**ABSTRACT**

“HOSTEL MANAGEMENT SYSTEM” is software developed for various activities in hostel

From the past few years the number of educational institutions are increasing

rapidly .

There by number of hostels is also increasing for the accommodation of the

students studying in this institution and hence there is a lot of strain in the

person who are running the hostel and software’s are not usually used in this

context.

This particular project deals with problems on managing the hostel and avoids the

problems which occur when carried manually. This eliminates the time delay and paper

transactions being marked.

The Admin is provided with better control over the transactions like adding the details

of new students in the hostel, modifying the details of the students, deleting the

students, viewing the students details in the hostel.

The common transactions of the hostel includes the maintenance of information

about the students in the hostel , enrolling of the new students and payments etc.

are stored into the databases and reports are generated according to the users

requirements .

Identification of the drawbacks of the existing system leads to the designing of

computerized system and that will be compactable to the existing system with the

system which is more users friendly and more GUI oriented.

**ABSTRACT**

**ACKNOWLEDGMENT**

**TABLE OF CONTENTS**

**LIST OF FIGURES**

Chapter 1**. INTRODUCTION**

* 1. Background
  2. Introduction about the project

Chapter 2. **FRONTEND AND BACKEND**

2.1 About front end

2.2 About back end

Chapter 3.  **SPECIFICATIONS**

3.1 Hardware Requirements

3.2 Software Requirements

3.3 Functional Requirements

3.4 Non-Functional Requirements

Chapter 4. **METHODOLOGY**

4.1 Introduction

4.2 Design Strategy

4.3 Input Design

4.4 Output Design

4.5 ER Diagram

4.6 Schema Diagram

Chapter 5**. IMPLEMENTATION**

5.1 Tables Description

5.2 Stored Procedure

5.3 Triggers

Chapter 6. **RESULTS AND SCREENSHOTS**

Chapter 7. **CONCLUSION AND REFERENCES**

**LIST OF FIGURES**

**Fig. No. Descriptions Page**

Fig 2.1 ER diagram of hostel management system

Fig 2.2 Relational Schema of hostel management system

Fig 4.1 Home Page of Hostel Management System

Fig 4.2 User Registration

Fig 4.3 Home page of User

Fig 4.4 Profile of Student

Fig 4.5 Change of password window

Fig 4.6 Room details

Fig 4.7 Access Log details

Fig 4.8 Admin Login

Fig 4.9 Admin Profile

Fig 4.10 Add course window

Fig 4.11 Add room window

Fig 4.12 Registration form for the user

Fig 4.13 Manage Student window

Fig 4.14 Access Log window

**Chapter 1**

**INTRODUCTION**

* 1. **Background**

A database is an organized collection of data, generally stored and accessed

electronically from a computer system where database are more complex they are

often developed using formal design and modeling techniques.

The database management system (DBMS) is the software that interacts

with end-users ,Applications, the database itself to capture and analyze the data and

provide facilities to administer the database.

The sum total of database, the DBMS and the associated applications can be

referred to as “DATABASE SYSTEM” .Often the “database” is also used to loosely

referred to any of the DBMS, the database system or an application associated with the

database.

The DBMS three important things: the data, the database engine that allows data

to be accessed, logged and modify the database schema, which defines the database’s

logical structure. These three foundational elements help provide concurrency, security,

data integrity and uniform administration procedures. the typical database administration

task supported by the DBMS include change management , performance monitor/tuning

and backup and recovery many database management systems are also responsible for automated roll backs, restarts and recovery as well as the logging and auditing of

activity.

* 1. **Introduction about the Project**

The need to create the hostel management system(HMS) is to organize the rooms, students

record and the other information about the students. How many students can live in a

room ,and the students of the hostel can be recognized from the register number.

This software product hostel management to improve the services of all

the students of the hostel. This also reduce the manual work of the person in

admin panel and bundle of registers that where search when to find the information of a previous student , because through this system you can store the data of those

students who had leaved the hostel.

Through the Admin login the admin can check the profiles of all the current students within few minutes the database of the system will help you to check a particular one. Through the user registration A student can create his profile by providing his basic information.

Through the user login the user can login into the profile and one can register the room as per their requirements and based on his requirements the updating of information can be done. The permission to user login is granted only to the students who have used user registration .

**Chapter 2**

**FRONTEND AND BACKEND**

**2.2 About Front End**

In order to make the site dynamic and more interactive we have tried to include

a database link to our Hostel Management System. Hence Admin can update

student details and the facilities detail, room details , course details and the login

details.

The students can also edit their profiles and update them ,to perform the

operations initially the users should register and login and the user can allocate the

room as per his requirements.

1. **HTML**

Hyper Text Markup language (HTML) is the standard markup language for creating web

pages and web applications. with Cascading Style Sheets(CSS) and JavaScript it

forms a triad of cornerstone technologies for the World Wide Web .Web browser

receives HTML documents from a web server or from local storage and render

them into multimedia web pages.HTML describes the structure of a web pages

semantically and originally included cues for the appearance of the documents.

HTML elements are the building blocks of HTML pages. With HTML

constructs, images and other objects , such as interactive forms, may be embedded into

the rendered pages .It provides a means to create structured documents by denoting

structural semantics for text such as headings, paragraphs ,lists , links quotes and other

items. HTML Elements are delineated by tags, written using angular brackets .Tags such

as <img/> and <input/> introduce content into the page directly .Others such as

<p>..</p> surrounds and provides information about document text and may include

other tags as sub-elements .Browsers do not display the HTML tags, but use them to

interpret the content of the page. HTML can embedded written in a scripting

languages JavaScript which affects the behavior and content of the web pages

.Inclusion of CSS defines the look and layout of content.

1. **CSS**

Cascading Style sheets (CSS) is a style sheet language used for describing the

presentation of a document written in a markup language .Although most often used

to set the visual style of web pages and user interfaces written in HTML and XHTML,

the languages can be applied to any XML documents, including plain XML,SVG AND

XUL, and is applicable to rendering in speech , or no other media .Along with HTML

and JavaScript ,CSS is a cornerstone technology used by most websites to create visually

engaging wed pages ,user interfaces for web applications ,and user interfaces for many

mobile applications.

CSS is designed primarily to enable the separation of presentation and

content, including aspects such as the layout ,colors and fonts. This separation can

improve content accessibility , provide more flexibility and control in specifications of

presentation characteristics , enable multiple HTML pages to share formatting by

specifying the relevant CSS in a separate.CSS in a separate .css file, and reduce

Complexity and repetition in the structural content

1. **JAVA SCRIPT**

JavaScript (sometime abbreviated JS) is prototype-based scripting language that is

dynamic, weekly typed, general purpose programming language and has first-class

fuctions.It is a multi-parading language , supporting object oriented, imperative and

functional programming styles.

