PROJECT REPORT
On
STUDENT COMPLAINT DATABASE MANAGEMENT SYSTEM

PREPARED BY: KENGUVA ANANTH KUMAR (15CO124)

REDDI NARASIMHA PRASAD (15CO133)

COURSE: DATABASE MANAGEMENT SYSTEM

TABLE OF CONTENTS:

SYNPOPSIS:

Abstract:

Student complaint database management system can be used by education institutes to know the complaints of the students easily. Achieving this objective is difficult using a manual system
As the information is scattered, can be redundant and collecting relevant information may be very time consuming .All these problems are solved using this project.

NAME OF THE PROJECT: Student complaint database management system.

Objectives:

Online registration of complaints
Maintenance of student's records and complaints
Searching student complaints

User's Views:

Administrator
Student

Platform:

Microsoft Windows.

Technologies Used:

Front end: HTML, JAVASCRIPT, CSS, BOOTSTRAP.
Backend: PHP, MYSQL.

Software requirements:

PHP 5.0
Mysql server
Xampp server
Laravel framework
Bootstrap programming for front end design.

Hardware requirements:

Intel Pentium IV processor or equivalent or higher.
512 MB Ram or higher.
20 GB HDD or higher.
Network Connectivity

SOFTWARE REQIUREMENT SPECIFICATION

1. INTRODUCTION:

1.1) GOAL:

Our goal is to deliver a database with a user interface (website) so that students feel hassle free as it avoids the need of going to various offices present in the college to register a complaint as well as to check their status. Management can resolve various issues of students and also gets feedback from students once the complaint has been resolved through online service. The focus is to create an "easy to use" website, which will allow students to give complaints with ease.

1.2) SCOPE:

Without a **Student complaint database management System**, managing and maintaining the details of the student is a tedious job for any organization.
Student Information system will store all the details of the students including their personal details and all the information related to their college complaints.

Login module: Login module will help in authentication of user accounts. Users who have valid login id and password can only login into their respective accounts.

Search module: Suppose there are hundreds of students and from this we have to search a particular student complaints and we know the name of the student .In manual system it is a tedious task though we know the name of the student, but using this module we can easily search the student complaints by specifying the name of the student in the search criteria. Thus this module will help the administrator in searching the student with various criteria easily.

Complaint Registration Module: This module will help the student register complaints from anywhere if internet is present. This module will really simplify the task of on paper registration of complaints. Also after successful Registering the complaints the user can check the status of their complaint.

1.3) Definitions:

Student details: Details of student such as user id, phone number, address, image, department, e-mail address etc.

Administrator: A Login Id representing the user is an administrator & can access all the complaints registered by the students.

1.4) OVERVIEW:

The current system where student has to fill the form by going to the office or by calling the particular person to clear his complaint which consumes the time of the student and more of paper work. To avoid this we can have a better complaint database for complaints. Student Complaint database management system provides online services for the complaints of student in both academics and hostel. Students can easily post their complaint by sitting in their rooms and can check the status of their Complaints at any time. All the complaints arrived from the students are separated according their type of complaint (academics (or) hostel). Administrator goes through all those complaints and try to solve them according to their complaint type. It saves a lot of time compared to the manual work.

2) OVERALL DESCRIPTION:

2.1) Product Perspective:

Student complaint database Management System deals with all type of student complaints, academic related complaints and Hostel related complaints too. It tracks all the complaints of a student and store them in the database which helps the admin to know the complaints and solve them according to type of the complaint.

Our design can facilitate us to explore all the complaints facing by the students in the college, even we can get to know on which faculty the student is facing problem on. The current status of a student complaint tells us the progress of his/her complaint. The student complaint database Management system is an automated version of manual Student complaint Management System. It can handle all the complaints of a student.

In case of manual system they need a lot of time, manpower etc. Here almost every work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do with in a few minutes. Our system has two type of accessing modes, administrator and user. Student complaint database management system is managed by an administrator. It is the job of the administrator to insert update and monitor the whole process. When a user log in to the system, He/she can only view his/her complaint details or records of the student. He/she can't perform any changes other than updating his profile details. Our system has seven modules, they are administrator, student, course, department, faculty and complaint section. These modules and its attributes with entity relationship module presented in the ER diagram section.

2.2) Product Functions:

There are two different users who will be using this product:
Administrator who can view and update the status of the complaints of any students.
Students who can view their details as well as they can register the complaints.

The features that are available to the Administrator are:

An Administrator can login into the system and perform any of the available operations.
Can enable/disable student.
Can make search for a specific student.
Can access all the complaints of the student.

The features that are available to the student are:

Student can login into the system and can perform any of the available options.
Can view his/her personal details.
Can edit his/her personal details
Can upload his/her image.
Can register the complaints.
Can submit the feedbacks.

2.3) Users:

Two types of users should be able to use the system: students and administrator. Admin is the user works with the system on the daily basis. Admin will have his or her own account to log on to. He or she is the one responsible for processing complaints. Admin provides login, user name and passwords to students in college to log in into website. Students are the users who visit the website and can register the complaints by selecting the type of complaint with a small description of it, in the text box. Administrator or super user has the ultimate control of the system, he or she can accept, process and change the status of the complaints or delete or create a login accounts for the students.

3) REQUIREMENT ANALYSIS:

The system is made up of three layers. At the top there is the GUI (Graphical User Interface) layer, the middle layer is the storage and query manager, the bottom layer is the underlying database.

GUI layer:

The GUI layer allows users to access the system. All the functionalities of the system must be available through the GUI. There are two separate GUI’s. One is for the students to register complaints and one is for admin for processing complaints and administration purposes. The GUI should prevent input errors and in case of errors that could not be prevented, provide clear error messages. Students need a username and a password to gain access to the system. The administrator has access to add users to the system to give them access.

Storage and query manager layer:

The storage and query manager layer is responsible for information storage, retrieval, authorizations and error checking. This layer allows selecting, adding, updating, deleting entities and relations in the database by using different queries. Some instructions are limited to users with authorization, such as deleting data from the database which should only be allowed by the administrator.

Database layer:
The database layer contains all the data of entities and their relationships.

3.1) REQUIREMENTS:

In the following paragraph, the functional requirements for each type of users are listed. Non-functional requirements of the system are also mentioned here.

3.1.1) FUNCTIONAL REQUIREMENTS:

STUDENTS:

1. Users must be able to log in through their account.
2. Users must be able to edit profile and update their accounts.
3. Users must be able to change the passwords of their accounts.
4. Users must be able to create a new complaint.
5. Users must be able to customize a complaint by:
    5.1) Users must be able to view a list of category of complaints.
    5.2) Users must be able to modify or update the registered complaint.
    5.3) Users must be able to delete the registered complaint.
    5.4) Users must be able to get a response from the admin about the expected date, time of resolving the issue.
    5.5) Users must be able to check the status of their complaints.

6. The user must be able to view the list of complaints registered by him or her.
7. Users must receive an email or a sms message after registering a complaint and after resolving complaint.
8. Users must be able to log out from their accounts.

ADMINISTRATOR:

1. Administrator must be able to log in and out of the system.
2. Administrator must be able to view the list of complaints received from the students.
3. Administrator must be able to mark the status of complaints as “received” or “processing” or resolved”.
4. Administrator must be able to add /delete/edit other users.
5. Administrator must use an appropriate mechanism to send automatic responses to students about their complaints.

6. Only users with respective rights (administrators) must be able to use all these “Administrators” features.

3.1.2) NON-FUNCTIONAL REQUIREMENTS:

As an operational requirement, the system will run as a database with a website as user interface. As performance requirement the system must be accessible 24 hours a day, seven days a week. Due to the nature of the system as a complaints website, the system must have a low response time, preferably shorter than second, with a maximum of five seconds. The exception is viewing order logs which could have a higher response time (of seconds) as the log increases in size over time. Due to the low complexity of the system, no problems with response time are expected.

Students who visit the website to register a complaint will get a session ID for their visit, which is used to identify them while using the system. For every action they take, a timestamp is stored. From time to time a service on the server will scan session ID’s and timestamps. Session ID’s which have not been active for more than three hours will be deleted along with the corresponding information.

4) APPROACH:

During the process of selecting the tools and methods to be used during the process, we need to a look at our current expertise and the requirements of the project. The database is going to be a mysql driven database, php showed to be the right choice. PHP has an excellent integration with mysql. Because php is an open source system, there are a huge amount of hosts available. For managing the mysql database, we are going to use laravel framework. This is the logical choice because it is free, and it integrates so well. For creating the code, we are going to use sublime text editor. This is a multipurpose xhtml editor, supporting a lot of different syntax languages. We used this program because we already had experience using it. The system will be divided into two main parts. The smaller administration area, and the larger student area. Both parts have their own folders,for example: localhost/student is the folder containing the student, and localhost/student/ Admin/ is the folder containing admin page.

5) DESIGN PHASE:

1 INTRODUCTION:

Scope and purpose:

The purpose of the design phase is to develop a clear understanding of what the developer want people to gain from his/her project. As you the developer work on the project, the test for every design decision should be "Does this feature fulfil the ultimate purpose of the project?"

A purpose statement affects the design process by explaining what the developer wants the project to do, rather than describing the project itself.

The Design Document will verify that the current design meets all of the explicit requirements contained in the system model as well as the implicit requirements desired by the customer.

Overall system design objectives:

The overall system design objective is to provide an efficient, modular design that will reduce the system's complexity, facilitate change and result in an easy implementation. This will be accomplished by designing strongly cohesion system with minimal coupling. In addition, this document will provide interface design models that are consistent user friendly and will provide straight forward transition through the various system functions.

Structure of design document:

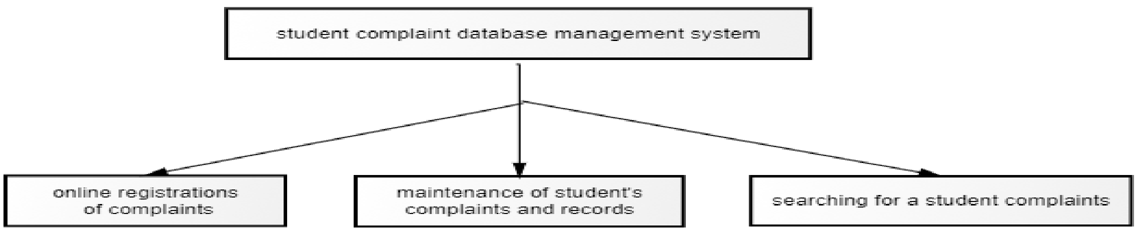
System architecture design – the system architecture section has detailed diagram of the system.

Data design – the data design include an ERD as well as Relational schema.

2) SYSTEM ARCHITECTURE DESIGN:

The student complaint database management system is a system which contain major part which include: student, student complaints.

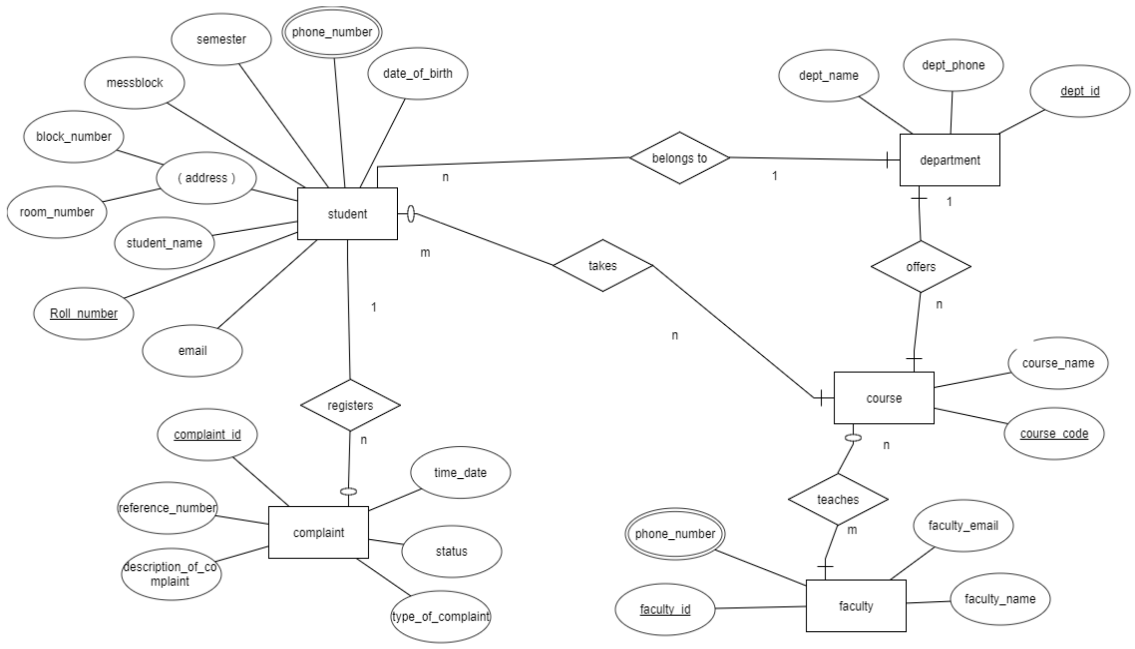
The user selects one of the available options as an input to the system. According to the input by the user the system acts and the rest of the functions are performed accordingly. The administrator can operate on any student details. But the normal student or users can only access their details of all the functionalities.

