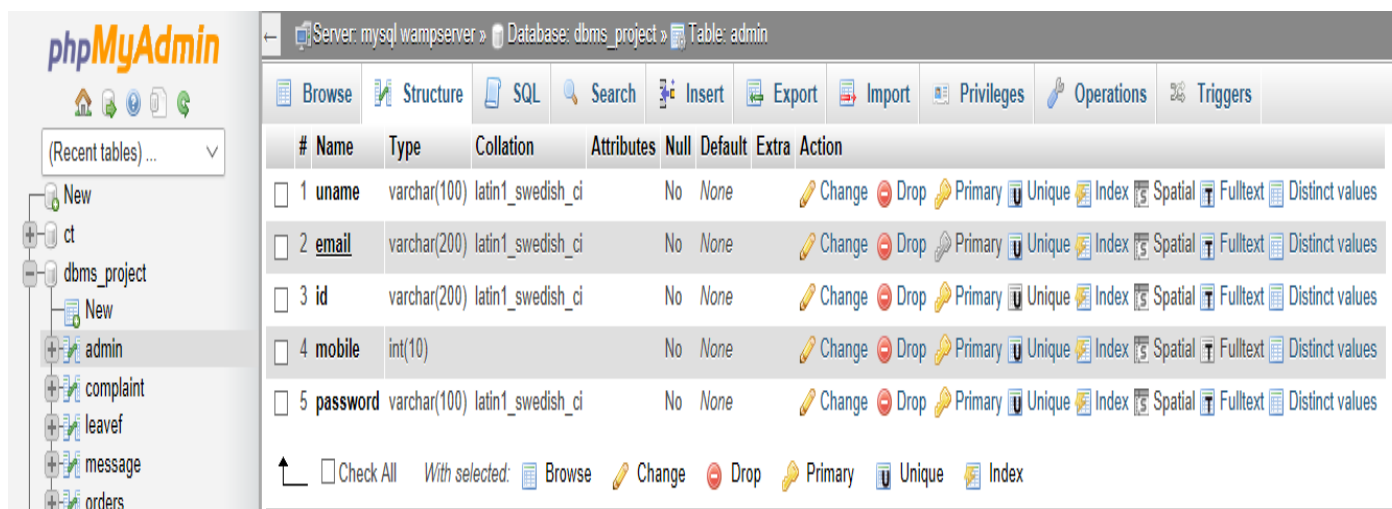
Finally, the logout button is there to get out of the website and active session.

## SCHEMA/STUCTURE OF TABLES USED

The structure of table is mentioned along with columns etc. to understand the database schema completely.

Each table has different columns which are mentioned clearly.

## ADMIN TABLE



The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a tree view of databases and tables. The 'dbms\_project' database is selected, and the 'admin' table is highlighted. The main panel displays the 'Structure' tab for the 'admin' table. It shows a table with 5 columns: 'uname', 'email', 'id', 'mobile', and 'password'. Each column has a checkbox for selection and a set of icons for actions like Change, Drop, Primary, Unique, Index, Spatial, Fulltext, and Distinct values. The 'id' column is marked as the primary key.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 <u>uname</u>	varchar(100)	latin1_swedish_ci		No	None		
<input type="checkbox"/>	2 <u>email</u>	varchar(200)	latin1_swedish_ci		No	None		
<input type="checkbox"/>	3 <u>id</u>	varchar(200)	latin1_swedish_ci		No	None		
<input type="checkbox"/>	4 <u>mobile</u>	int(10)			No	None		
<input type="checkbox"/>	5 <u>password</u>	varchar(100)	latin1_swedish_ci		No	None		

↑ ☐ Check All With selected:



## COMPLAINT TABLE

(Recent tables) ...	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
New	1	name	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
ct	2	regno	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
dbms_project	3	room	int(11)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
New	4	compt	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
admin	5	comp	text	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
complaint									
leavef									
message									
orders									

☐ Check All    With selected: Browse Change Drop Primary Unique Index

## LEAVE FORM TABLE

(Recent tables) ...	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
New	1	id	varchar(250)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
ct	2	name	varchar(100)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
dbms_project	3	auth	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
New	4	ladd	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
admin	5	reason	varchar(500)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
complaint	6	dol	date			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
leavef	7	doa	date			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
message	8	temp	int(11)			No	None	AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
orders	9	flag	int(11)			No	0		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  More
registration									
information_schema									
iwp									

## MESSAGES TABLE

(Recent tables) ...	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
New	1	messages	text	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
ct	2	mes_date	datetime			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
dbms_project									
New									
admin									

☐ Check All    With selected: Browse Change Drop Primary Unique Index

## ORDERS TABLE

(Recent tables) ...	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
New	1	name_stud	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  More
ct	2	reg_no	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  More
dbms_project	3	doa	date			No	None		Change  Drop  Primary  Unique  Index  Spatial  More
New	4	address	varchar(200)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  More
admin	5	room_stud	int(11)			No	None		Change  Drop  Primary  Unique  Index  Spatial  More
complaint	6	flag	varchar(200)	latin1_swedish_ci		Yes	NULL		Change  Drop  Primary  Unique  Index  Spatial  More
leavef	7	num	int(11)			No	None	AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  Spatial  More
message									
orders									
registration									
information_schema									

☐ Check All    With selected: Browse Change Drop Primary Unique Index

REGISTRATION TABLE

(Recent tables) ...