JavaScript was formalized in the ECMA script language standard and is

primarily used in the form of client-side JavaScript ,implemented as part of a Web

browser in order to provide enhanced user interfaces and dynamic websites .This enables

programmatic access to computational objects within a host environment. JavaScript‟s

use in application outside Web pages for example in PDF documents ,site-specific

browsers, and desktop widgets in also significant. In this applications, JavaScript is

used for validation purpose like text box validation ,email validation , phone number

validation. Java Script is the good tool for validating the web-applications.

1. **APACHE SERVER**

Apache HTTP server, colloquially called Apache, is free and open-source cross-platform

web server software, released under the terms of Apache License 2.0.Apache is

developed and maintained by open community of development under the auspices of the

Apache Software Foundation.

Apache supports a variety of feature, May implemented as complied modules

which extend the core functionality .These can range from server-side programming

language support to authentication schema .Some common languages interfaces support

Perl, python, Tcl, and PHP. Popular authentication modules include mod\_access,

mod\_auth, mod\_digest, and mod\_auth\_digest, the successor to mod\_digest. A sample of

other features secure socket layer and Transport layer security support ,a proxy module, a

URL rewriting module ,custom log file, and filtering support(mod\_include and

mod\_ext\_filter).

**2.2 About Back End**

1. **MYSQL**

MySQL is based on a client-server model. The core of MySQL is MySQL server, which

handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client- server networked environment and as a library that can be embedded (or linked) into seperate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer. MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.SQL is a small, fast and embeddable database where the database engine and the interface are combined into a single library. It also has the ability to store all the data in a single file. It is a software library that implements a selfcontained, serverless, zero configuration, and transactional SQL database engine. SQL is the most widely deployed SQL database engine in the world. All the SQL statements start with any of the keywords like select, create, update, insert, delete, alter, drop etc., and all the statements end with a semicolon (;).

▪ Connection to Mysql Server

$connection= mysql\_connect (“localhost”, $username, $password);

▪ Access the database

Mysql\_select\_db (“databasename”, $connection);

▪ Perform SQL operations

Example: $result=mysql\_query ($query, $connection)

▪ Disconnect from the server

Mysql\_close ($connection)

**B.PHP**

PHP is a server –side scripting language designed primarily for web development but also

used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994 , the PHP reference implementation is now produces by the PHJP Development Team.PHP originally stood for personal Home Page , but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template system, web content management system and

web frameworks.PHP code is usually processed by a PHP interpreter implemented as a

module in the web server or as a common Gateway Interface(CGI) executable. . The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphically applications.

The standard PHP interpreter, powered by the Zend Engine, is free software

released under the PHP License.PHP has been widely ported and can be deployed on

most web servers on almost every operating system and platform free of charge.The PHP language evolved without a written formal specifications or standards until 2014,

leaving the canonical PHP.

Interpreter as a de facto standard .Since 2014 works has gone on to create a formal

PHP specifications .HP development began in 1995 when Rasmus Lerdorf wrote several

Common Gateway Interfaces (CGI) program in c, which he used to maintain his

personal homepage. He extended them to work with the web forms and to

communication with database, and called this implementation “Personal Home

Page/Forms Interpreter” or PHP/FI.

**CHAPTER 3**

**SPECIFICATIONS**

3.1 HARDWARE REQUIREMENTS

* PROCESSOR : Intel Pentium or Higher Version
* RAM : Minimum 1GB
* HARD DISK : 60GB and above

3.2 SOFTWARE REQUIREMENTS

* SOFTWARE : XAMP
* SUPPORTED BROWSERS : Google Chrome / Mozilla Firefox / Internet Explorer
* Notepad++
* Windows Vista\* or Newer Version, or MACos, or Linux (32/64 bit)

3.3 FUNCTIONAL REQUIREMENTS

The Functional Requirements Specification documents the Operations and activities that a system must be able to perform. Functional Requirements include:

* Addition, Deletion, and modification of records
* Connection ( New customer, valid user)
* Generation of bills
* Customer record (customer name, address, phone)
* Book details (price, description)
* Login (Login ID, password)

The functional requirements specifications is designed to be read by a general audience. Readers should understand the system, but no particular technical knowledge should be required to understand the document.

These are the functional requirements specification documents for the project analysis. A software requirement specification helps to alternate the time and energy needed by the developers to attain their desired goals and additionally minimizes the value of development.

Following Factors are used to measure software development quality:

Each attribute may be accustomed measure of the product performance. These attributes may be used for Quality assurance similarly as quality control. Quality assurance activities are directed towards prevention of introduction of defects and internal control activities are aimed toward detecting defects in product and services.

1. Reliability

Measure if product is reliable enough to sustain in any condition. Give systematically correct results. Product dependability is measured in terms of operation of project underneath different operating atmosphere and different conditions.

1. Maintainability

Different versions of the product ought to be easy to maintain. For development it ought to be easy to feature code to existing system, ought to be easy to upgrade for brand new options and new technologies time to time. Maintenance ought to be value effective and simple. System be easy to take care of and correcting defects or making a change within the software system.

1. Usability

This can be measured in terms of ease of use. Application should be user friendly. Easy to use for input preparation, operation and also for interpreting of output.

1. Portability

This can be measured in terms of Costing issues related to porting, Technical issues related to porting, Behavioural issues related to porting.

3.4 NON FUNCTIONAL REQUIREMENTS

Satisfactory will probably not be assessed on the system where the program is developed, tested or first installed.

**CHAPTER 4**

**SYSTEM DESIGN**

4.1 INTRODUCTION

System design is the first design stage for devising the basic approach to solving the problem. During system design, developers decide the overall structures and styles. The system architecture determines the organization of the system into subsystems. In addition, the architecture provides the context for the detailed decisions that are made in later stages .during design, developers make decisions about how the problem will be solved, first at the high level and then with more detail.

* 1. DESIGN STRATEGY

The design strategy is a vital aspect of the system to be developed. The design of the software reflects the basic understanding of the problem. For designing a good system what we have to be is to get correct definition of the problem and analyze the problem thoroughly.

The design of a system should be such that if a small portion is changed. The rest of the system should be unaffected. This is the flexibility of the system. Greater the system flexibility greater will be the system reliability. While carrying out the job of designing of a new system one has to consider many factors. These factors include the drawbacks and limitations of the present manual system as well as of the features and advantages of the proposed system. It should be designed in such a manner that even a layman can run it without any difficulty.