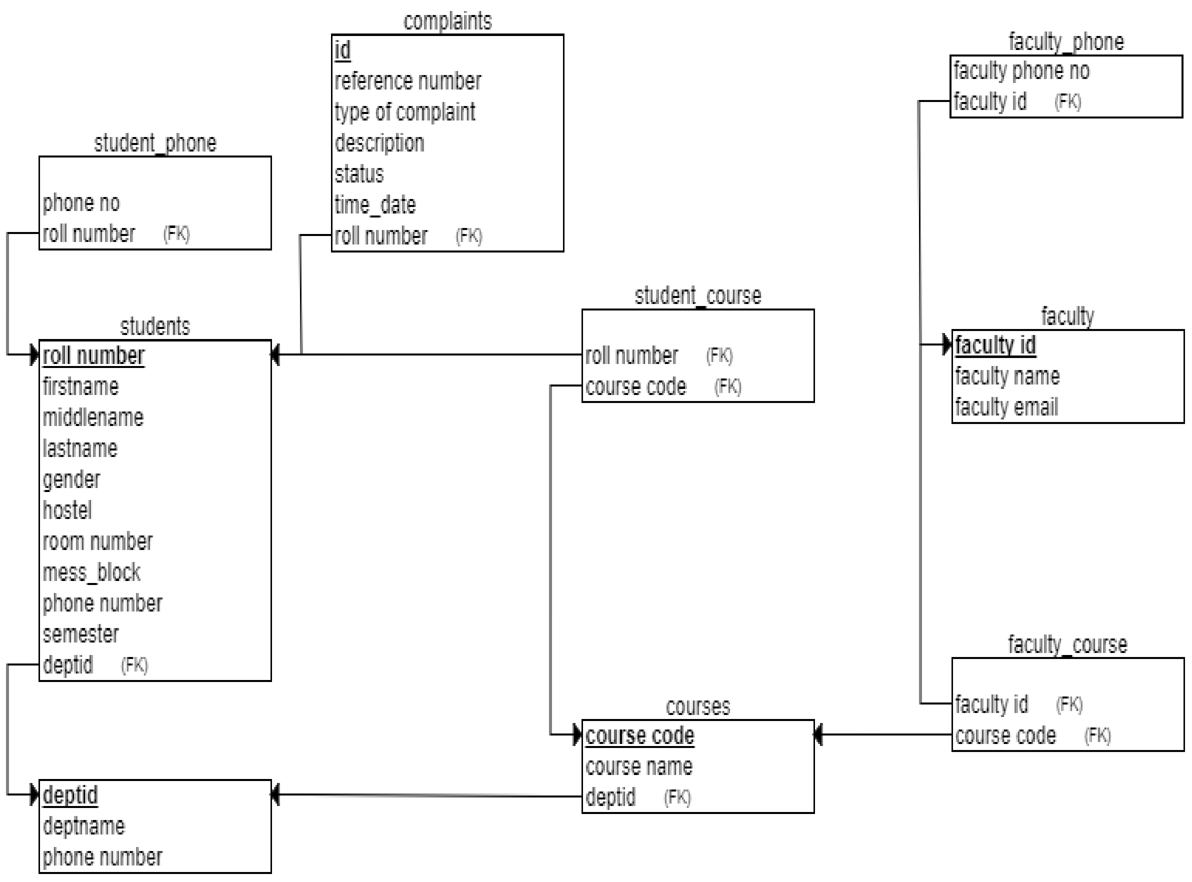


student complaint database management system
online registrations of complaints
maintenance of student's complaints and records
searching for a student complaints

ARCHITECTURE DESIGN

3) DATA DESIGN: -- ER DIAGRAM





RELATION SCHEMA

6) IMPLEMENTATION:

The technology selected for implementing student complaint database management system is PHP/MYSQL. XAMPP is used as the HTTP server. The development was done in a 'windows' environment laravel framework.

PHP:

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. PHP code is embedded into the HTML source document. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content. It can also be used for command-line scripting and client-side

GUI applications. PHP can be deployed on many web servers and operating systems, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

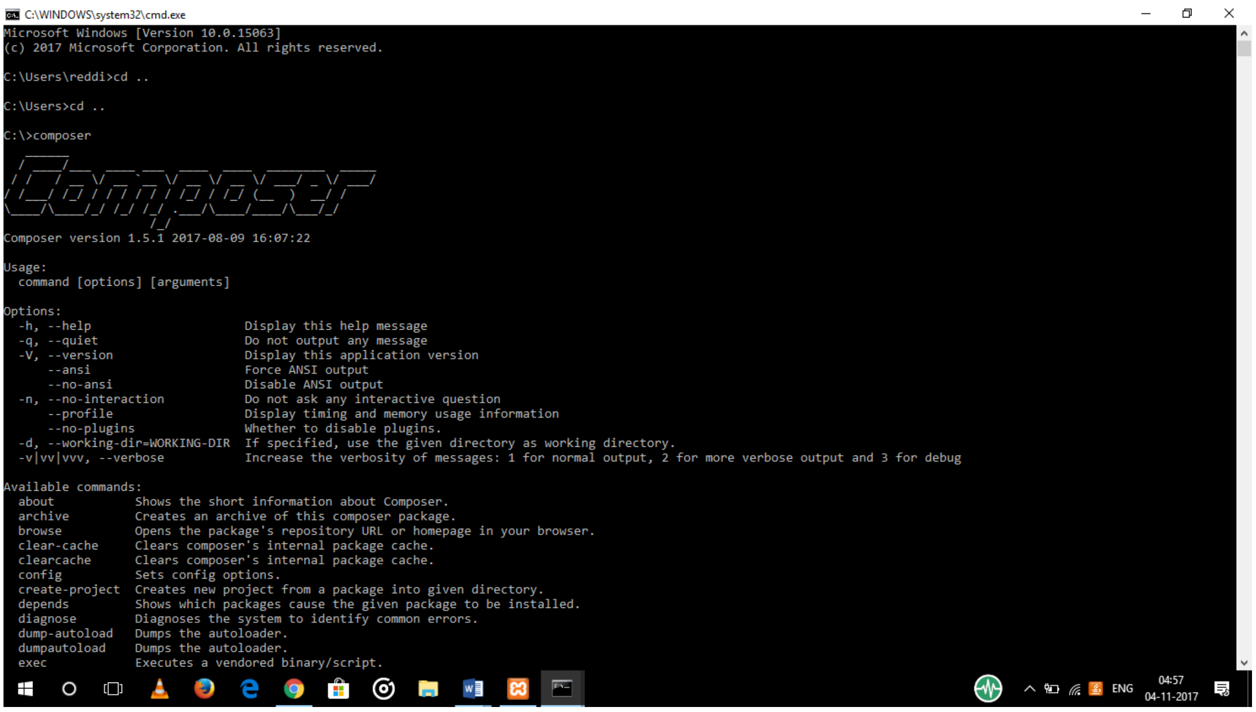
MYSQL:

MySQL is a relational database management system (RDBMS) that runs as a server providing multi user access to a number of databases. MySQL is a popular choice of database for use in web applications and is an open source product. The process of setting up a MySQL database varies from host to host, however we will end up with a database name, a user name and a password. Before using our database, we must create a table. A table is a section of the database for storing related information. In a table we will set up the different fields which will be used in that table. Creating a table in 'HEIDI SQL' is simple, we just type the name, select the number of fields and click the 'go' button. we will then be taken to a setup screen where you must create the fields for the database. Another way of creating databases and tables in HEIDI SQL is by executing simple SQL statements. We have used this method in order to create our database and tables.

LARAVEL:

The Laravel local host is a web server that we have used in this project. Laravel is a web application framework with expressive, elegant syntax. We believe development must be an enjoyable, creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, sessions, and caching. Laravel aims to make the development process a pleasing one for the developer without sacrificing application functionality. Happy developers make the best code. To this end, we've attempted to combine the very best of what we have seen in other web frameworks, including frameworks implemented in other languages, such as Ruby on Rails, ASP.NET MVC, and Sinatra. Laravel is accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly integrated unit testing support give you the tools you need to build any application with which you are tasked.

It is installed using **“composer”.** So, first download composer from https://getcomposer.org/ and install it. After that go to cmd and then type Composer, it should display as below



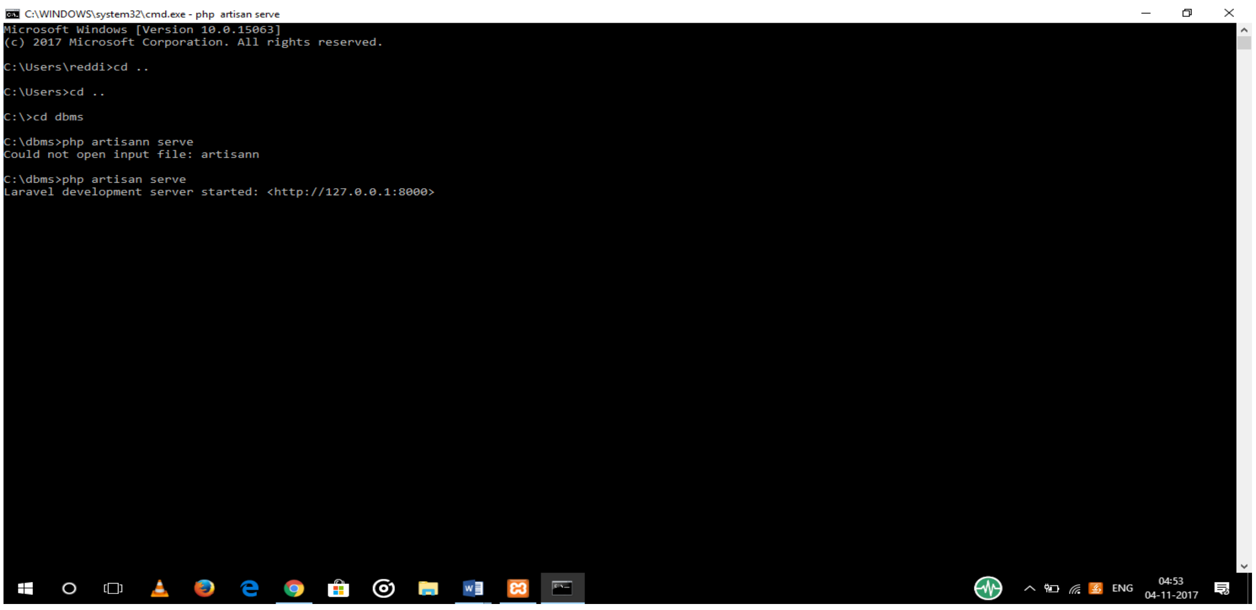
```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\reddi>cd ..

C:\Users>cd ..

C:\>composer
   ______
  / ____/___  ____ ___  ____  ____  ________  _____
 / /   / __ \/ __ `__ \/ __ \/ __ \/ ___/ _ \/ ___/
/ /___/ /_/ / / / / / / /_/ / /_/ (__  )  __/ /
\____/\____/_/ /_/ /_/ .___/\____/____/\___/_/
                    /_/
Composer version 1.5.1 2017-08-09 16:07:22

Usage:
  command [options] [arguments]

Options:
  -h, --help                     Display this help message
  -q, --quiet                    Do not output any message
  -V, --version                  Display this application version
      --ansi                     Force ANSI output
      --no-ansi                  Disable ANSI output
  -n, --no-interaction           Do not ask any interactive question
      --profile                  Display timing and memory usage information
      --no-plugins               Whether to disable plugins.
  -d, --working-dir=WORKING-DIR  If specified, use the given directory as working directory.
  -v|vv|vvv, --verbose           Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about           Shows the short information about Composer.
  archive         Creates an archive of this composer package.
  browse          Opens the package's repository URL or homepage in your browser.
  clear-cache     Clears composer's internal package cache.
  clearcache      Clears composer's internal package cache.
  config          Sets config options.
  create-project  Creates new project from a package into given directory.
  depends         Shows which packages cause the given package to be installed.
  diagnose        Diagnoses the system to identify common errors.
  dump-autoload   Dumps the autoloader.
  dumpautoload    Dumps the autoloader.
  exec            Executes a vendored binary/script.
```

It makes sure that laravel is installed in computer. Now install laravel using composer by typing

```
composer global require "laravel/installer=~1.1"
```

in command prompt.