New

ct

dbms\_project

New

admin

complaint

leavef

message

orders

registration

information\_schema

iwp

mysql

performance\_schema

software\_lab

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	namec	varchar(250)	latin1_swedish_ci	No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	2	email	varchar(250)	latin1_swedish_ci	No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	3	reg_no	varchar(100)	latin1_swedish_ci	No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	4	branch	varchar(100)	latin1_swedish_ci	No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	5	room_no	decimal(4,0)		No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	6	room_type	varchar(250)	latin1_swedish_ci	No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	7	bed_type	varchar(100)	latin1_swedish_ci	No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	8	mess	varchar(100)	latin1_swedish_ci	No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	9	psswrđ	varchar(250)	latin1_swedish_ci	No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	
<input type="checkbox"/>	10	phone	int(10)		No	None	Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values	

☐ Check All

With selected: Browse Change Drop Primary Unique Index

## **Normalisation of Tables**

## REGISTRATION TABLE

	namec	email	reg_no	branch	room_no	room_type	bed_type	mess	psswr	phone
admin										
leavef										
registration										
	Aayush Gupta	aayush.gupta2015@vit.ac.in	15bec1214	ECE	319	AC	4 Bed	Special	ttt	0
	animesh	animeshsengupta@gmail.com	15BCE1198	MECH	315	AC	3-Bed	south-veg	opijh	0
	Ayush Sharma	ayush.sharma2015@vit.ac.in	15CSE1335	CSE	330	Non-AC	4-bed	Special	ooo	0
	blessvin	blessvinchrister@gmail.com	15BCE1333	ECE	1011	Non-AC	2-Bed	South	iuui	0
	Abhishek Deshmukh	deshmukhabhishek.randhvir@vit.ac.in	15bme1008	MECH	315	AC	3-Bed	North-Non-Veg	rrr	0
	ddixit	dhruv.dixit999@vit.ac.in	15ECE1324	ECE	330		4-Bed	North-Non-Veg	iii	931239494
	Har Shobhit Dayal	harshobhit.dayal2015@vit.ac.in	15BCE1192	CSE	316	AC	3-Bed	Special	qqq	0
	justus	justusvmc@gmail.com	15BCE1043	CSE	322	AC	3-Bed	special	llll	0
	Kajale Omkar Avinash	kajaleomkar.avinash@vit.ac.in	15BCL1039	BCL	319	AC	4-Bed	South-Veg	yyy	0
	kashish	kashishmiglani10@gmail.com	15BCE1003	CSE	713	AC	2-Bed	special	qwewre	0
	ketan	ketanatri10@gmail.com	15BCE1223	BCL	814	AC	2-Bed	North	akansha	0
	mumma	mumma@gmail.com	898998	ll	0		a	o	oopp	99999
	osho	oshoagyeya@gmail.com	15BCE1326	CSE	713	AC	2-Bed	special	fdjhfdh	0
	osho	oshopop@gmail.com	1234	qwe	90		io	io	osho	909009009
	ovpc	ovpc@gmail.com	15BCE1023	MECH	322	Non-AC	4-Bed	North	oooo	0
	Rupam Sarma	rupam.sarma@vit.ac.in	15BCE1069	CSE	319	AC	4-Bed	South-Non-Veg	uuu	0
	Sachin Gopal	sachin.gopal2015@vit.ac.in	15BCE1188	CSE	316	AC	3-Bed	Special	ppp	0
	Sanchay Gupta	sanchay.gupta2015@vit.ac.in	15BCL1192	BCL	316	AC	3-Bed	Special	www	0
	Shivam Kapoor	shivam.kapoor@vit.ac.in	15BCE1190	ECE	315	AC	3-Bed	North-Non-Veg	eee	0
	shubham	shri@gmail.com	15bce131444	cse	212		3	special	ffgghh	2147483647
	shriansh	shrianshshrivastava@gmail.com	15BCE1314	CSE	320	AC	4-Bed	North-Veg	cxcvbn	0
	shubham	shubhamprashar@gmail.com	15BCE1318	BCL	320	AC	4-Bed	special	asddf	0
	tushar	tusharpahuja@gmail.com	15BCE1090	CSE	714	AC	4-Bed	special	osho	0
	utsav	utsavrai@gmail.com	15BCE1352	CSE	430	NON-AC	3-Bed	North	talwar	0

Given functional dependencies for table registration:

reg\_no->namec, email, branch, psswr, phone, room\_no, mess

room\_no->room\_type, bed\_type

1NF:

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore it is in 1<sup>st</sup> normal form.