An important quality of a software must enjoy is “user friendliness”. It can be achieved in many ways like providing menu, giving context sensitive help, doing automatic validation to input data, etc. another main factor is speed efficiency. In order to achieve speed efficiency, the program should be designed accordingly and the user is provided with a compiled copy of the software package with necessary data file format rather than source code.

Design of input and output formats is equally important for any design. The output format should be designed in such a way that it must reflect all the required information in detail. The design of the database itself such as type of data stored, size of data etc. some of the decisions made doing database design are:

* Which data items are to be recorded and in which database.
* Determine what information is to display.
* Decide whether to display or print the information retrieved, processed, generated from the system.
* Arrange the presentation of information in the acceptable format.
* Decide how to distribute the output to the intended recipients.
* Length of each record, based on the characteristics of the data items on which it is based.
* Data who’s unauthorized change must be prevented.
* Data which must be avoided from redundancy.
* Maintenance of data integrity etc.
* Avoid over writings.
* Prevents invalid data access and changes.

Having all this, a positive interaction with clients at every stage of development is the core around which software is built.

* 1. INPUT DESIGN

Input design is the process of converting user-originate inputs to a computer-based format. The goal of design input data is to make data entry as easy, logical and free. The most common source of data processing errors is inactive input data. Effective design of the input data minimizes the error made by data entry operators. Catching errors on input is far less costly than correcting after data storage is complete.

User friendly input design enables quick error detecting and correction. Verification and validation is the most important in input design. Since the system is used interactively, it has two types of inputs. Interactive input-which is the point contact of the user with the system and the input to the internal system i.e. databases.

For full efficiency of the system, it is necessary that the input must be accurate. Since the user of the system may not be a technical person and may not know input concepts so it is required that he warn, prevent and correct invalid data entry.

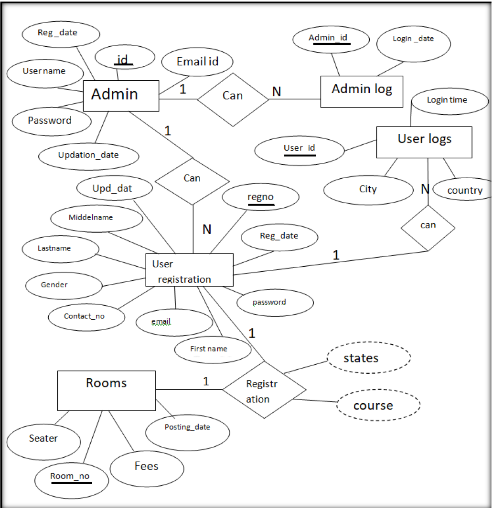
There are many ways that can be designed to handle such a situation. We can prevent the user entering invalid data into the databases by warning. Neglecting or messaging appropriately. The user is then allowed to input correct data. Some help provisions may aid the user to point out the error. In this system inputs are collected from terminals through keyboard.

4.4 OUTPUT DESIGN

Output design has been an ongoing activity from the very beginning of the project. The objective of the output design is to convey the information of all past activities, current status and to emphasize important events. The output generally refers to the results and information that is generated from the system.

The output design of the system is accomplished keeping in mind what should be displayed. we present the information in an acceptable format. Decide how to distribute the output to the intended recipients.

* 1. E R DIAGRAM



* 1. **ER DIAGRAM FOR HOSTEL MANAGEMENT SYSTEM**

Description of ER Diagram

The ER diagram in the Fig: 1.1 describe entities, attributes and relationships.

Relationships like Can, Registration are in diamond boxes, attached to entity types with straight lines.

∙ Attributes are shown in ovals, each attached by a straight line to entity or relation

type and .key attributes (like id) are underlined.

∙ The entities are represented in the Rectangular boxes.

∙ Components attributes of a composite attributes are attached to oval representing

it.

The above ER diagram contains the following entities and their related attributes

❖ ADMIN

1. Id

2. Email id

3. Reg \_date

4. User name

5. Password

6. Updating \_date

❖ ADMIN LOG

1. Admin\_id

2. Login date

❖ USER REGISTRATION

1. Reg\_no

2. Reg\_date

3. Password

4. First name

5. Middle name

6. Last name

7. Email

8. Contact \_no

9. Gender

10. Upd\_date

❖ USER LOGS

1. User \_id

2. City

3. Country

4. Login time

❖ ROOMS

1. Room-no

2. Seater

3. Fees

4. Posting date

**2.1.1 E-R Diagram Relationships Description**

1.Admin :Admin log is of the cordiality 1:N as one admin can have more than

one Admin log and therefore connected via „can‟ relationship. There is no total

participation of both entities.

2.Admin: User registration which is of the cardinality 1:N as one admin can register

N number of users and therefore connected via „can‟ relationship. There is no total

participation of both the entities.

3.User registration : user log which is of the cardinality 1:N as one user can have N

number of logins with his email and password and therefore connected via „can‟

relationship . There is no total participation of both the entities

.

4.User registration : Rooms which is of the cardinality 1:1 as one user can register

only one room in the hostel and therefore connected via „registration‟ relationship .

There is no total participation of both the entities.

**2.2 RELATIONAL SCHEMA DIAGRAM**

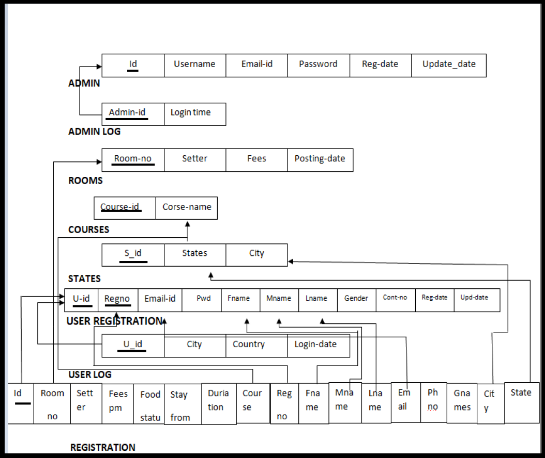


Fig .2.2 Relational schema of Hostel Management System

**2.2.1 Constraints General**

1) NULL Constraint: Attributes that are under NOT NULL constraints have to be filled

compulsorily. Almost all the attributes in the project are under NOT NULL constraint

2) Entity Integrity Constraint: This constraint makes sure that no primary key can have a

NULL value assigned to it. The primary keys involved in the project include:

• Id

• Room no

• User-id

• Course-id

• S-id

3) Referential Integrity Constraints: A table in the back end of the project may have

references pointing to an attribute in another table. For example: The REGISTRATION

table the room-no are referred to the room-no in the ROOMS table. .The various tables

are also linked with multiple foreign keys which are all set to cascade any update or

delete operation on the attribute in the main table. The various Foreign Key attributes are:

∙ Room-no

∙ Email-id

∙ Fname

∙ Mname

∙ Lname

**2.2.2 Schema Description**

The above Fig2.2 shows the relational schema of Hostel Management System. IT

has the following entities.