In order to test that everything has been installed correctly, first start the laravel Server by going to command prompt(WINDOWS+R) and go to directory where the project folder is located and type **“php artisan serve”** If configured correctly, we will be presented with a screen similar to that of the one below.



```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\reddi>cd ..

C:\Users>cd ..

C:\>cd dbms

C:\dbms>php artisann serve
Could not open input file: artisann

C:\dbms>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

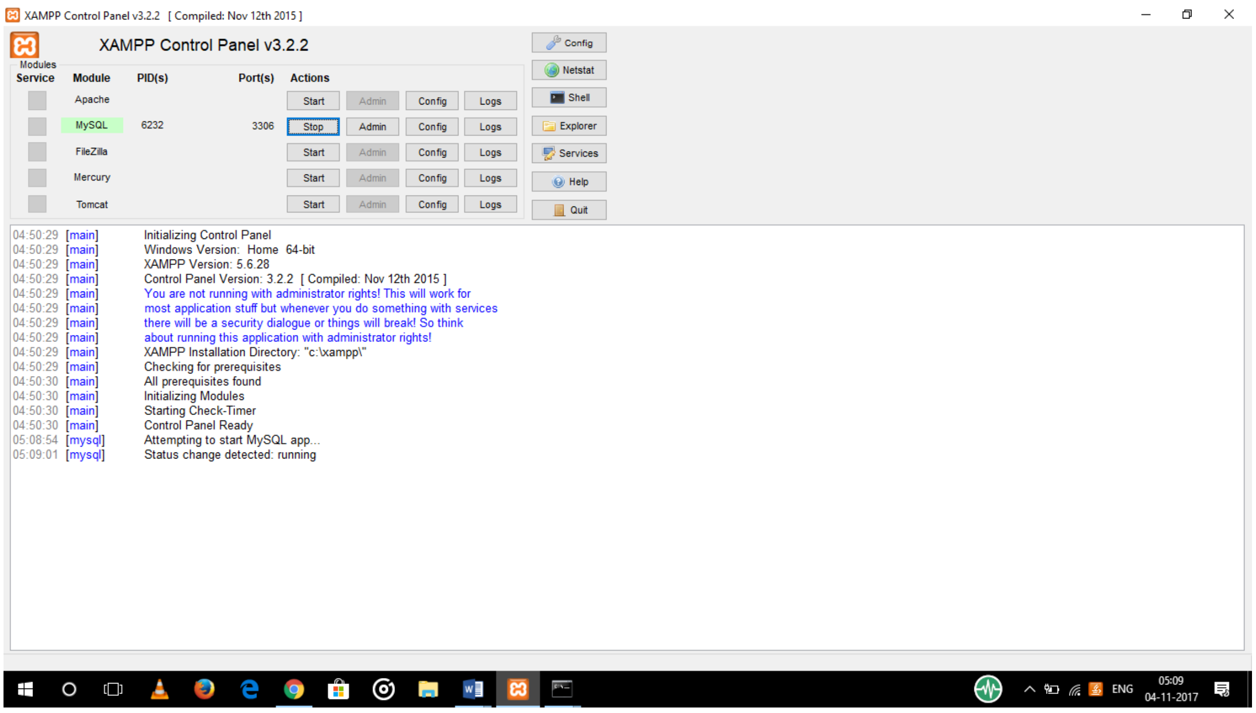
XAMPP:

XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL.

XAMPP is available as a free download in two specific packages: full and lite. While the full package download provides a wide array of development tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standards. The light version is a small package containing Apache HTTP Server, PHP, MySQL, phpMyAdmin, Openssl, and SQLite.

Obtaining and Installing XAMPP:

As previously mentioned, XAMPP is a free package available for download and use for various web development tasks. All XAMPP packages and add-ons are distributed through the Apache Friends website at the address: http://www.apachefriends.org/. Once on the website, navigate and find the Windows version of XAMPP and download the self-extracting ZIP archive. After downloading the archive, run and extract its contents into the root path of a hard disk or USB drive. For example, the extract path for a local Windows installation would simply be C:\. If extracted properly we will notice a new xampp directory in the root of your installation disk. Now open xampp control panel and start mysql it will look as


XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]

XAMPP Control Panel v3.2.2

Service	Module	PID(s)	Port(s)	Actions			
	Apache			Start	Admin	Config	Logs
	MySQL	6232	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
04:50:29  [main]    Initializing Control Panel
04:50:29  [main]    Windows Version:  Home  64-bit
04:50:29  [main]    XAMPP Version: 5.6.28
04:50:29  [main]    Control Panel Version: 3.2.2  [ Compiled: Nov 12th 2015 ]
04:50:29  [main]    You are not running with administrator rights! This will work for
04:50:29  [main]    most application stuff but whenever you do something with services
04:50:29  [main]    there will be a security dialogue or things will break! So think
04:50:29  [main]    about running this application with administrator rights!
04:50:29  [main]    XAMPP Installation Directory: "c:\xampp\"
04:50:29  [main]    Checking for prerequisites
04:50:30  [main]    All prerequisites found
04:50:30  [main]    Initializing Modules
04:50:30  [main]    Starting Check-Timer
04:50:30  [main]    Control Panel Ready
05:08:54  [mysql]   Attempting to start MySQL app...
05:09:01  [mysql]   Status change detected: running
```

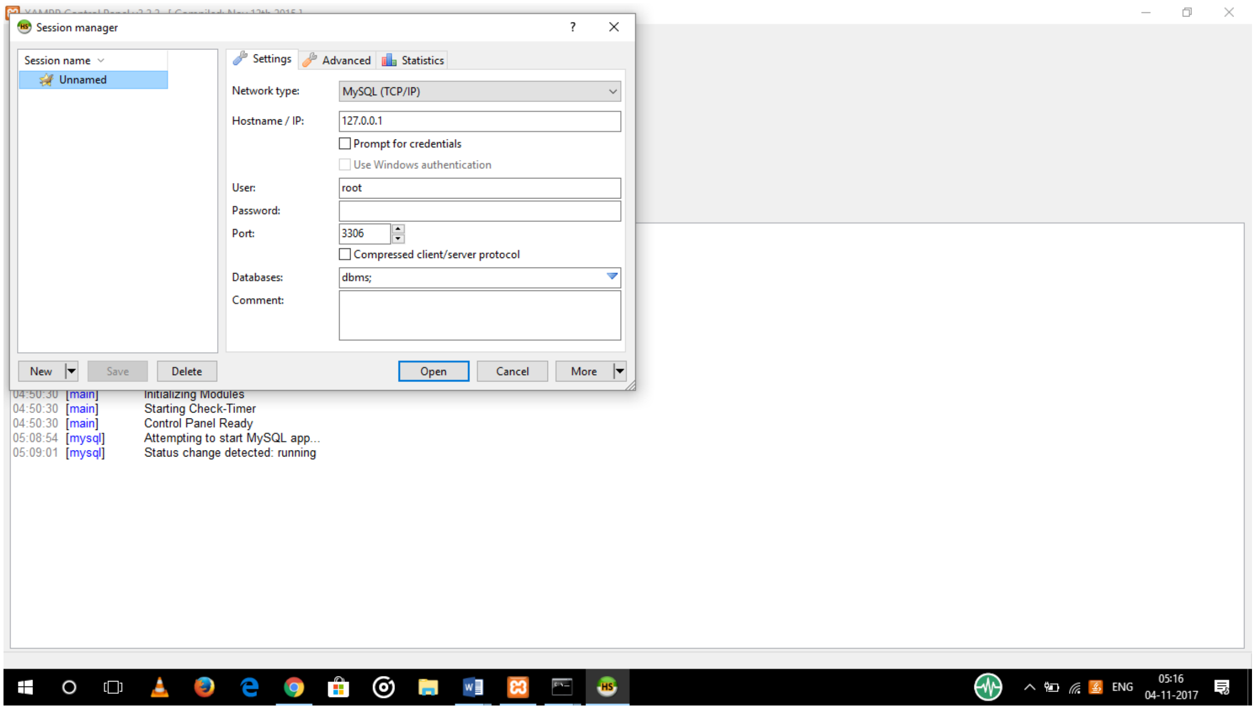
Now open shell present on the right side. I will open a window of mysql and connect mysql to laravel server by typing "mysql –u root –h 127.0.0.1".

Databases and table which will hold information to be used by our website. We will be using HeidiSQL to create a database and table, and enter test data.

HEIDI SQL:

Heidi SQL is a useful and reliable tool designed for web developers using the popular MySQLserver, Microsoft SQL databases and PostgreSQL. It enables you to browse and edit data, create and edit tables, views, procedures, triggers and scheduled events. This software can be downloaded from https://www.heidisql.com/ and install it.

Once XAMPP and laravel is launched, it will automatically detect server and database we are using it will look like as

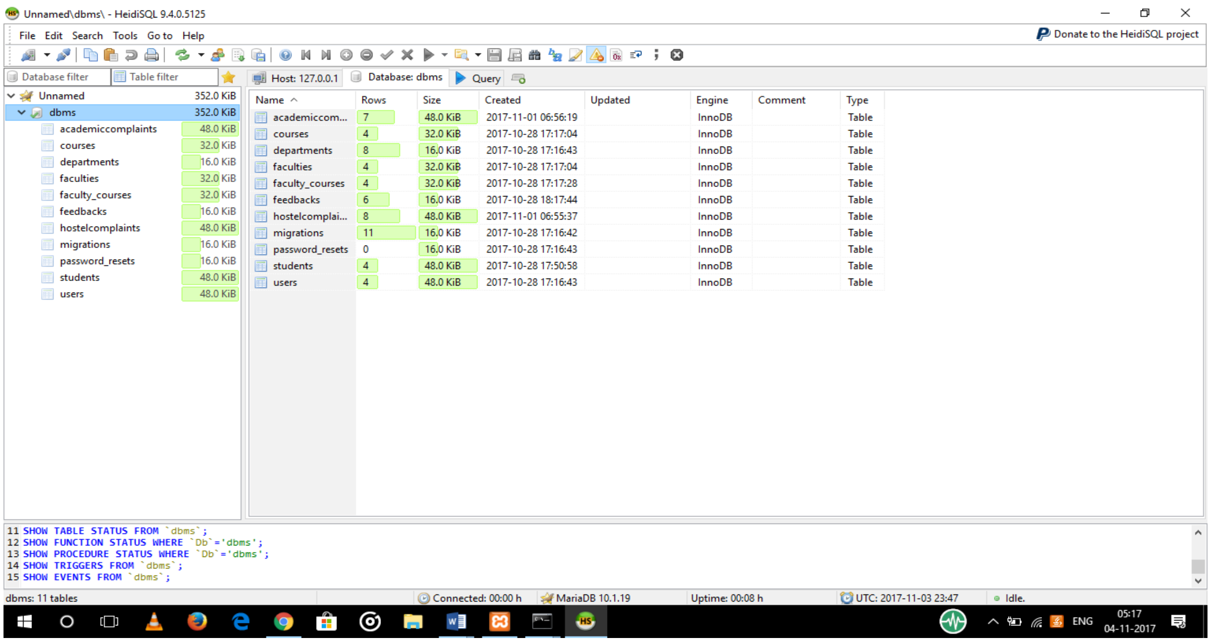


Session manager

Network type:	MySQL (TCP/IP)
Hostname / IP:	127.0.0.1
	☐ Prompt for credentials
	☐ Use Windows authentication
User:	root
Password:	
Port:	3306
	☐ Compressed client/server protocol
Databases:	dbms;
Comment:	

```
04:50:30  [main]     Initializing Modules
04:50:30  [main]     Starting Check-Timer
04:50:30  [main]     Control Panel Ready
05:08:54  [mysql]    Attempting to start MySQL app...
05:09:01  [mysql]    Status change detected: running
```

Click open and it will display the database and tables we are using, if it is empty create databases and tables that are needed for the website in this software. Entire picture can be shown as



Unnamed\dbms\ - HeidiSQL 9.4.0.5125

Host: 127.0.0.1 | Database: dbms | Query

Name	Rows	Size	Created	Updated	Engine	Comment	Type
academiccom...	7	48.0 KiB	2017-11-01 06:56:19		InnoDB		Table
courses	4	32.0 KiB	2017-10-28 17:17:04		InnoDB		Table
departments	8	16.0 KiB	2017-10-28 17:16:43		InnoDB		Table
faculties	4	32.0 KiB	2017-10-28 17:17:04		InnoDB		Table
faculty_courses	4	32.0 KiB	2017-10-28 17:17:28		InnoDB		Table
feedbacks	6	16.0 KiB	2017-10-28 18:17:44		InnoDB		Table
hostelcomplai...	8	48.0 KiB	2017-11-01 06:55:37		InnoDB		Table
migrations	11	16.0 KiB	2017-10-28 17:16:42		InnoDB		Table
password_resets	0	16.0 KiB	2017-10-28 17:16:43		InnoDB		Table
students	4	48.0 KiB	2017-10-28 17:50:58		InnoDB		Table
users	4	48.0 KiB	2017-10-28 17:16:43		InnoDB		Table

```
SHOW TABLE STATUS FROM `dbms`;
SHOW FUNCTION STATUS WHERE `Db`='dbms';
SHOW PROCEDURE STATUS WHERE `Db`='dbms';
SHOW TRIGGERS FROM `dbms`;
SHOW EVENTS FROM `dbms`;
```

dbms: 11 tables | Connected: 00:00 h | MariaDB 10.1.19 | Uptime: 00:08 h | UTC: 2017-11-03 23:47 | Idle.

Insert values into table by clicking on particular table and navigating to data icon present within it.