2NF:

A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is reg\_no. Therefore non-prime attributes are namec, email, branch, psswr, phone, room\_no, mess.

phone, room\_no, mess ,room\_type, bed\_type. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

### **3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute.

Here, non prime attributes are nameec, email, branch, psswr, phone, room\_no, mess ,room\_type, bed\_type, but room\_type,bed\_type are determined by room\_no . Therefore the table is not in 3NF. Now, to convert it into 3NF, we break the table registration as follows:

registration=

registration1(reg\_no,nameec,email,branch,psswr,phone,room\_no,mess)

Its functional dependencies are:

reg\_no->nameec, email, branch, psswr, phone, room\_no,mess

+

registration2(room\_no,room\_type,bed\_type)

Its functional dependencies are:

room\_no->room\_type,bed\_type

Now the table is in 3NF.

### **BCNF:**

A relation is in BCNF if it is in 3NF and for each  $X \rightarrow Y$  , X is a super key.

For table

registration1(reg\_no,nameec,email,branch,psswr,phone,room\_no,mess)

Its functional dependencies are:

reg\_no->nameec, email, branch, psswr, phone, room\_no,mess

Here reg\_no is a super key. Therefore this table is in BCNF.

For table registration2(room\_no,room\_type,bed\_type)

Its functional dependencies are:

room\_no->room\_type,bed\_type

Here room\_no is the superkey. Therefore this table is also in BCNF.

---

### **ADMIN TABLE**

admin	uname	email	id	mobile	password	
leavef	warden	warden@gmail.com	12we	999900	opop	
registration						

**Given functional dependencies for table admin:**

**id->uname,email,mobile,password**

#### **1NF:**

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore in 1<sup>st</sup> normal form.

#### **2NF:**

A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is id. Therefore non-prime attributes are uname, email,mobile,password. All of

them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

### **3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are uname,email,mobile,password and all are determined by the candidate key only. Therefore the table is in 3NF.

### **BCNF:**

A relation is in BCNF if it is in 3NF and for each  $X \rightarrow Y$ ,  $X$  is a super key. Here for table admin, id is a super key. Therefore the table is in BCNF.

---

## LEAVE FORM TABLE

admin	id	name	auth	ladd	reason	dol	doa	temp	flag
leave	15BCE1352	kaahsis	warden1	kakaka	opop	0004-03-04	0344-03-31	1	0
registration	15BCE1352	kaahsis	warden1	kk	kk	0002-02-02	0003-03-31	2	0
	15BCE1352	kaahsis	warden1	kk	kk	0002-02-02	0003-03-31	3	0
	15BCE1352	kaahsish	warden1	33	33	0033-03-31	0044-04-04	4	0
	15BCE1352	utsav	warden1	33	33	0033-03-31	0044-04-04	5	-1

Given functional dependencies for table leavef:

$id, temp \rightarrow name, reason, ladd, auth$

$reason \rightarrow dol, doa$

**1NF:**

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore in 1<sup>st</sup> normal form.

**2NF:**



A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is (id,temp). Therefore non-prime attributes are name, reason,ladd,auth,reason,dol,doa. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

### **3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are name, reason,ladd,auth,reason,dol,doa but dol,doa is determined by reason. Therefore the table is not in 3NF. Now, to convert it into 3NF, we break the table leavef as follows:

leavef=leavef1(id,temp,name,reason,ladd,auth)

Its functional dependencies are:

id,temp->name,reason,ladd,auth

+

leavef2(reason,dol,doa)

Its functional dependencies are:

reason->dol,doa

Now the table is in 3NF.

### **BCNF:**

A relation is in BCNF if it is in 3NF and for each  $X \rightarrow Y$ , X is a super key. Here for table

leavef=leavef1(id,temp,name,reason,ladd,auth)

Its functional dependencies are:

---

id,temp->name,reason,ladd,auth

(id,temp) is a super key. Therefore, the table is in BCNF.

For table

leavef2(reason,dol,doa)

Its functional dependencies are:

reason->dol,doa

Here , reason is a super key. Therefore, the table is in BCNF.