∙ ADMIN: The admin is the master table where the admin can perform some

operations that the user cannot do it. Such as adding of the new rooms and

addition of the new courses etc..so The table is the master table.

∙ ADMIN LOG : This contains the login information of the Admin for the

security purpose ,So that we can know who has login into the admin login.

∙ ROOMS: This table contains the information about the rooms that are available

in the hostel and the related information about the rooms.

∙ COURSE: This table contains the information related to the subject that are

taken or being done by the users or the students who stay in the hostel.

∙ STATES : This table contains the information of the city and states from where the users are from.

∙ USER REGISTRATION: This table contains the information of the people who

have registered to the hostel database.

∙ USER LOG: This contains the information of the users login with the date of

login.

∙ REGISTRATION: This contains the information of the user/students who

have registered the hostel room. If the hostel room is not assigned then

students data will be in the USER REGISTRATION table but not in the REGISTRATION.

**Chapter 5**

**IMPLEMENTATION**

5.1 **Table Description**

**1. ADMIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **default** |
| ID | NUMBER(5) | NO | PRIMARY | NULL |
| USER-NAME | VARCHAR(20) | NO |  | NULL |
| EMAIL-ID | VARCHAR(40) | NO |  | NULL |
| PASSWORD | VARCHAR(10) | NO |  | NULL |
| REG-DATE | DATE | NO |  | NULL |
| UPD-DATE | DATE | NO |  | NULL |

2.**ADMIN LOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** |
| ADMIN-ID | NUMBER(5) | NO | PRIMARY | NULL |
| LOGIN-DATE | DATE | NO |  | NULL |

3.**ROOMS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** |
| ROOM\_NO | NUMBER(5) | NO | PRIMARY | NULL |
| SEATTER | NUMBER(5) | NO |  | NULL |
| FEES | NUMBER(7) | NO |  | NULL |
| POSTING-DATE | DATE | NO |  | NULL |

4.**COURSE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** |
| COURSE-ID | NUMBER(7) | NO | PRIMARY | NULL |
| COURSE-NAME | VARCHAR(20) | NO |  | NULL |

**5.STATES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** |
| S-ID | NUMBER(4) | NO | PRIMARY | NULL |
| STATES | VARCHAR(20) | NO |  | NULL |
| CITY | VARCHAR(20) | NO |  | NULL |

**6.USER REGISTRATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** |
| U-ID | NUMBER(5) | NO | PRIMARY | NULL |
| REG-NO | NUMBER(5) | NO | PRIMARY | NULL |
| EMAIL-ID | VARCHAR(20) | NO |  | NULL |
| PASSWORD | VARCHAR(20) | NO |  | NULL |
| FIRSTNAME | VARCHAR(20) | NO |  | NULL |
| MIDDLENAME | VARCHAR(20) | NO |  | NULL |
| LASTNAME | VARCHAR(20) | NO |  | NULL |
| GENDER | VARCHAR(2) | NO |  | NULL |
| PH-NO | NUMBER(10) | NO |  | NULL |
| REG-DATE | DATE | NO |  | NULL |
| UPD-DATE | DATE | NO |  | NULL |

**7.USER LOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** |
| U-ID | NUMBER(5) | NO | PRIMARY | NULL |
| CITY | VARCHAR(20) | NO |  | NULL |
| COUNTRY | VARCHAR(20) | NO |  | NULL |
| LOGIN-DATE | DATE | NO |  | NULL |

**8.REGISTRATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **NULL** | **Key** | **Default** |
| ID | NUMBER(4) | NO | PRIMARY | NULL |
| ROOM-NO | NUMBER(5) | NO |  | NULL |
| SEATER | NUMBER(2) | NO |  | NULL |
| FEESPM | NUMBER(6) | NO |  | NULL |
| FOODSTATUS | NUMBER(1) | NO |  | NULL |
| STAY FROM | DATE | NO |  | NULL |
| DURATION | NUMBER(3) | NO |  | NULL |
| COURSE | VARCHAR(20) | NO |  | NULL |
| REG-NO | NUMBER(10) | NO |  | NULL |
| FNAME | VARCHAR(20) | NO |  | NULL |
| MNAME | VARCHAR(20) | NO |  | NULL |
| LNAME | VARCHAR(20) | NO |  | NULL |
| EMAIL-ID | VARCHAR(30) | NO |  | NULL |
| G-PHNO | NUMBER(20) | NO |  | NULL |
| G-NAME | VARCHAR(20) | NO |  | NULL |
| CITY | VARCHAR(20) | NO |  | NULL |
| STATE | VARCHAR(20) | NO |  | NULL |

**3.2 Stored procedure and views**

**3.2.1 Stored Procedure**

Here we have used a procedure to display the details of the students who have

registered and the respective rooms allocated to them will be displayed when the

admin logins and selects the student info. This is where the procedure is used.

**3.2.2 Triggers**

. CREATE TRIGGER AFTER INSERT ON USER REGISTRATION

BEGIN

insert into phonelog values(New.firstName,New.contactNo);

END

The contact info. And the first name of the user will be updated to the table phonelog

After inserting values on user registration.