7) SCREENSHOTS AND DESCRIPTION:

6.1.1) IMPLEMENTATION ON USERS SIDE:-

1 LOGIN PAGE:

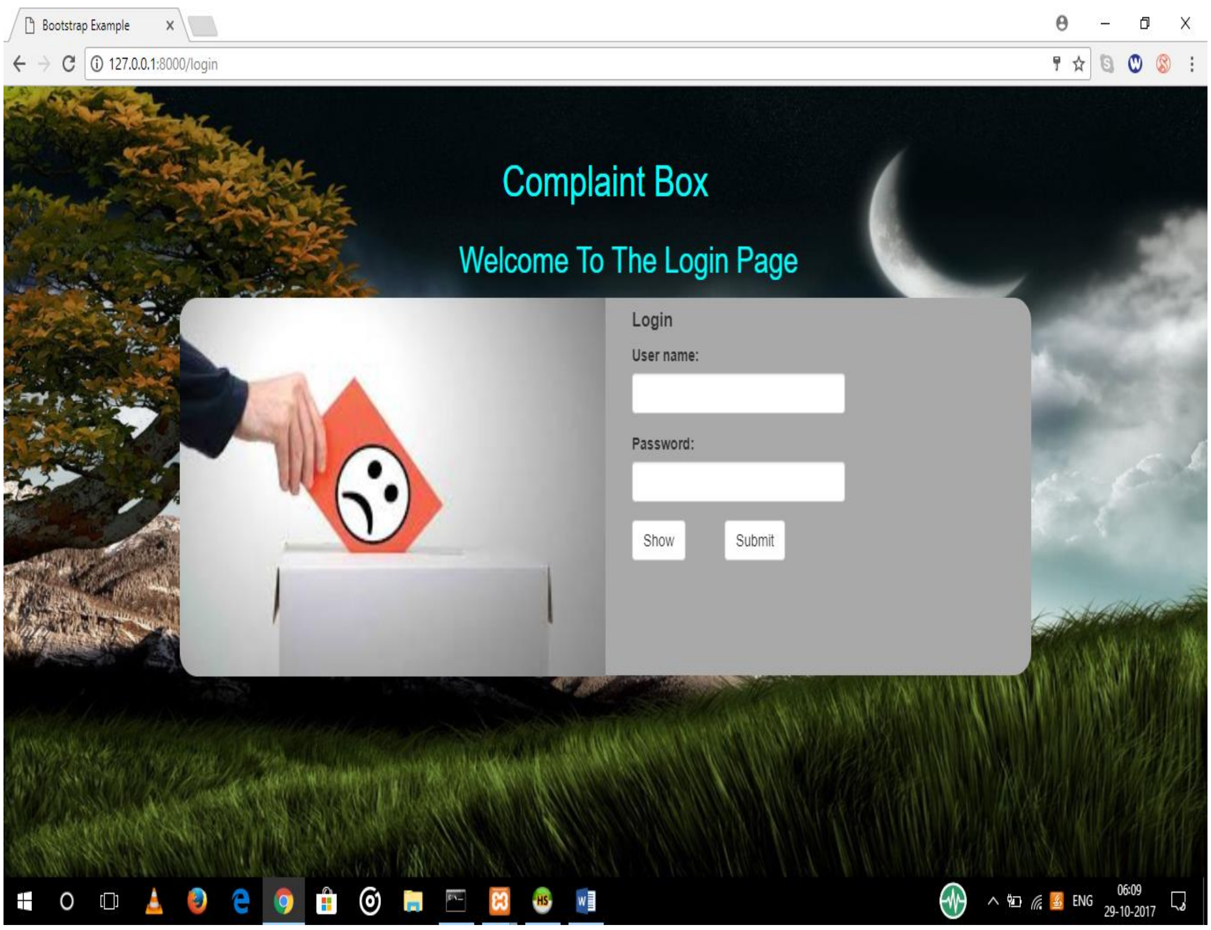
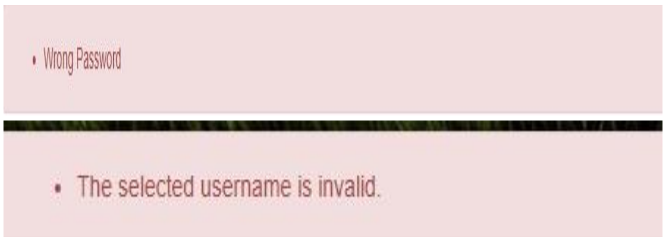


Fig2.2 login page

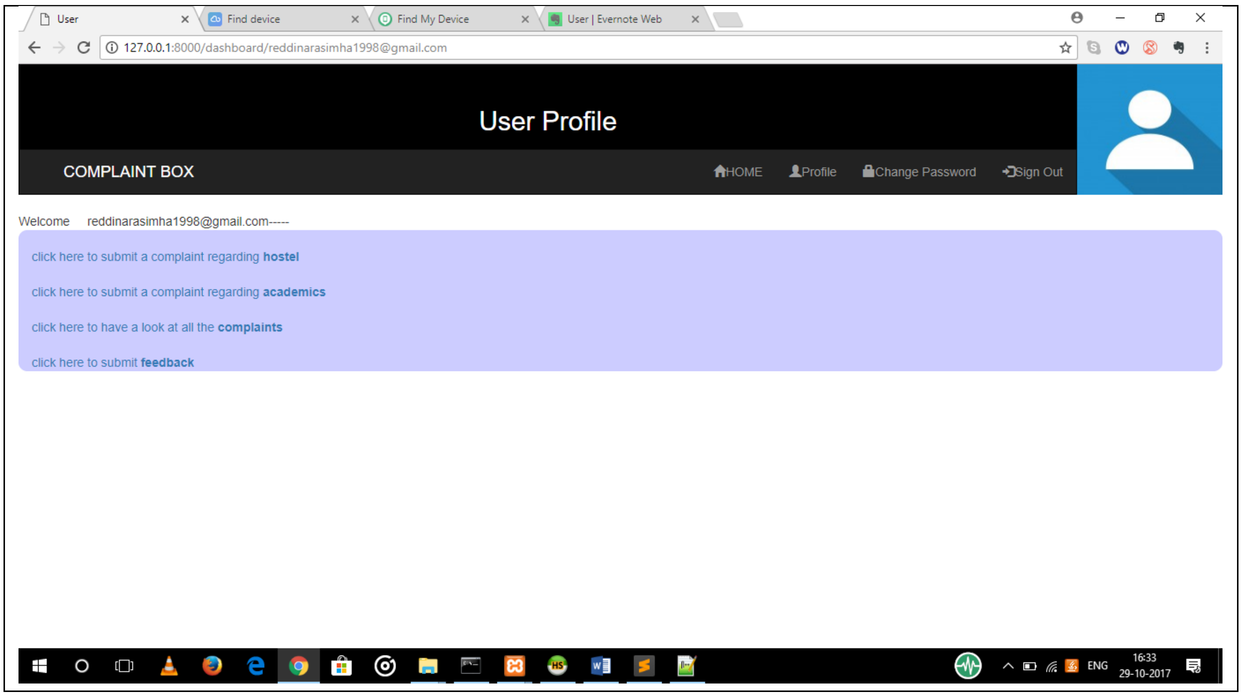
DESCRIPTION:

After clicking the login button, It redirects to login page as follows, User are provided with the text input buttons for entering the username and password .Users can view the password by clicking show button as per his wish. Once user clicks on submit button, dashboard page of user will be shown on the page. If User enters a wrong password or incorrect username, errors will be shown as below.


- Wrong Password
- The selected username is invalid.

This login page is same for both admin and students. Based on whether user is admin or student, corresponding dashboard page will be displayed. This functionality is achieved by having one additional column is_admin. We set is_ admin=1 for admin and 0 for student.

2 DASHBOARD PAGE:


127.0.0.1:8000/dashboard/reddinarasimha1998@gmail.com

User Profile

COMPLAINT BOX

HOME Profile Change Password Sign Out

Welcome reddinarasimha1998@gmail.com-----

click here to submit a complaint regarding **hostel**

click here to submit a complaint regarding **academics**

click here to have a look at all the **complaints**

click here to submit **feedback**

Fig2.3 dashboard page for user

DESCRIPTION:

In dashboard page, Student will see his profile photo on top right corner of the page, he can update the photo by clicking on avatar icon. In header section, on navigation bar he can see four icons namely home, profile, change password and sign out. On clicking the particular icon, user is redirected to the corresponding page.

In the body section, there will be 4 links

(i) For submitting a hostel complaint
(ii) For submitting an academic complaint
(iii) For viewing all submitted complaints
(iv) For submitting a feedbacks.

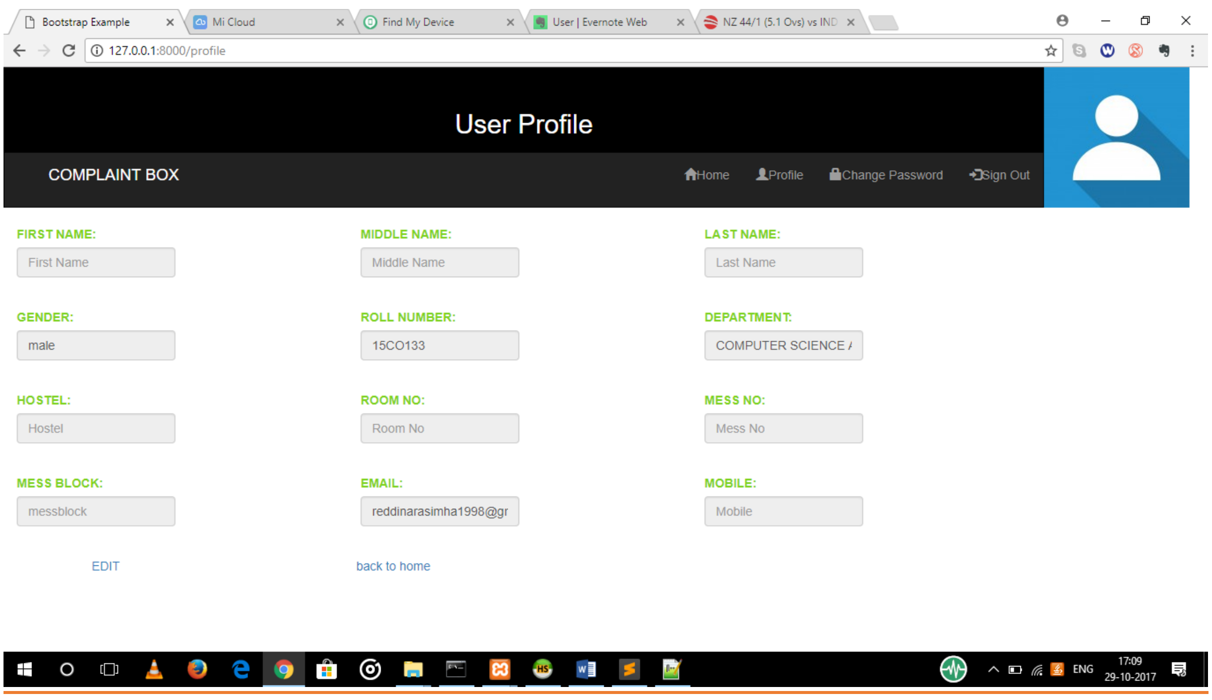
On clicking the particular link, User views the page to perform the desired action in the page.

3) PROFILE PAGE:

DESCRIPTION:

This page is displayed when user clicks on profile icon on dashboard page. This page contains the details of student such as his first name, middle name, last name, phone number, roll number, hostel details, department, gender, email etc.

On the bottom an **EDIT** option is given to change his profile details.