---

## **COMPLAINT**

admin	name	regno	room	compt	comp
complaint	kk	15BCE1003	11	Water Problems	Water Problems
leavef	shubham	15BCE1318	320	Water Problems	Water Problems
orders	shriansh	15BCE1314	320	Water Problems	Water Problems
registration					

**Given functional dependencies for table complaint( assuming one student from a room can lodge only one complaint)**

**reg\_no->name,room**

**name->compt**

**compt->comp**

### **1NF:**

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore in 1<sup>st</sup> normal form.

### **2NF:**

A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is (reg\_no). Therefore non-prime attributes are name, room,compt,comp. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

### **3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are name, room,compt,comp but compt is determined by name and comp is determined by compt. Therefore the table is not in 3NF. Now, to convert it into 3NF, we break the table leavef as follows:

complaint=complaint1(reg\_no,name,room)

Its functional dependencies are:

reg\_no->name,room

+

complaint2(name,compt)

Its functional dependencies are:

name->compt

+

complaint3(compt.comp)

Its functional dependencies are:

compt->comp

Now the table is in 3NF.

### **BCNF:**

A relation is in BCNF if it is in 3NF and for each  $X \rightarrow Y$ ,  $X$  is a super key. Here for table complaint1(reg\_no,name,room)

Its functional dependencies are:

reg\_no->name,room

reg\_no is a super key. Therefore, this table is in BCNF.

For table

complaint2(name,compt)

Its functional dependencies are:

name->compt

name is super key. . Therefore, this table is in BCNF.

For table

complaint3(compt,comp)

Its functional dependencies are:

compt->comp

Here, compt is a super key. Therefore, the table is in BCNF.

---

## ORDERS

	name_stud	reg_no	doa	address	room_stud	flag	num
complaint	kashish	158CE1003	0999-03-31	pp	99	ACCEPTED	1
leavef	utsav	158CE1352	1993-02-04	yoy	222	ACCEPTED	2
orders	utsav	158CE1352	1993-02-04	mad	222	ACCEPTED	3
registration	utsav	158CE1352	1993-02-04	mad	222	ACCEPTED	4
	kashish	158CE1003	1997-09-04	yo	44	ACCEPTED	5
	kashish	158CE1003	0000-00-00	uuu	444	ACCEPTED	6
	kashish	158CE1003	0000-00-00	rrrr	99	ACCEPTED	7

Given functional dependencies for table orders

num->name\_stud,reg\_no,doa,address,room\_stud

reg\_no->name\_stud,room\_stud

address->doa

### 1NF:

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore in 1<sup>st</sup> normal form.

### 2NF:

A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is (num). Therefore non-prime attributes are name\_stud, reg\_no,doa,address,room\_stud. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

### **3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are name\_stud, reg\_no,doa,address,room\_stud. But name\_stud,room\_stud are determined by reg\_no. Also, doa is determined by address. Therefore the table is not in 3NF. Now, to convert it into 3NF, we break the table orders as follows:

orders=orders1(num,reg\_no,address)

Its functional dependencies are:

**num->,reg\_no,address**

+

order2(reg\_no,name\_stud,room\_stud)

Its functional dependencies are:

**reg\_no->name\_stud,room\_stud**

+

orders3(address,doa)

Its functional dependencies are:

**address->doa**

Now the table is in 3NF.

### **BCNF:**

A relation is in BCNF if it is in 3NF and for each  $X \rightarrow Y$ , X is a super key. Here for table

orders=orders1(num,reg\_no,address)

---

Its functional dependencies are:

num->,reg\_no,address

here,num is super key. Therefore the table is in BCNF.

For table

order2(reg\_no,name\_stud,room\_stud)

Its functional dependencies are:

reg\_no->name\_stud,room\_stud

reg\_no is super key. Therefore the table is in BCNF.

For table

orders3(address,doa)

Its functional dependencies are:

address->doa

Here, address is super key. Therefore the table is in BCNF.