**CHAPTER 6**

**RESULTS AND SNAPSHOTS**

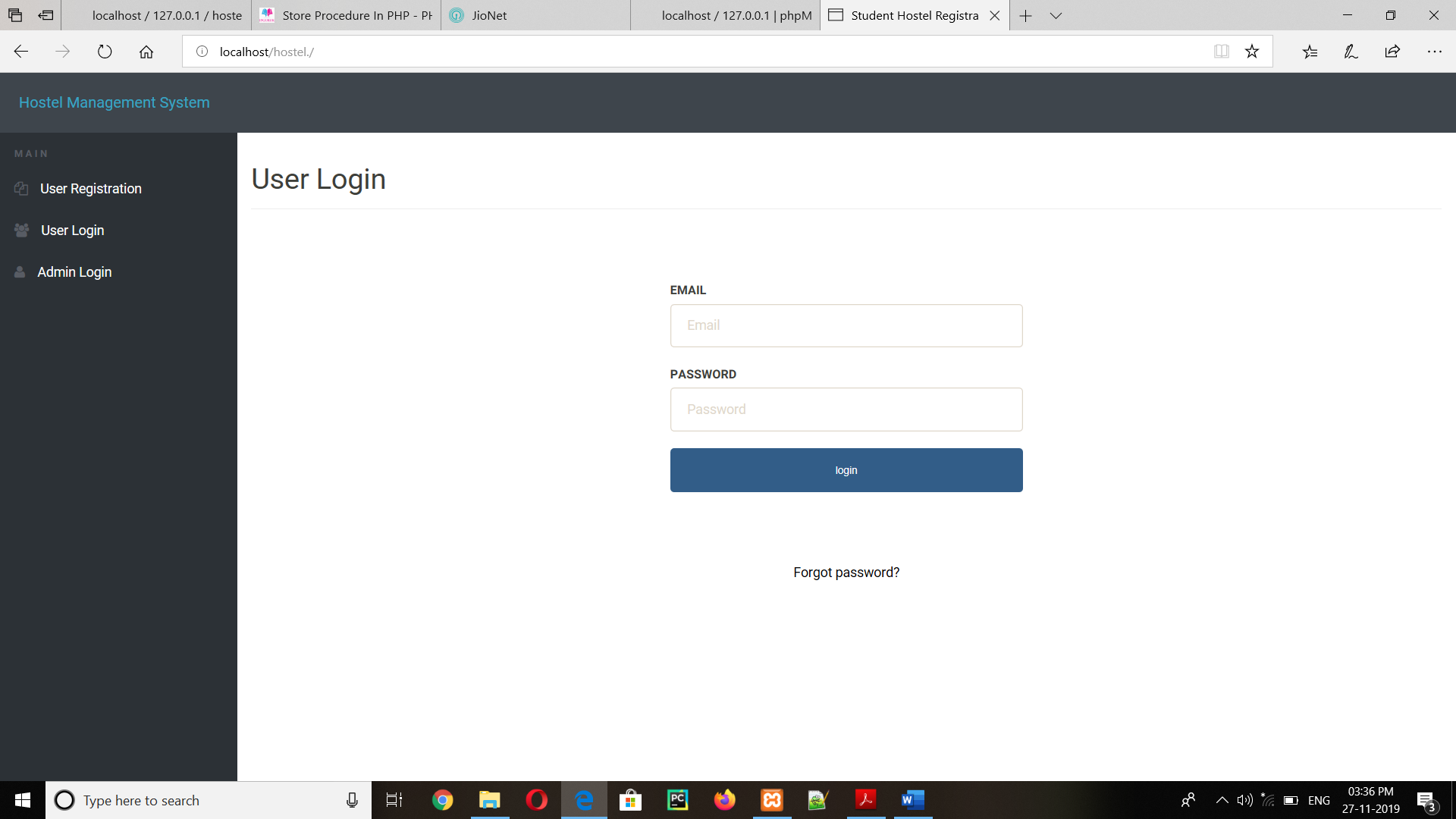


Fig 4.1: The home page of Hostel Management.

The Fig 4.1 is the Home page of the Hostel Management System .Where you can find

three Options:

•User Registration: registration of the new student into the hostel is done in user

Registration option

•User Login: Here the user can login through the email –id and the password

•Admin Log: Here the Admin can login into to his account.