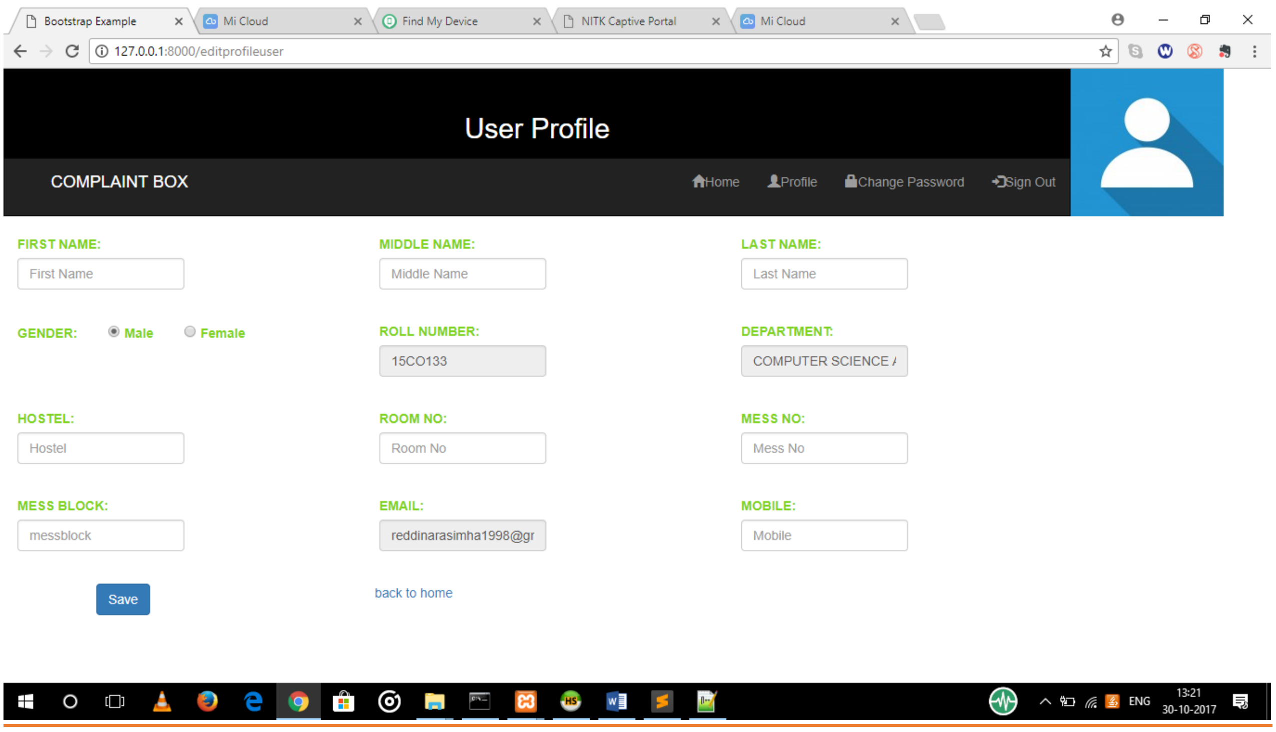


User Profile

FIRST NAME: First Name	MIDDLE NAME: Middle Name	LAST NAME: Last Name
GENDER: male	ROLL NUMBER: 15CO133	DEPARTMENT: COMPUTER SCIENCE /
HOSTEL: Hostel	ROOM NO: Room No	MESS NO: Mess No
MESS BLOCK: messblock	EMAIL: reddinarasimha1998@gr	MOBILE: Mobile

Fig2.3 profile page for user

4 EDIT PROFILE PAGE:



User Profile

COMPLAINT BOX — Home | Profile | Change Password | Sign Out

FIRST NAME: First Name	MIDDLE NAME: Middle Name	LAST NAME: Last Name
GENDER: [✓ Male] Female	ROLL NUMBER: 15CO133	DEPARTMENT: COMPUTER SCIENCE /
HOSTEL: Hostel	ROOM NO: Room No	MESS NO: Mess No
MESS BLOCK: messblock	EMAIL: reddinarasimha1998@gr	MOBILE: Mobile

Save back to home

Fig2.4 edit profile page for user

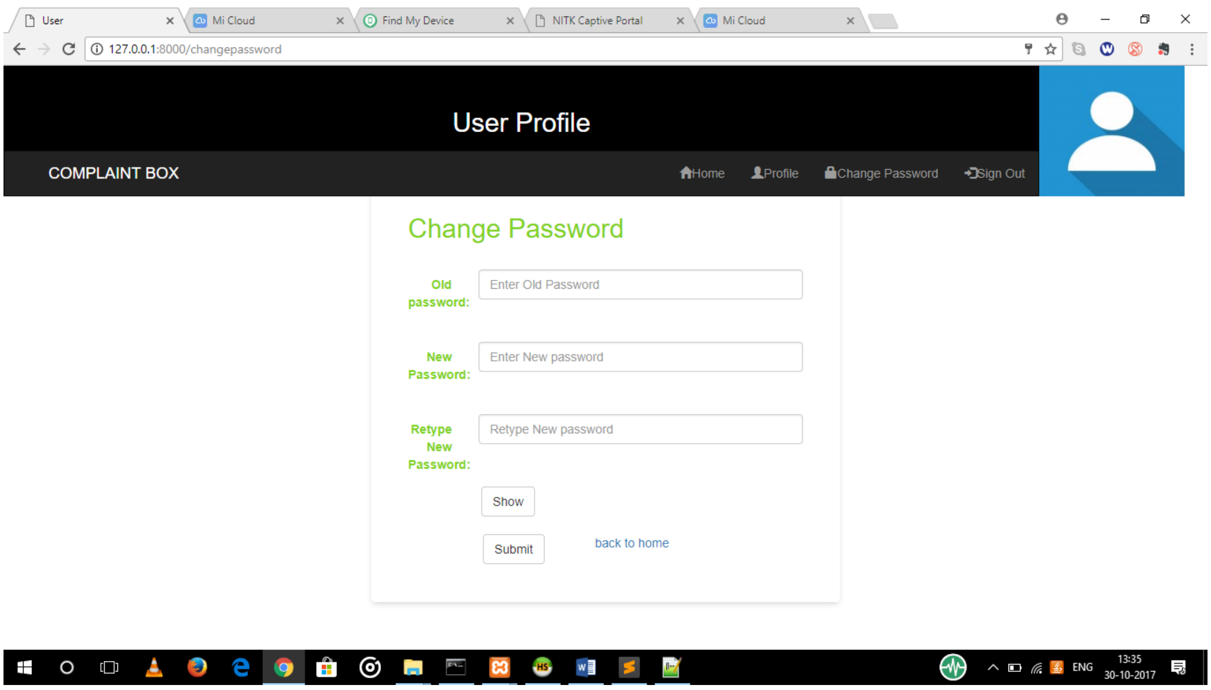
DESCRIPTION:

This page is displayed after clicking edit option in profile page. In this page student can edit his information like name, mess details, phone number and hostel details and gender. The information entered by the user is updated into database once he clicks save option. Information like roll number, email, department cannot be edited by student. Validation for each field in the field has been done by using javascript and php so that user can enter only valid information. If any invalid information is given, errors will be displayed on the page itself.

5 CHANGE PASSWORD PAGE:

DESCRIPTION:

When user clicks change password icon home page or any other page, this page will come on to the screen. Here, user change update his password. By entering old password and new password he can change his password. If user incorrectly enters old password and new password and confirmation password does not match, error messages will be displayed on the screen.



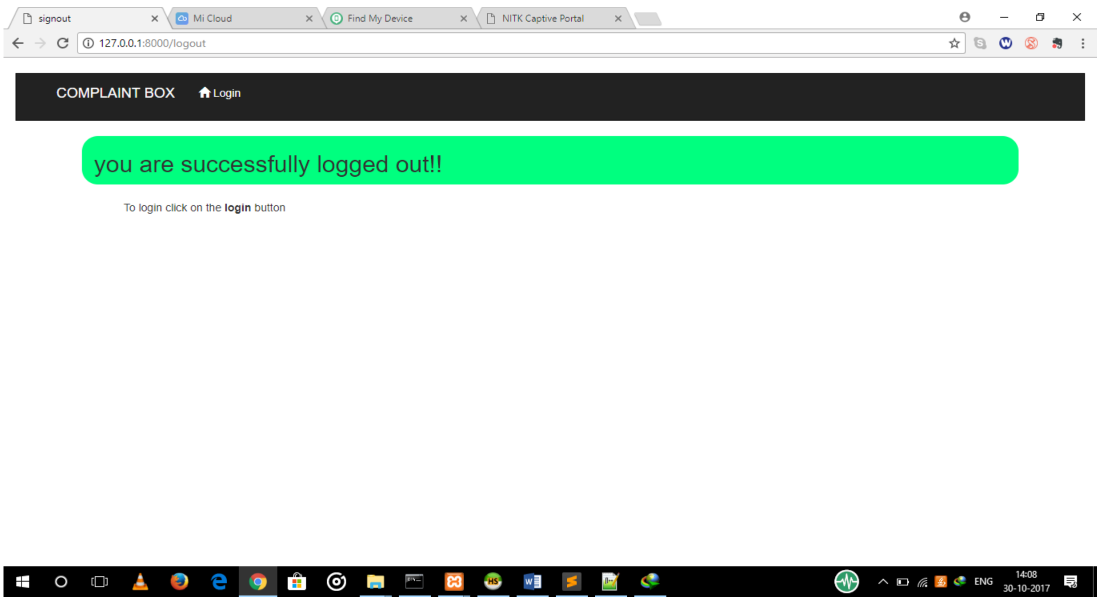
Change Password

Old password: Enter Old Password

New Password: Enter New password

Retype New Password: Retype New password

6 SIGN OUT PAGE:



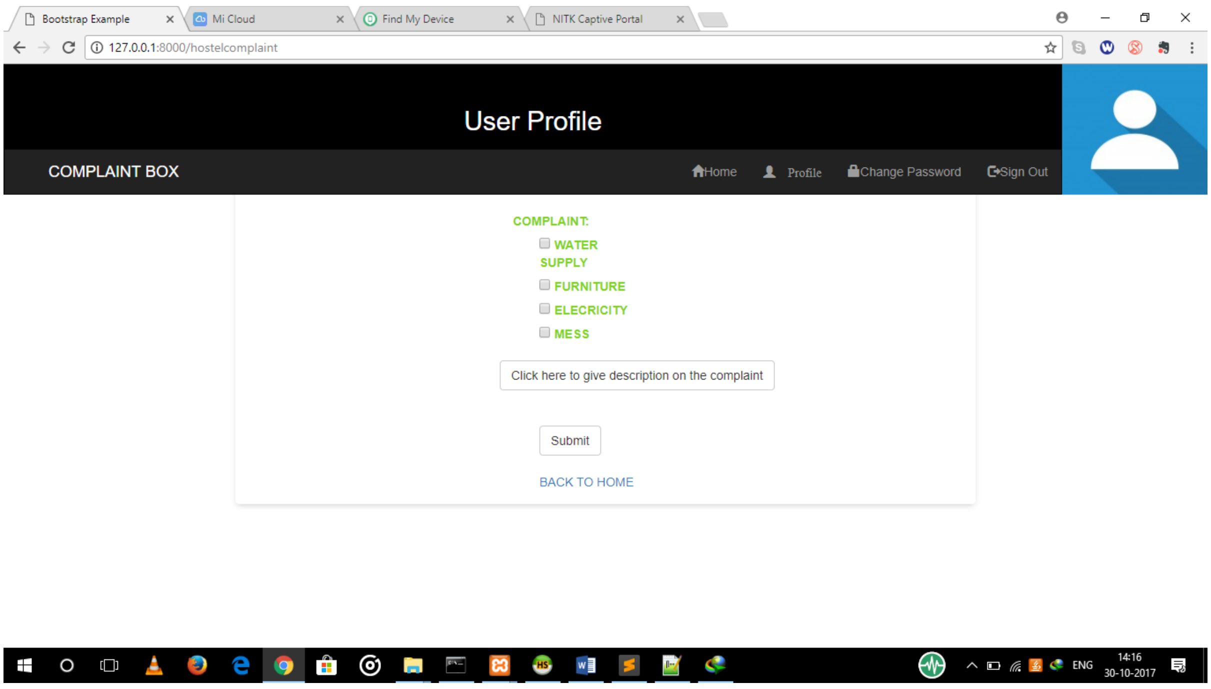
you are successfully logged out!!

To login click on the **login** button

DESCRIPTION:

When user clicks sign out icon on home page or any other page, this page will come on to the screen. User will logout from the application .If wishes to login again, he can do so by clicking on the login here in the page.

7 HOSTEL COMPLAINT PAGE:


Bootstrap Example | Mi Cloud | Find My Device | NITK Captive Portal

127.0.0.1:8000/hostelcomplaint

User Profile

COMPLAINT BOX | Home | Profile | Change Password | Sign Out

COMPLAINT:

- [ ] WATER SUPPLY
- [ ] FURNITURE
- [ ] ELECRICITY
- [ ] MESS

Click here to give description on the complaint

Submit

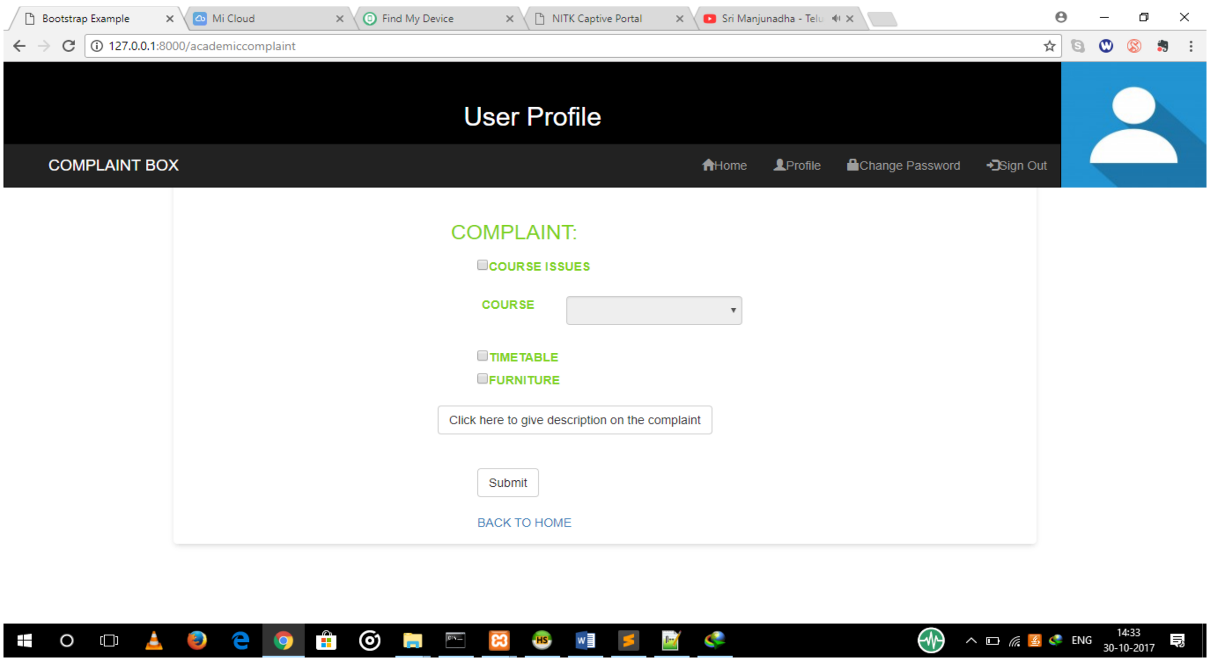
BACK TO HOME

ENG 14:16 30-10-2017

DESCRIPTION:

This page will be displayed when user clicks on hostel complaint form on dashboard page. In this page students are allowed to give hostel complaints. They can submit multiple complaints at a time with the help of check boxes provided on the page. They can also give a brief description of complaint. In the page, four categories of complaints has been shown. Once the user submits the complaint.