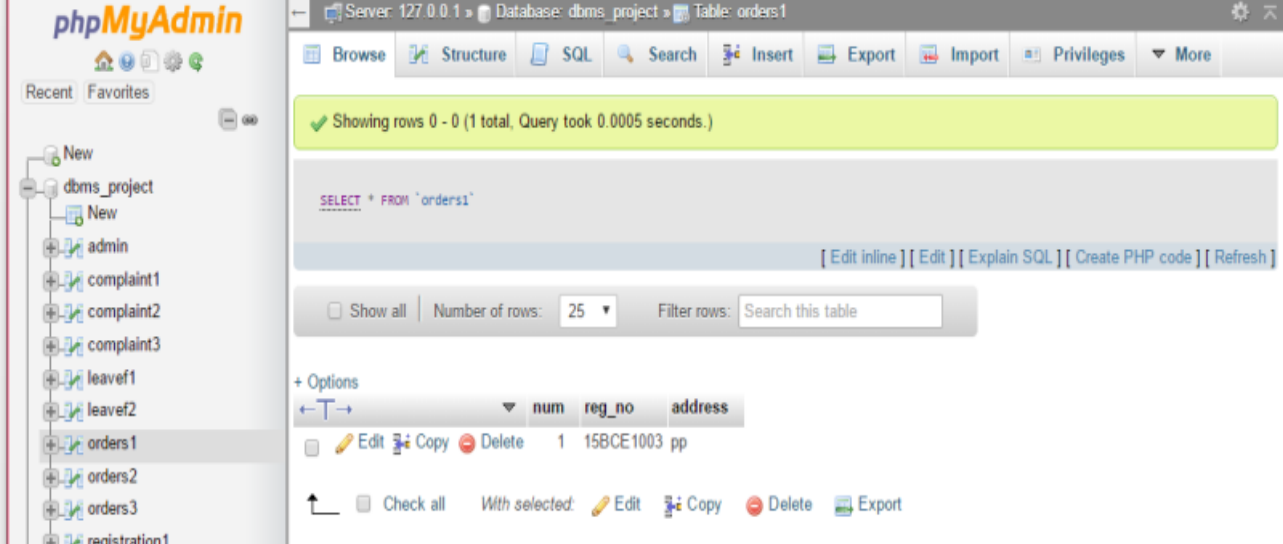
---






## ALL NORMALISED TABLES AND THEIR SCREENSHOTS:

### ORDERS 1



The screenshot shows the phpMyAdmin interface for the 'dbms\_project' database. The 'orders1' table is selected, and the 'Structure' tab is active. The table has three columns: 'num' (int(20)), 'reg\_no' (varchar(20)), and 'address' (varchar(20)). The table is currently empty, showing 0 rows. The SQL query bar contains 'SELECT \* FROM `orders1`'. The left sidebar shows the database structure, including 'dbms\_project' and its sub-databases: 'admin', 'complaint1', 'complaint2', 'complaint3', 'leave1', 'leave2', 'orders1', 'orders2', 'orders3', and 'registration1'.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	num	int(20)			No	None	AUTO INCREMENT	Change Drop Primary Unique More
2	reg_no	varchar(20)			No	None		Change Drop Primary Unique More
3	address	varchar(20)			No	None		Change Drop Primary Unique More



The screenshot shows the phpMyAdmin interface for the 'dbms\_project' database. The 'orders1' table is selected, and the 'Structure' tab is active. The table has three columns: 'num' (int(20)), 'reg\_no' (varchar(20)), and 'address' (varchar(20)). The table is currently empty, showing 0 rows. The SQL query bar contains 'SELECT \* FROM `orders1`'. The left sidebar shows the database structure, including 'dbms\_project' and its sub-databases: 'admin', 'complaint1', 'complaint2', 'complaint3', 'leave1', 'leave2', 'orders1', 'orders2', 'orders3', and 'registration1'.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	num	int(20)			No	None	AUTO INCREMENT	Change Drop Primary Unique More
2	reg_no	varchar(20)			No	None		Change Drop Primary Unique More
3	address	varchar(20)			No	None		Change Drop Primary Unique More

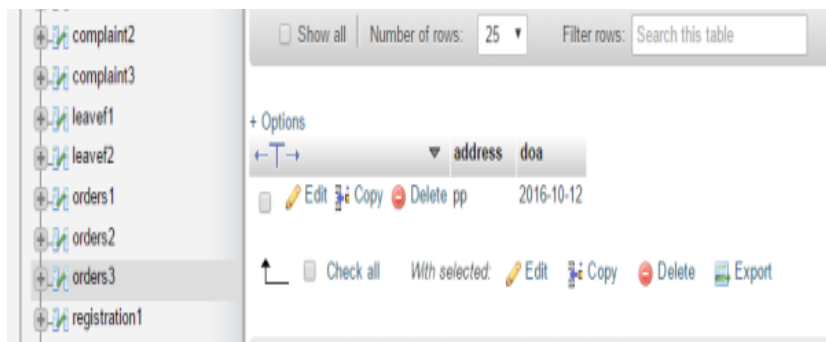
## ORDERS-2

The screenshot shows a database management interface. On the left, a tree view lists several tables: complaint2, complaint3, leavef1, leavef2, orders1, orders2 (selected), orders3, and registration1. The main area displays the 'orders2' table. At the top, there are controls for 'Show all', 'Number of rows: 25', and a 'Filter rows' search box. Below this, a '+ Options' section shows a table header with columns: reg\_no, name\_stud, and room\_stud. A single row is visible with the values: 158CE1003, kashish, and 99. At the bottom, there are action buttons: 'Check all', 'With selected', 'Edit', 'Copy', 'Delete', and 'Export'.