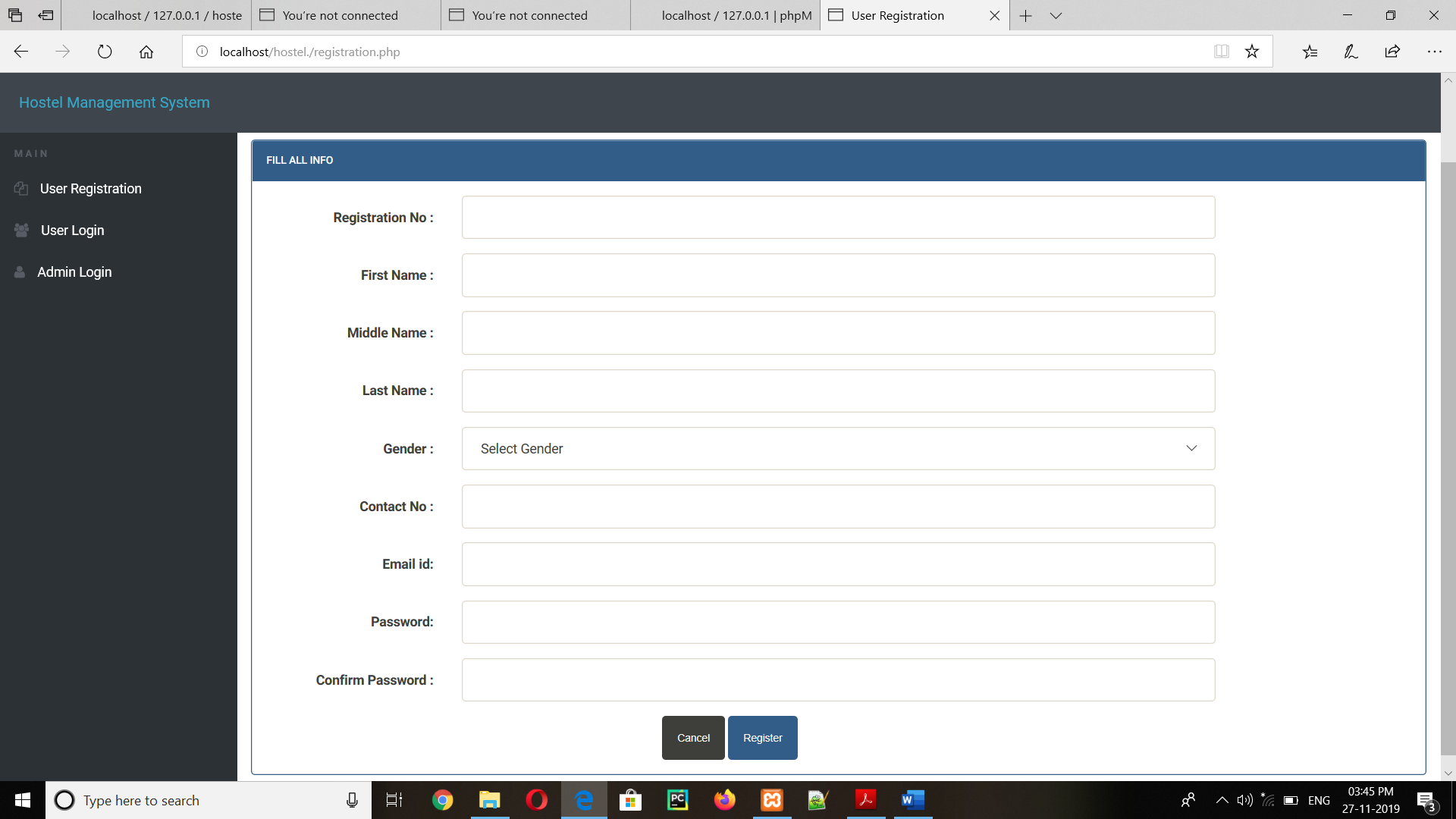


Fig 4.2: The user/student Registration

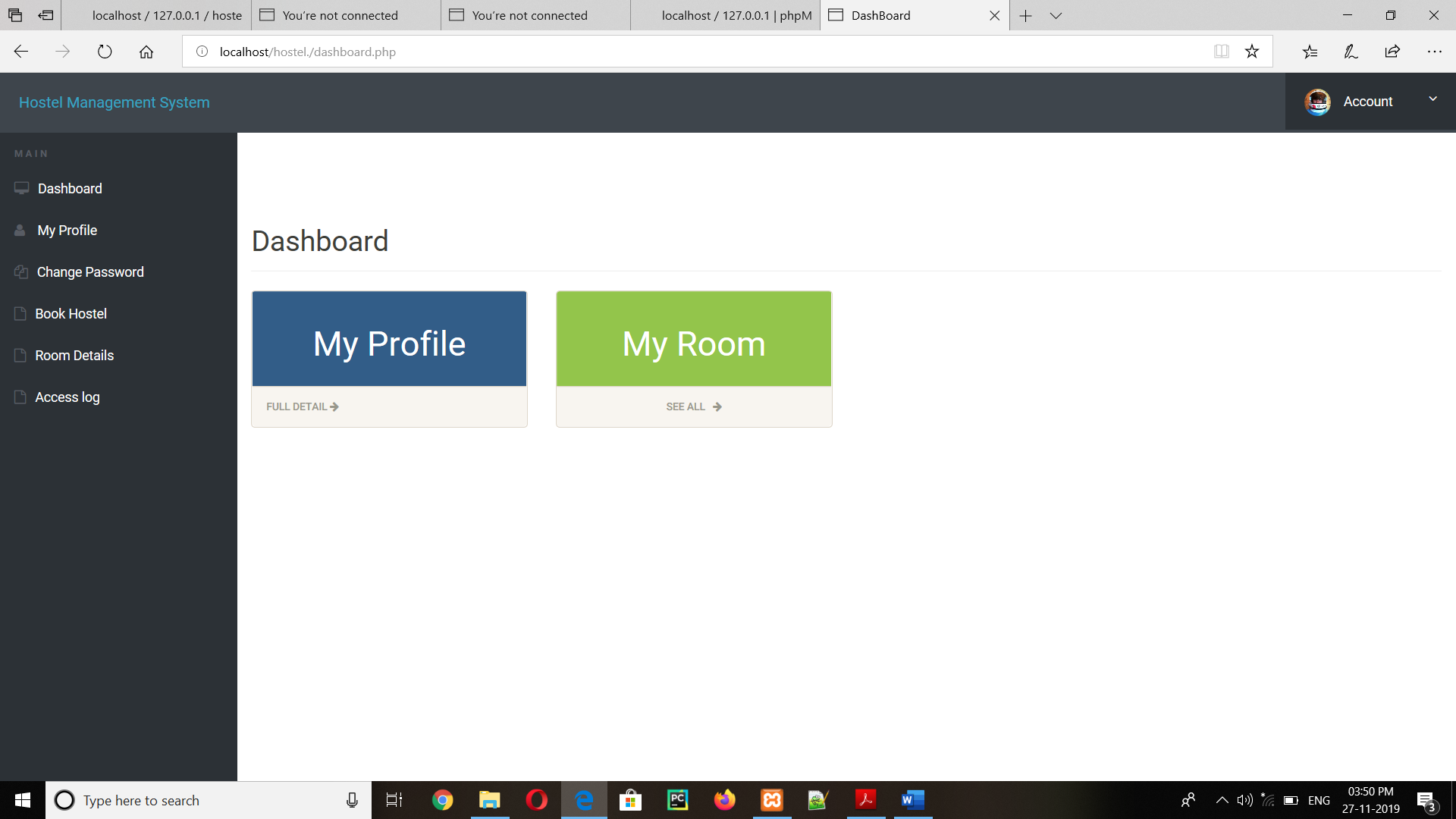


Fig 4.3 Home of the user login