8 ACADEMIC COMPLAINT PAGE:


Bootstrap Example | Mi Cloud | Find My Device | NITK Captive Portal | Sri Manjunadha - Telu

127.0.0.1:8000/academiccomplaint

User Profile

COMPLAINT BOX | Home | Profile | Change Password | Sign Out

COMPLAINT:

- [ ] COURSE ISSUES

COURSE

- [ ] TIMETABLE
- [ ] FURNITURE

Click here to give description on the complaint

Submit

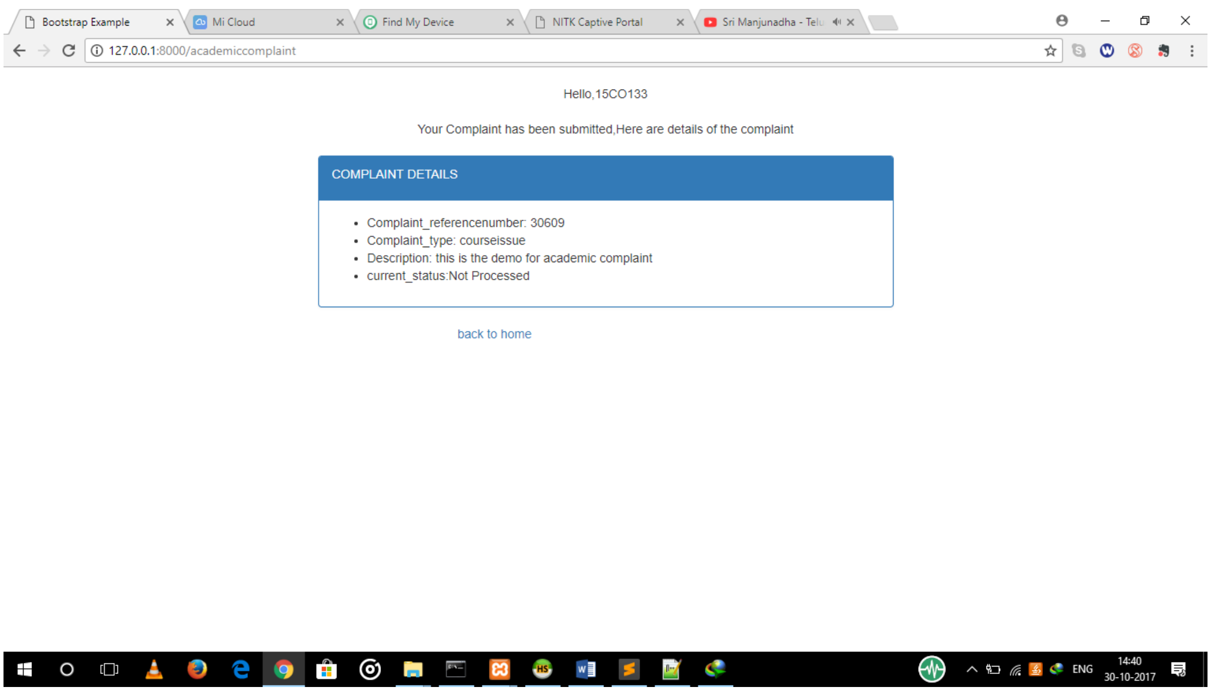
BACK TO HOME

ENG 14:33 30-10-2017

DESCRIPTION:

This page will be displayed when user clicks on academic complaint form on dashboard page. In this page students are allowed to give academic complaints. They can submit multiple complaints at a time with the help of check boxes provided on the page. They can also give a brief description of complaint. In the page, four categories of complaints has been shown. For course complaints, faculties and courses he is studying will be shown as drop down table, so that user can find it easy to give complaints of that type.

9 POST COMPLAINT PAGE:



Hello,15CO133

Your Complaint has been submitted,Here are details of the complaint

COMPLAINT DETAILS

- Complaint_referencenumber: 30609
- Complaint_type: courseissue
- Description: this is the demo for academic complaint
- current_status:Not Processed

back to home

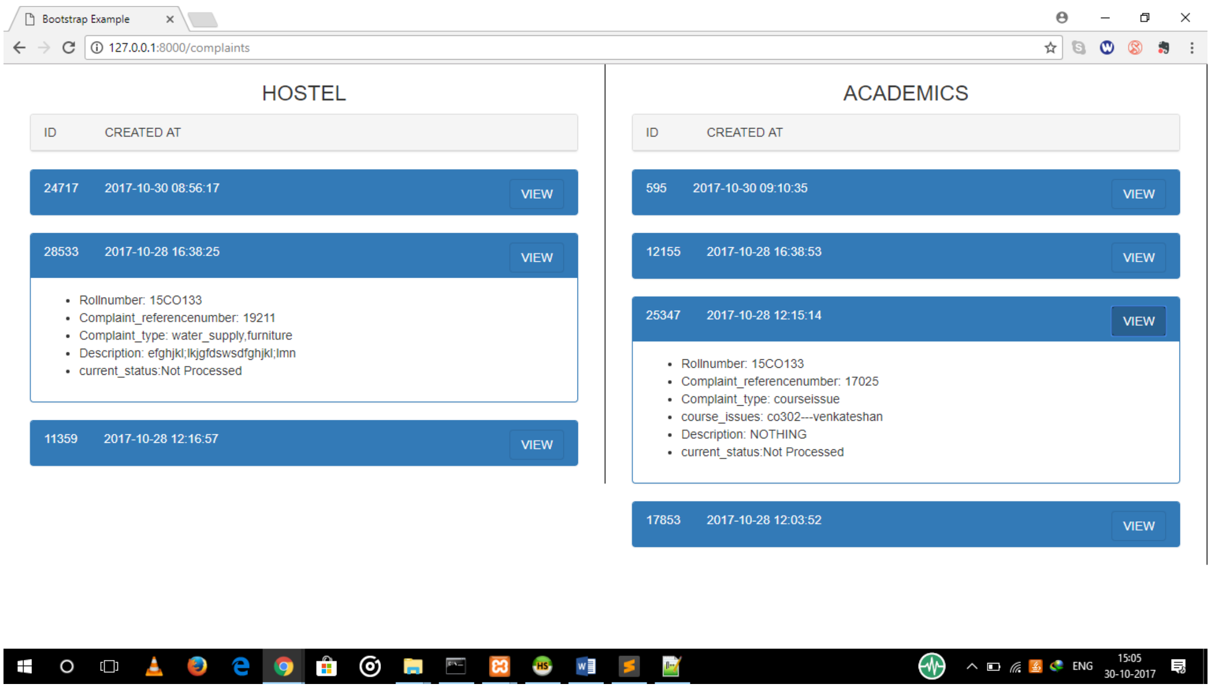
DESCRIPTION:

This page will be shown once the user submits either academic or hostel complaint. It contains details of the complaint submitted.

10 VIEW COMPLAINTS PAGE:

DESCRIPTION:

This page will be displayed when user clicks on view all complaints on dashboard page. Students can view all the complaints they have given till now. They can click on each complaint and can view the details of the complaint along with the status of the complaint.



Bootstrap Example

127.0.0.1:8000/complaints

HOSTEL

ID	CREATED AT	
24717	2017-10-30 08:56:17	VIEW
28533	2017-10-28 16:38:25	VIEW

- Rollnumber: 15CO133
- Complaint_referencenumber: 19211
- Complaint_type: water_supply,furniture
- Description: efghjkl;lkjgfdswsdfghjkl;lmn
- current_status:Not Processed

ID	CREATED AT	
11359	2017-10-28 12:16:57	VIEW

ACADEMICS

ID	CREATED AT	
595	2017-10-30 09:10:35	VIEW
12155	2017-10-28 16:38:53	VIEW
25347	2017-10-28 12:15:14	VIEW

- Rollnumber: 15CO133
- Complaint_referencenumber: 17025
- Complaint_type: courseissue
- course_issues: co302---venkateshan
- Description: NOTHING
- current_status:Not Processed

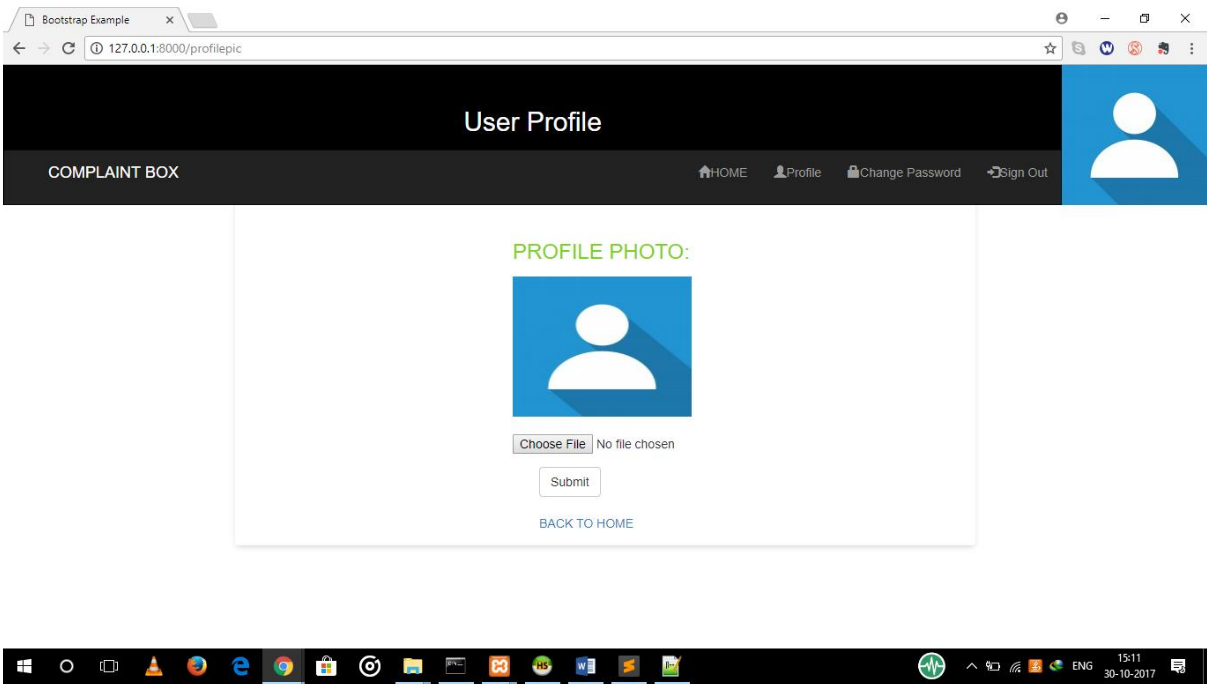
ID	CREATED AT	
17853	2017-10-28 12:03:52	VIEW

ENG 15:05 30-10-2017

11 PROFILE PIC PAGE:

DESCRIPTION:

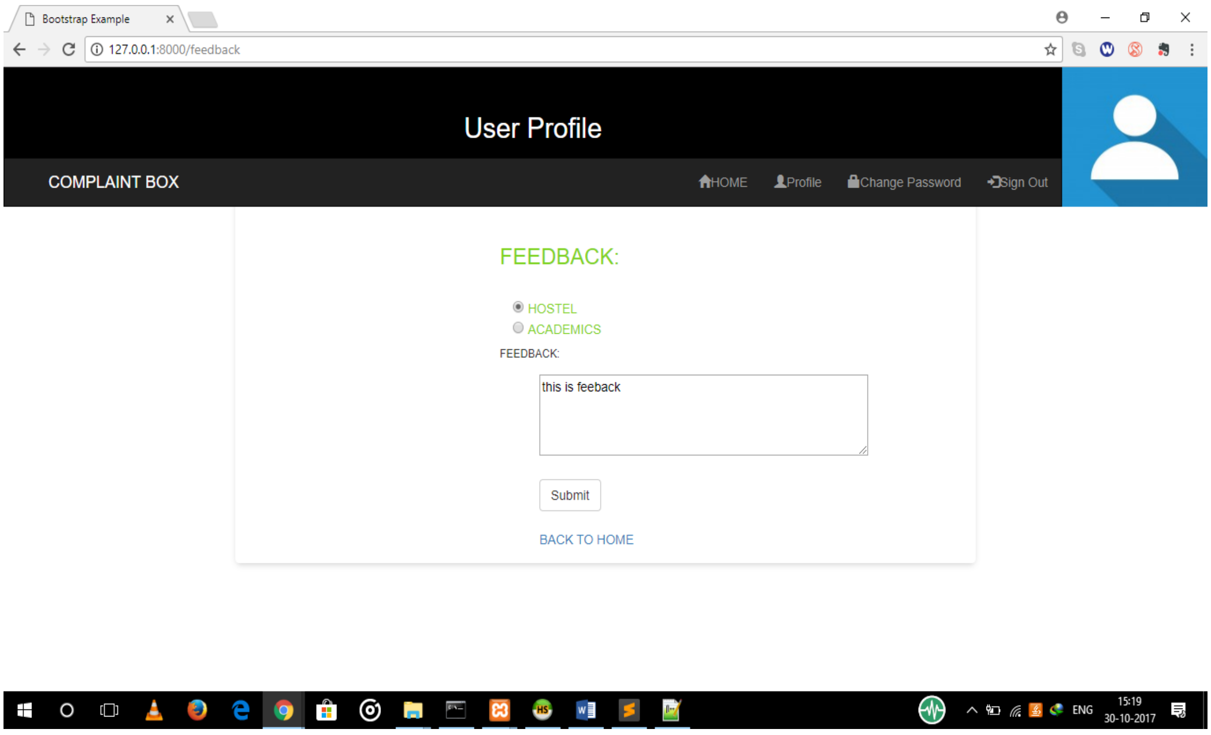
This page will be displayed when user clicks on profile photo on any page. Users can change their profile picture by uploading photo into it. After uploading photo, users can observe change in profile pic in every page.