The screenshot shows the same database management interface, but now displaying the structure of the 'orders2' table. On the left, the tree view shows 'dbms\_project', 'New', 'admin', 'complaint1', and 'complaint2'. The main area shows a table with columns: #, Name, Type, Collation, Attributes, Null, Default, Extra, and Action. The table contains three rows of data:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	reg_no	varchar(20)			No	None		Change Drop Primary Unique Index More
2	name_stud	varchar(20)			No	None		Change Drop Primary Unique Index More
3	room_stud	int(5)			No	None		Change Drop Primary Unique Index More

## ORDERS-3

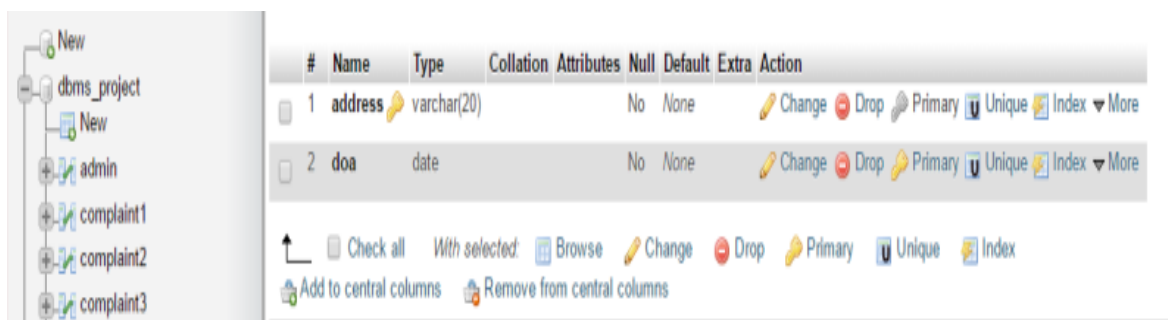


Number of rows: 25 Filter rows: Search this table

+ Options

	address	doa
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	pp	2016-10-12

Check all With selected: Edit Copy Delete Export



dbms\_project

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	address	varchar(20)			No	None		Change Drop Primary Unique Index More
2	doa	date			No	None		Change Drop Primary Unique Index More

Check all With selected: Browse Change Drop Primary Unique Index

Add to central columns Remove from central columns

## REGISTRATION-1



+ Options

	reg_no	namec	email	branch	psswr	phone	room_no	mess
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15bec1214	Aayush Gupta	aayush.gupta2015@gmail.com	ECE	ttt	0	319	Special

Check all With selected: Edit Copy Delete Export

Number of rows: 25 Filter rows: Search this table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	reg_no	varchar(20)			No	None		Change Drop Primary Unique Index More
2	namec	varchar(20)			No	None		Change Drop Primary Unique Index More
3	email	varchar(50)			No	None		Change Drop Primary Unique Index More
4	branch	varchar(20)			No	None		Change Drop Primary Unique Index More
5	psswr	varchar(20)			No	None		Change Drop Primary Unique Index More
6	phone	int(10)			No	None		Change Drop Primary Unique Index More
7	room_no	int(5)			No	None		Change Drop Primary Unique Index More
8	mess	varchar(20)			No	None		Change Drop Primary Unique Index More

## REGISTRATION 2

+ Options			
	room_no	room_type	bed_type
Edit Copy Delete	319	AC	4

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	room_no	int(5)			No	None		Change Drop Primary Unique Index More
2	room_type	varchar(20)			No	None		Change Drop Primary Unique Index More
3	bed_type	int(1)			No	None		Change Drop Primary Unique Index More