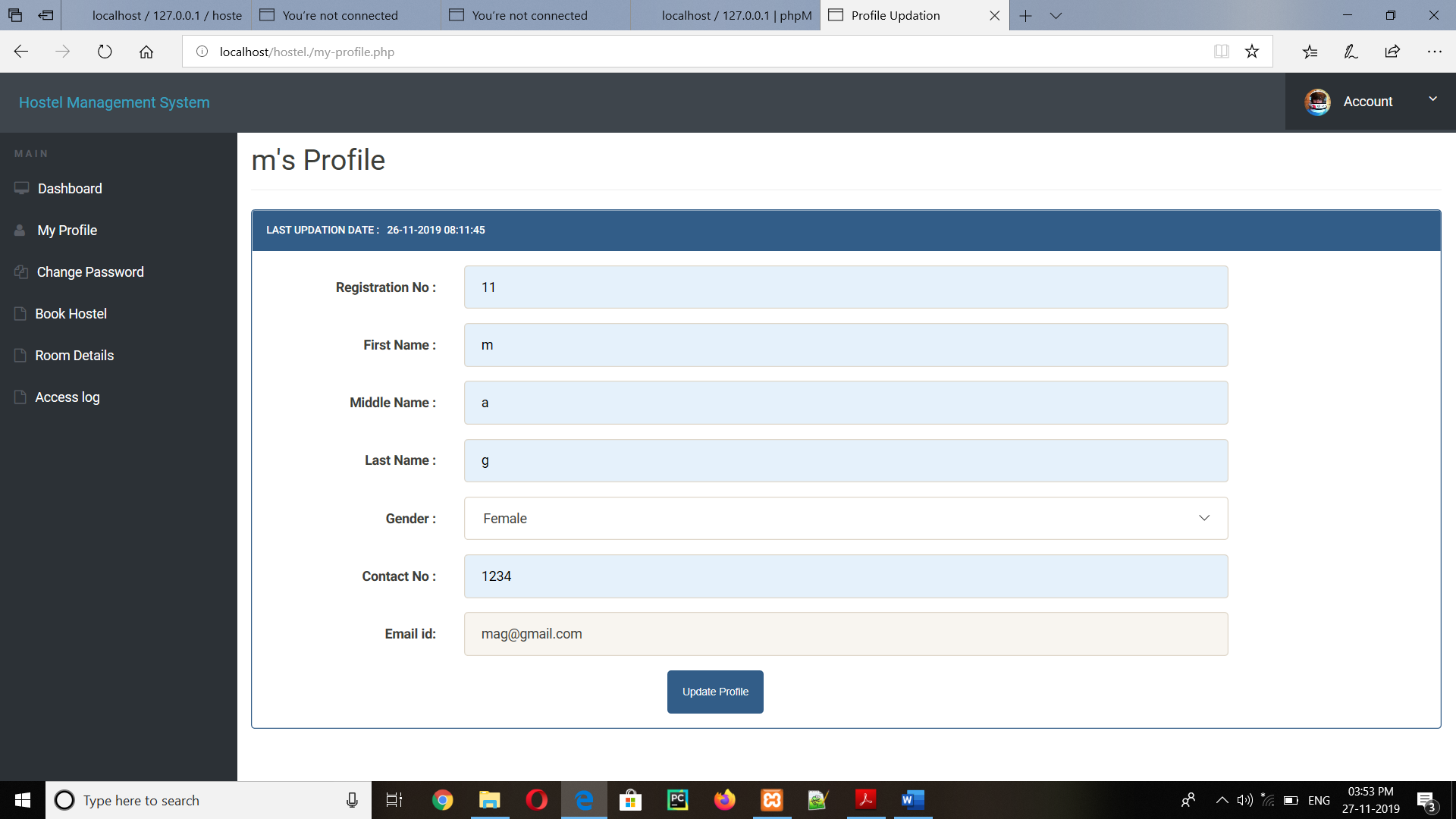


Fig 4.4 Student profile

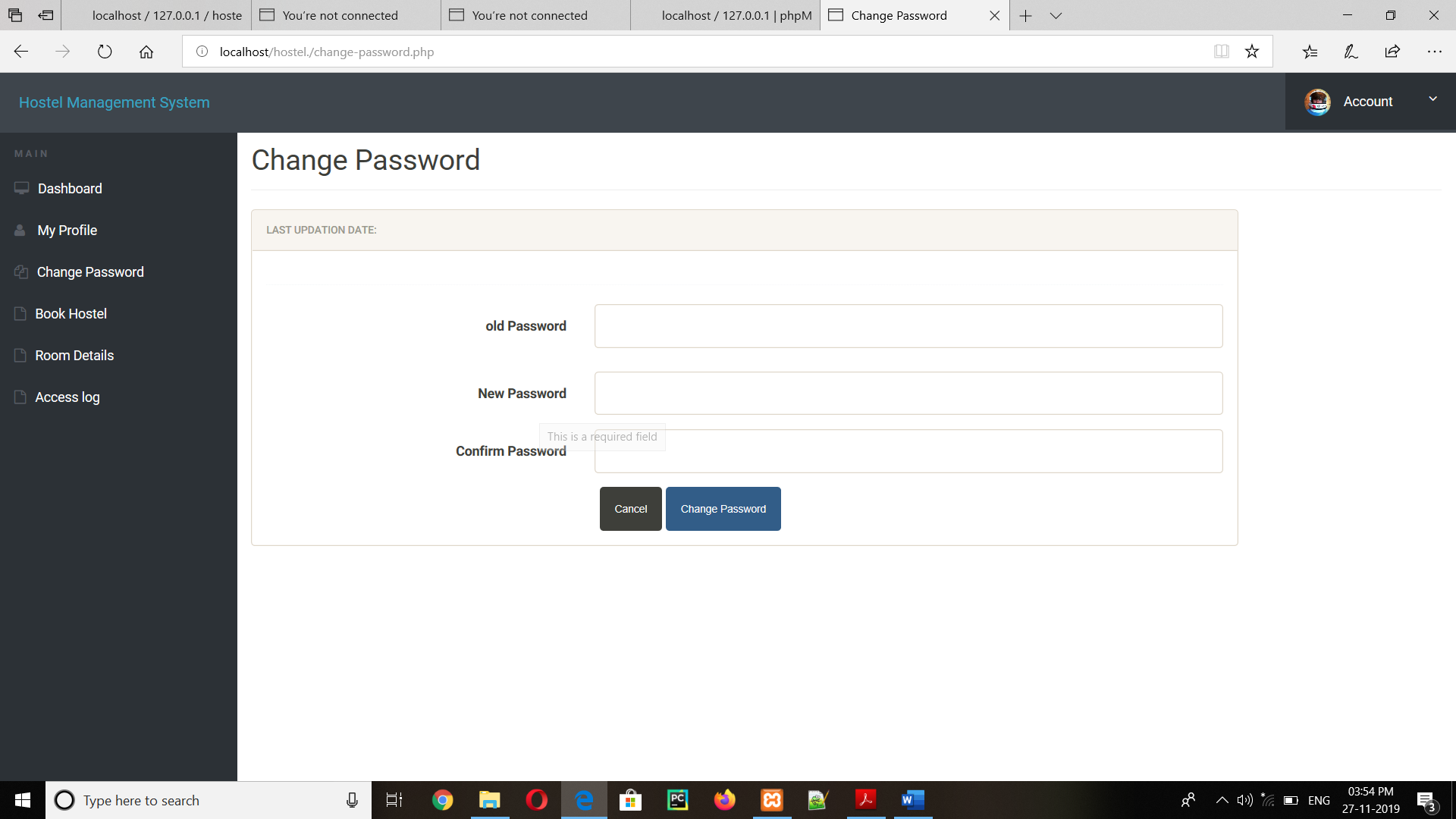


Fig 4.5: Change password window