12 FEEDBACK PAGE:

DESCRIPTION:

This page will be displayed when user clicks on give feedbacks on home page. User can give feedback of complaints at any time. Those feedbacks are anonymous. Once the user submits feedback, a success message will be displayed stating that feedback has been submitted successfully.



IMPLEMENTATION ON ADMIN SIDE:-

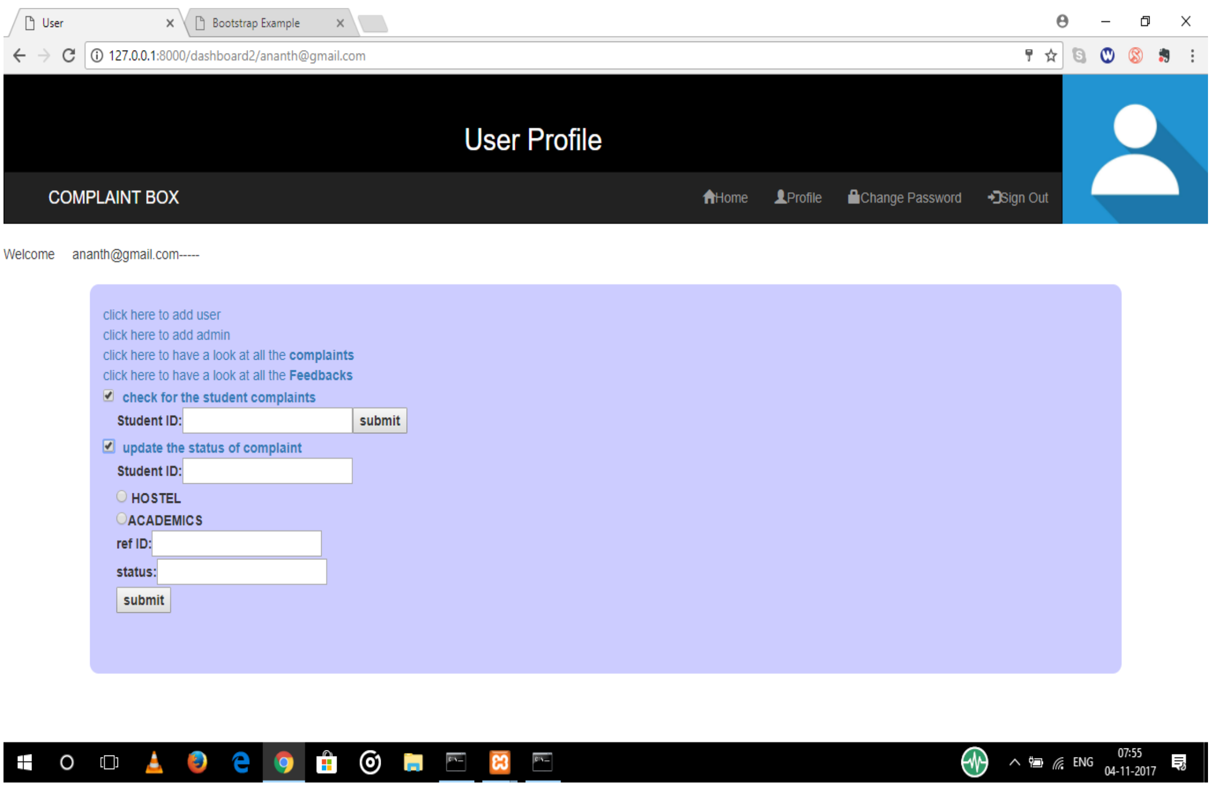
On admin side login page, dashboard page, change password page, sign out page, profile pic page, view complaints page, profile page will be same as that of student's page.

Here are the screen shots and description for above pages;

1 DASHBOARD PAGE:

DESCRIPTION:

This page will be similar to that of student's page except that of body section where it contains links to add student, add admin, view complaints, view feedbacks and to search for a particular student complaints and to update the status of the complaints.


127.0.0.1:8000/dashboard2/ananth@gmail.com

User Profile

COMPLAINT BOX

Home Profile Change Password Sign Out

Welcome ananth@gmail.com-----

click here to add user
click here to add admin
click here to have a look at all the complaints
click here to have a look at all the Feedbacks

✓ check for the student complaints

Student ID: submit

✓ update the status of complaint

Student ID:

HOSTEL

ACADEMICS

ref ID:

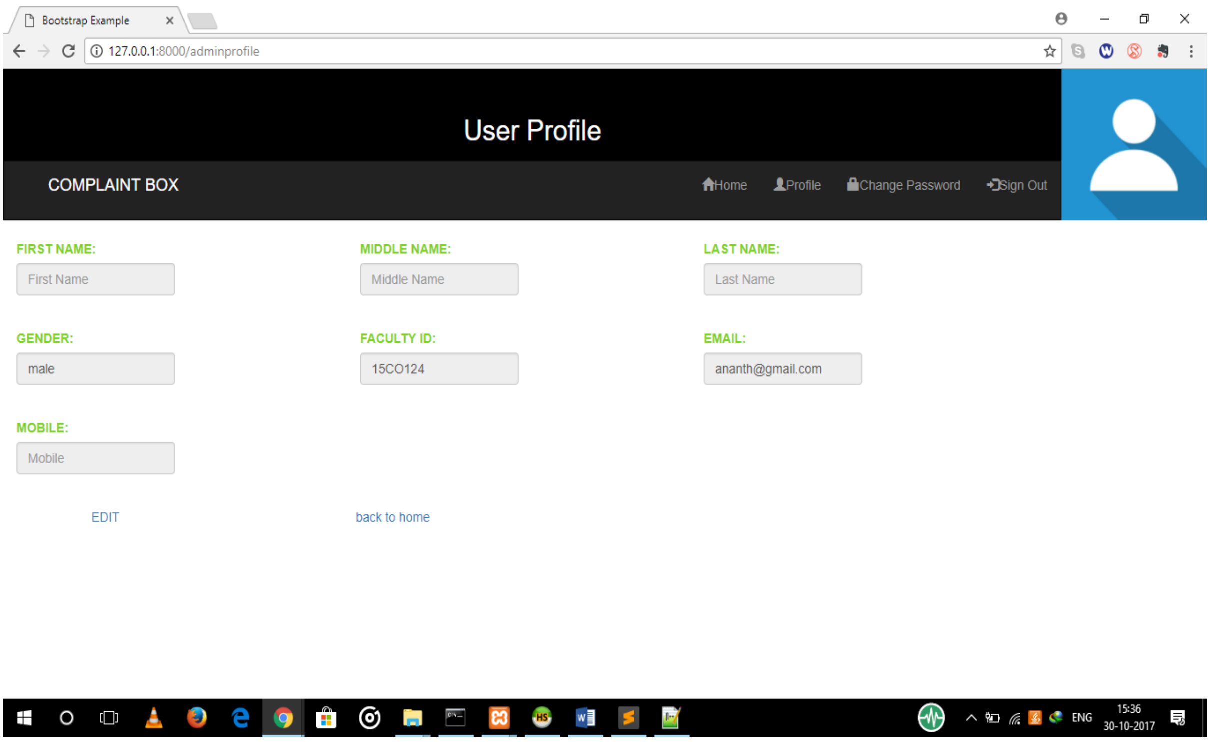
status:

submit

2 PROFILE PAGE:

DESCRIPTION:

This page will be similar to that of student's page except that it does not contain some fields which were present in student's page.


Bootstrap Example

127.0.0.1:8000/adminprofile

User Profile

COMPLAINT BOX

Home Profile Change Password Sign Out

FIRST NAME: First Name	MIDDLE NAME: Middle Name	LAST NAME: Last Name
GENDER: male	FACULTY ID: 15CO124	EMAIL: ananth@gmail.com
MOBILE: Mobile		

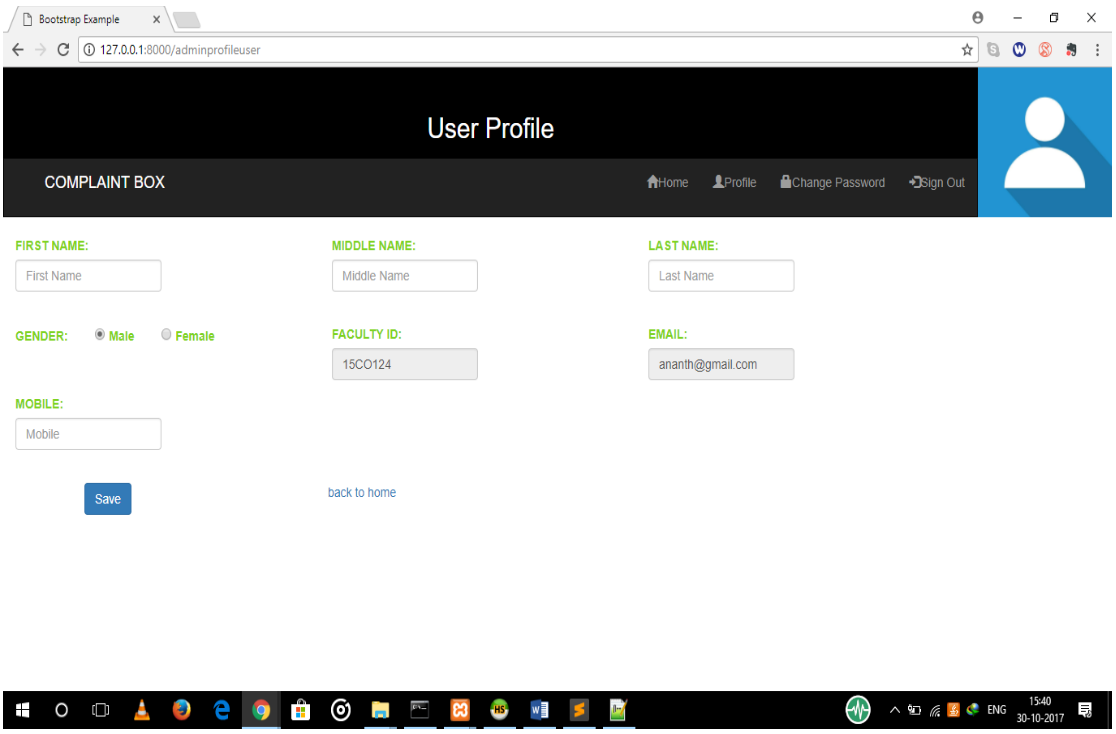
EDIT back to home

Admin can edit his profile by clicking on the edit option given on the bottom of page.

3 EDIT PROFILE PAGE :

DESCRIPTION:

This page will be similar to that of student's page except that it does not contain some fields which were present in student's page.