LEAVE FORM-1

leafef1

leafef2

orders1

orders2

orders3

registration1

Show all

Number of rows: 25

Filter rows: Search this table

+ Options

id	temp	name	reason	ladd	auth
15BCE1352	1	kahsis	opop	kakaka	warden1

New

dbms\_project

New

admin

complaint1

complaint2

complaint3

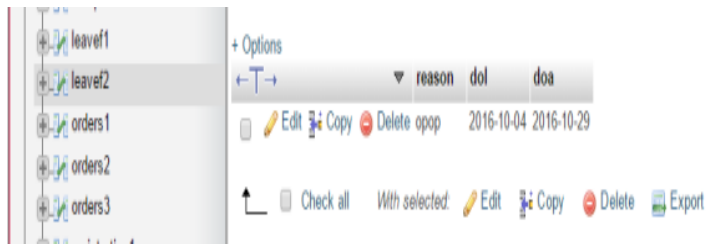
leafef1

leafef2

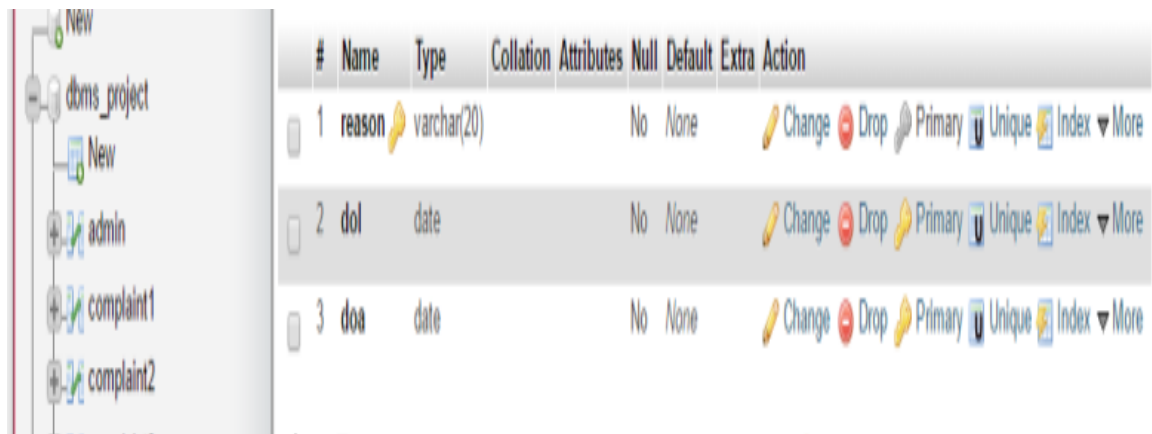
orders1

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	varchar(20)			No	None		<div>Change</div> <div>Drop</div> <div>Primary</div> <div>Unique</div> <div>Index</div> <div>More</div>
2	temp	int(20)			No	None		<div>Change</div> <div>Drop</div> <div>Primary</div> <div>Unique</div> <div>Index</div> <div>More</div>
3	name	varchar(20)			No	None		<div>Change</div> <div>Drop</div> <div>Primary</div> <div>Unique</div> <div>Index</div> <div>More</div>
4	reason	varchar(20)			No	None		<div>Change</div> <div>Drop</div> <div>Primary</div> <div>Unique</div> <div>Index</div> <div>More</div>
5	ladd	varchar(20)			No	None		<div>Change</div> <div>Drop</div> <div>Primary</div> <div>Unique</div> <div>Index</div> <div>More</div>
6	auth	varchar(20)			No	None		<div>Change</div> <div>Drop</div> <div>Primary</div> <div>Unique</div> <div>Index</div> <div>More</div>

## LEAVE FORM-2

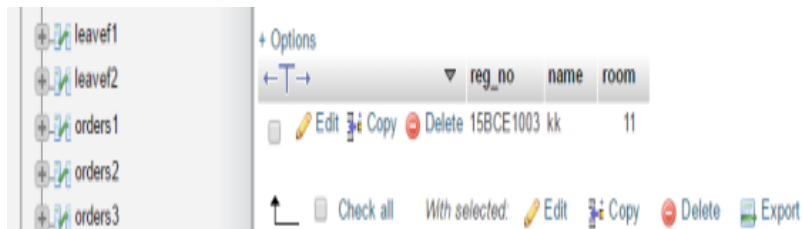


	reason	dol	doa
	opop	2016-10-04	2016-10-29



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	reason	varchar(20)			No	None		Change Drop Primary Unique Index More
2	dol	date			No	None		Change Drop Primary Unique Index More
3	doa	date			No	None		Change Drop Primary Unique Index More

## COMPLAINT-1



	reg_no	name	room
	15BCE1003	kk	11

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	reg_no	varchar(20)			No	None		Change Drop Primary Unique Index More
2	name	varchar(20)			No	None		Change Drop Primary Unique Index More
3	room	int(5)			No	None		Change Drop Primary Unique Index More

## COMPLAINT-2

name	compt
kk	Water Problems

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	name	varchar(20)			No	None		Change Drop Primary Unique Index More
2	compt	varchar(20)			No	None		Change Drop Primary Unique Index More

### COMPLAINT-3





orders1				
orders2				
orders3				
registration1				
registration2				

+ Options

id	uname	email	mobile	password
12we	warden	warden@gmail.com	9999990	opop

dbms_project				
New				
admin				
complaint1				
complaint2				
complaint3				
leavef1				
leavef2				

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	varchar(20)			No	None		Change Drop Primary Unique Index More
2	uname	varchar(20)			No	None		Change Drop Primary Unique Index More
3	email	varchar(50)			No	None		Change Drop Primary Unique Index More
4	mobile	int(10)			No	None		Change Drop Primary Unique Index More
5	password	varchar(20)			No	None		Change Drop Primary Unique Index More

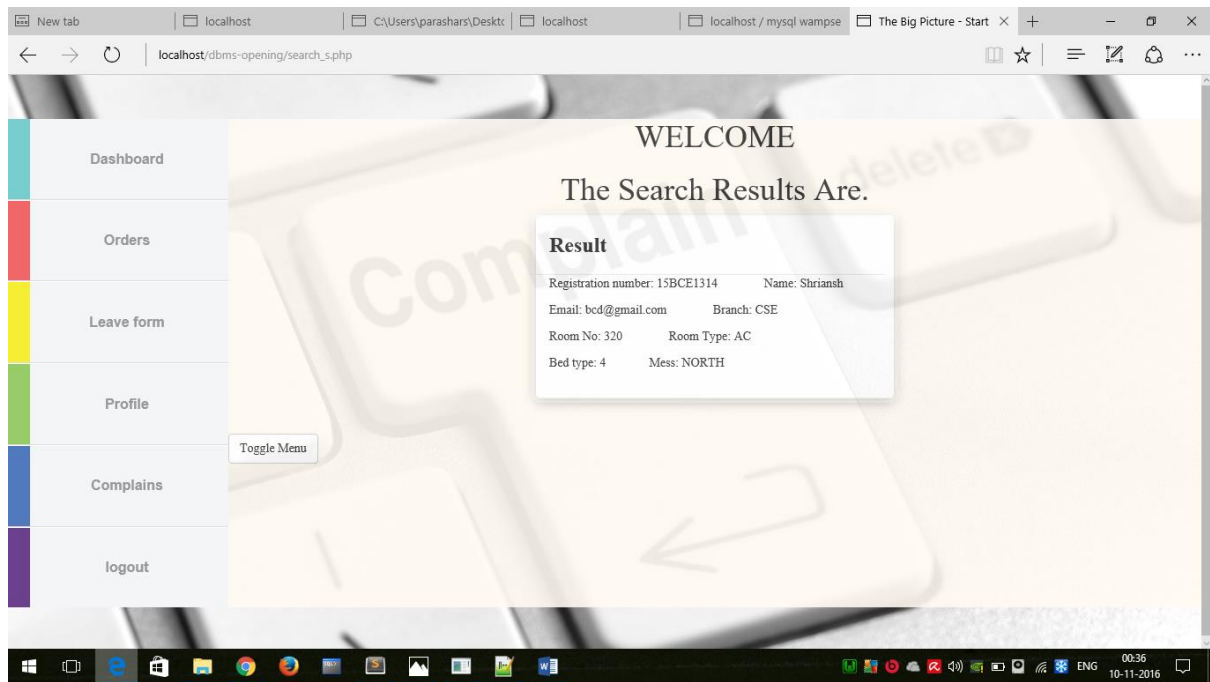
## Sample Code

The main functionality of the website is to search student and get results based on the search query.

The code given is:-

```
<?php
    if(isset($_POST['submit']))
    {
        $search=$_POST['search'];
        if(is_numeric($search[0]) && is_numeric($search[1]) && !is_numeric($search[2]))
        {
            $query1="select * from registration where reg_no='".$_.$search."'";
            $r=@mysql_query($query1) or die("no");
            while($row=@mysql_fetch_array($r)){
                echo "<div class='container1' id='signup'>
                    <div class='header'>
                        <h3>Result</h3>
                    </div>
                    <div class='sep'></div>
                    <p>Registration number: $row[reg_no] &nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Name: $row[namec]</p>
                        <p>Email: $row[email] &nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Branch: $row[branch]</p>
                        <p>Room No: $row[room_no] &nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Room Type:
$row[room_type]</p>
                        <p>Bed type: $row[bed_type]&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Mess: $row[mess]</p>
                    </div><br>";
            }
        }
    }
else if(is_numeric($search) && (strlen($search)==3 || strlen($search)==4) )
{
    $query2="select * from registration where room_no='".$_.$search."'";
    $r=@mysql_query($query2) or die("no");
    while($row=@mysql_fetch_array($r)){
        echo "<div class='container1' id='signup'>
            <div class='header'>
                <h3>Result</h3>
            </div>
            <div class='sep'></div>
            <p>Registration number: $row[reg_no] &nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Name: $row[namec]</p>
                <p>Email: $row[email] &nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Branch: $row[branch]</p>
```

[illegible]



## **BIBLIOGRAPHY**

[www.google.com](http://www.google.com)

[www.wikipedia.org](http://www.wikipedia.org).

[www.w3schools.com](http://www.w3schools.com)

DBMS book