Bootstrap Example

127.0.0.1:8000/adminprofileuser

User Profile

COMPLAINT BOX

Home Profile Change Password Sign Out

FIRST NAME: First Name	MIDDLE NAME: Middle Name	LAST NAME: Last Name
GENDER: [✓ Male] Female	FACULTY ID: 15CO124	EMAIL: ananth@gmail.com
MOBILE: Mobile		

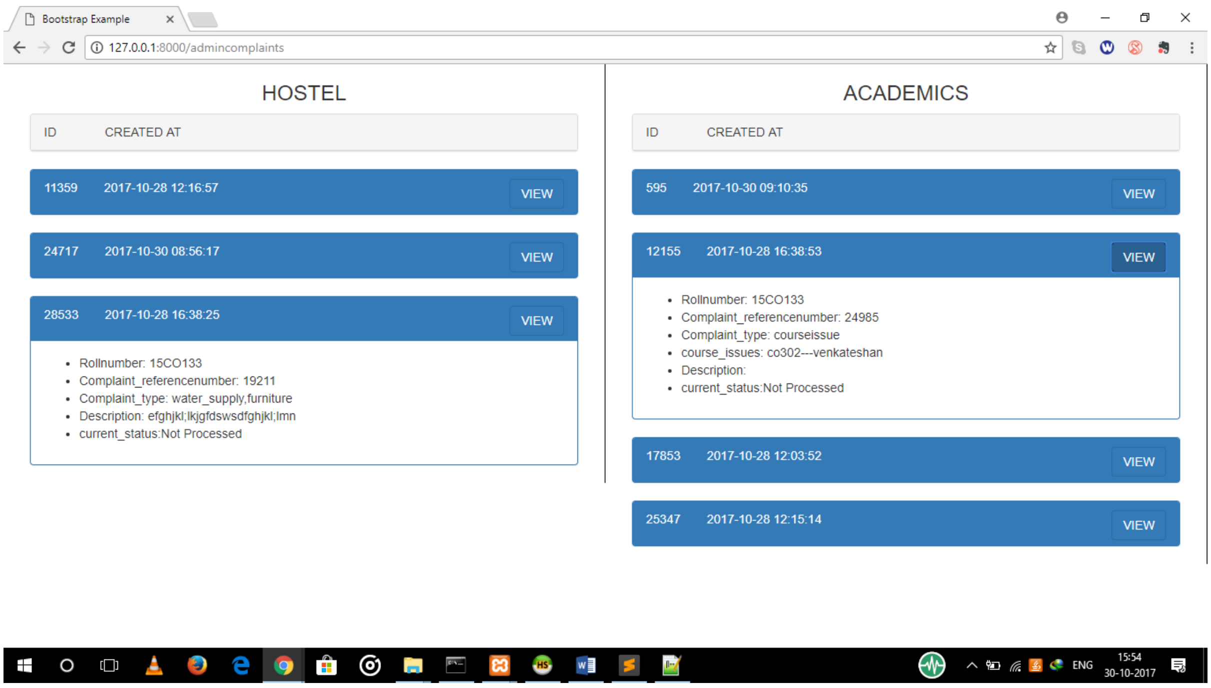
Save back to home

Admin can update his profile in this page and information will be directly stored in to the database.

4 VIEW COMPLAINTS PAGE:

DESCRIPTION:

This page will be similar to that of student's view complaint's page except that it contains complaints of all students.



HOSTEL

ID	CREATED AT
11359	2017-10-28 12:16:57
24717	2017-10-30 08:56:17
28533	2017-10-28 16:38:25

- Rollnumber: 15CO133
- Complaint_referencenumber: 19211
- Complaint_type: water_supply,furniture
- Description: efghjkl;lkjgfdswsdfghjkl;lmn
- current_status:Not Processed

ACADEMICS

ID	CREATED AT
595	2017-10-30 09:10:35
12155	2017-10-28 16:38:53

- Rollnumber: 15CO133
- Complaint_referencenumber: 24985
- Complaint_type: courseissue
- course_issues: co302---venkateshan
- Description:
- current_status:Not Processed

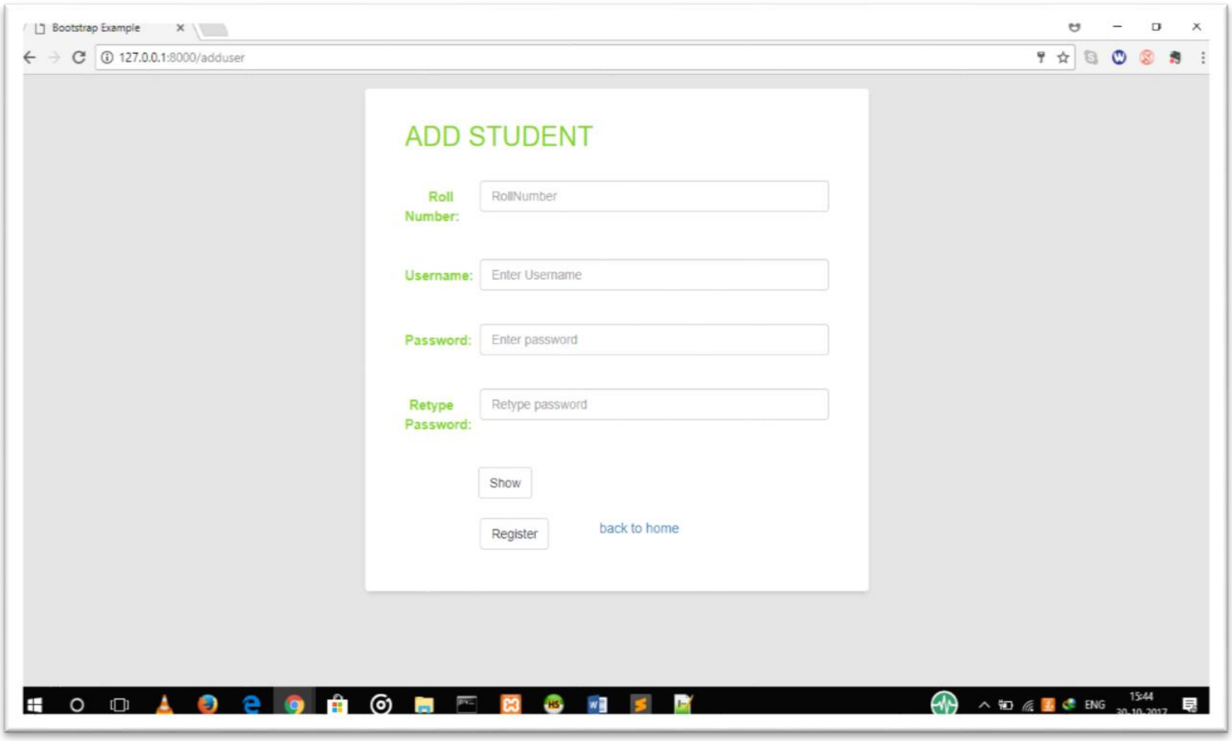
ID	CREATED AT
17853	2017-10-28 12:03:52
25347	2017-10-28 12:15:14

Additional pages that are implemented on admin side:-

1 ADD USER PAGE:

DESCRIPTION:

This page will be displayed when admin clicks on add user link on home page. Admin adds student into database in this page by providing his/her roll number, email and random password will be given to the student.



Bootstrap Example

127.0.0.1:8000/adduser

ADD STUDENT

Roll Number: RollNumber

Username: Enter Username

Password: Enter password

Retype Password: Retype password

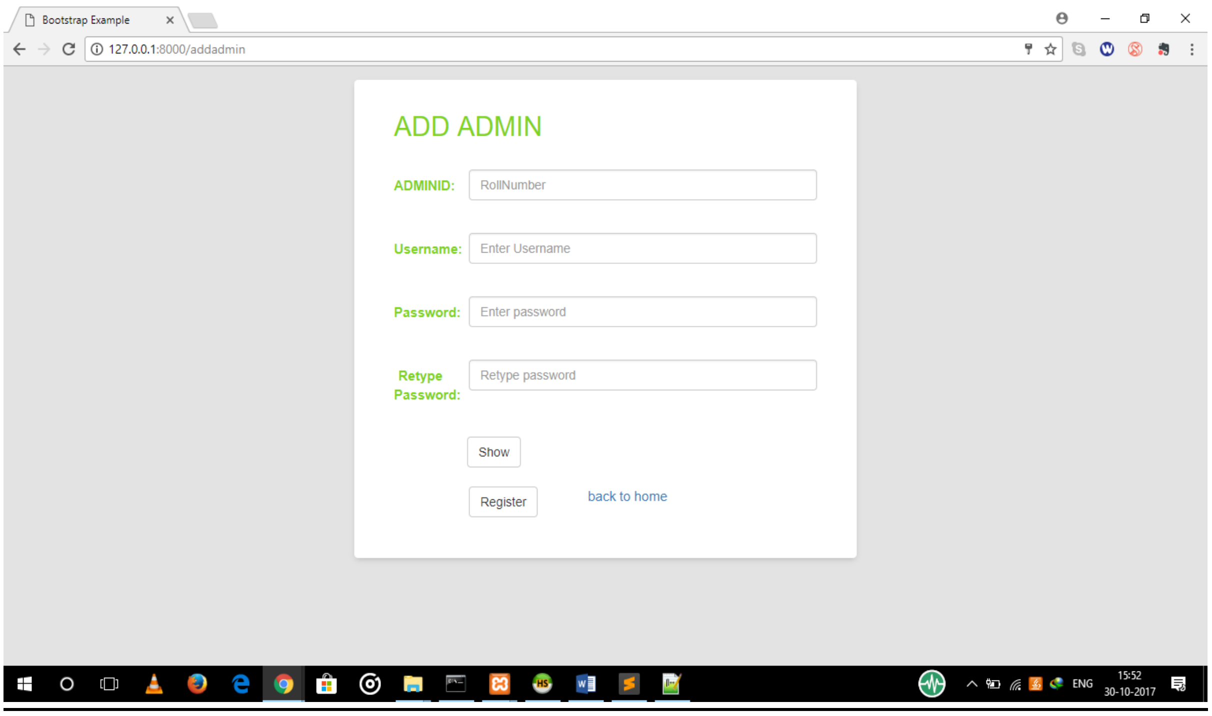
Show

Register back to home

3 ADD ADMIN PAGE:

DESCRIPTION:

This page will be displayed when admin clicks on add admin link on home page. Admin adds another into database in this page by providing his/her id, email and random password will be given to the student. This page is similar to that of add user page.



Bootstrap Example

127.0.0.1:8000/addadmin

ADD ADMIN

ADMINID: RollNumber

Username: Enter Username

Password: Enter password

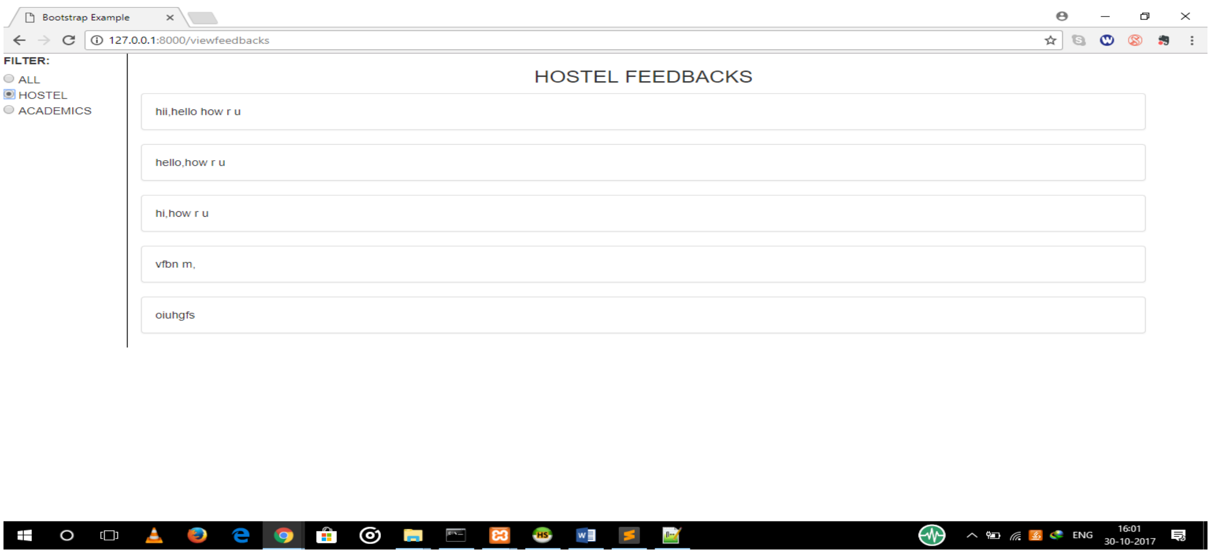
Retype Password: Retype password

Show

Register back to home

3 VIEW FEEDBACKS PAGE:

DESCRIPTION: This page will be displayed when admin clicks on view feedbacks link on home page. Admin can view all feedbacks submitted by students. He can filter feedbacks based on hostel and academics



Bootstrap Example

127.0.0.1:8000/viewfeedbacks

FILTER:

ALL [✓ HOSTEL] ACADEMICS

HOSTEL FEEDBACKS

hii,hello how r u

hello,how r u

hi,how r u

vfbn m,

oiuhgfs

8 CONCLUSION AND FUTURE WORK:

The final conclusion of our report is creating a database with a good user interface to register the complaints faced by the students. This report covers the project description and purpose and aim of the project. It also includes the entire process of making the project, requirement analysis and specifications for the system and the tools used to prepare the modules of the project and the languages used for having better interface and also for the database queries.

As the process of making project mentioned above can still be made some adjustments or improvements to get a better output.

Regarding the future work we will try to implement the automatic response functionality (like sending a message to registered mobile or email of the student) whenever student registers the complaint or any change in the status of complaint