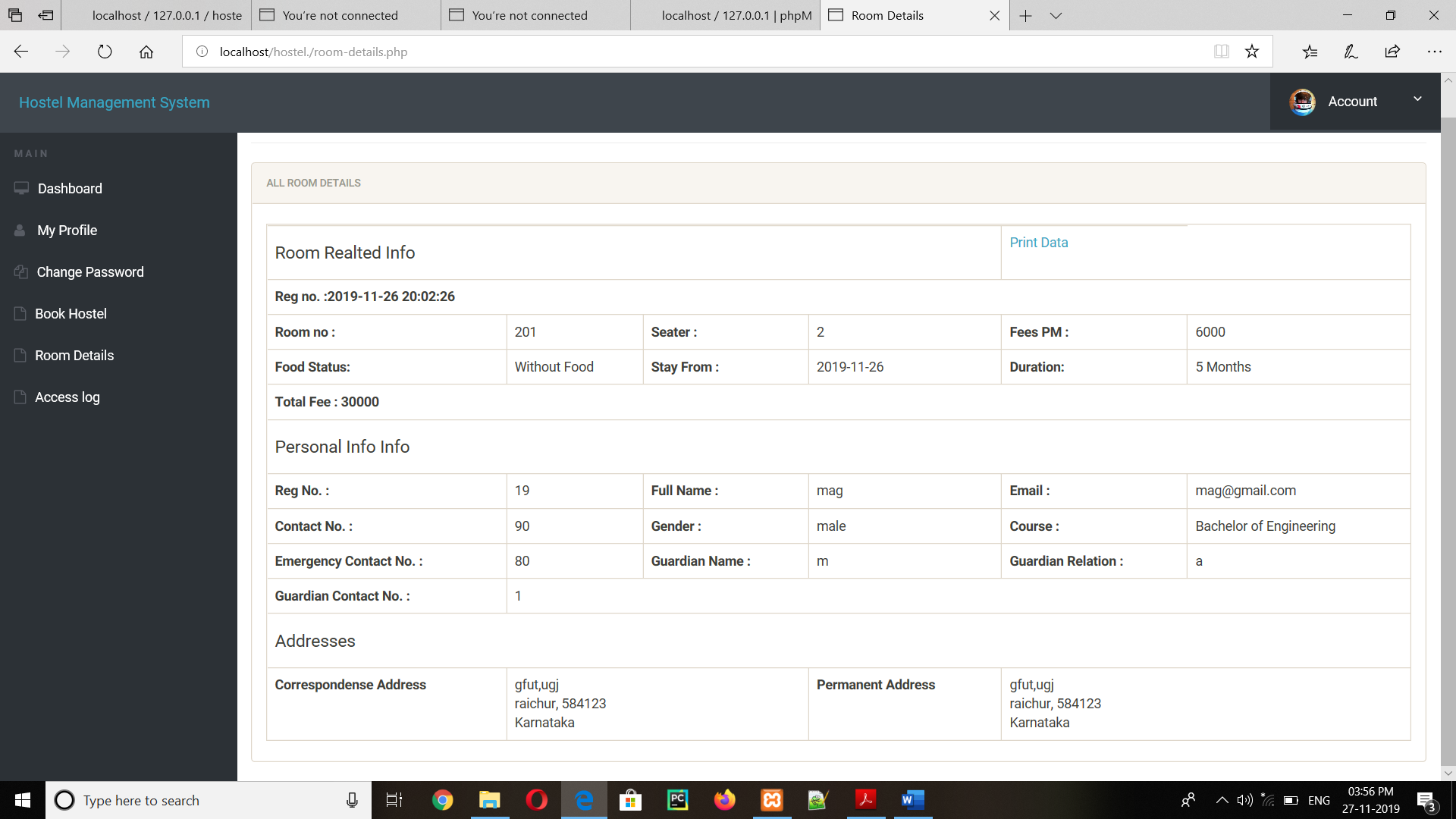


Fig 4.6: Room Details of the Student

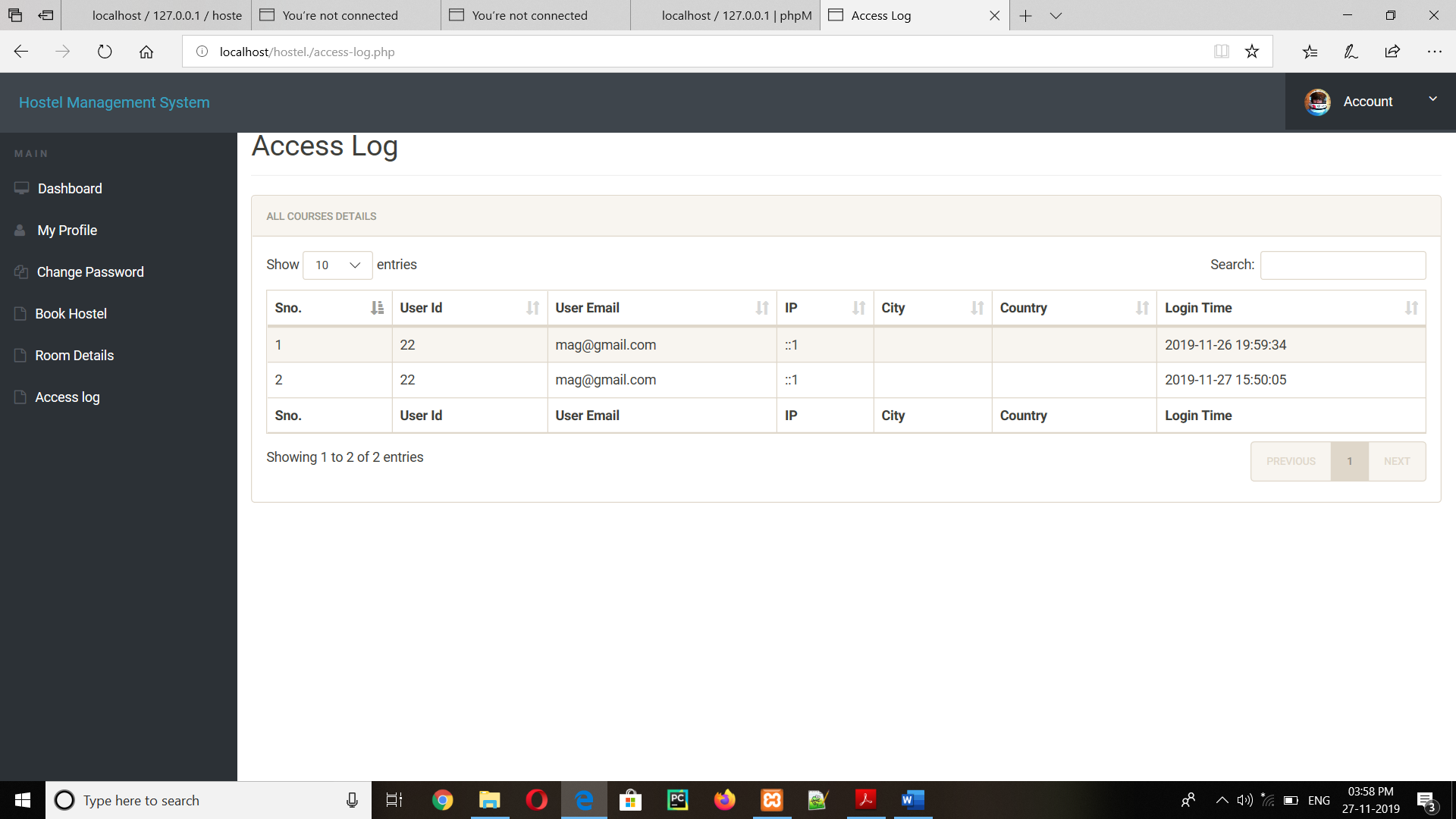


Fig 4.7: Access log of the User/Student

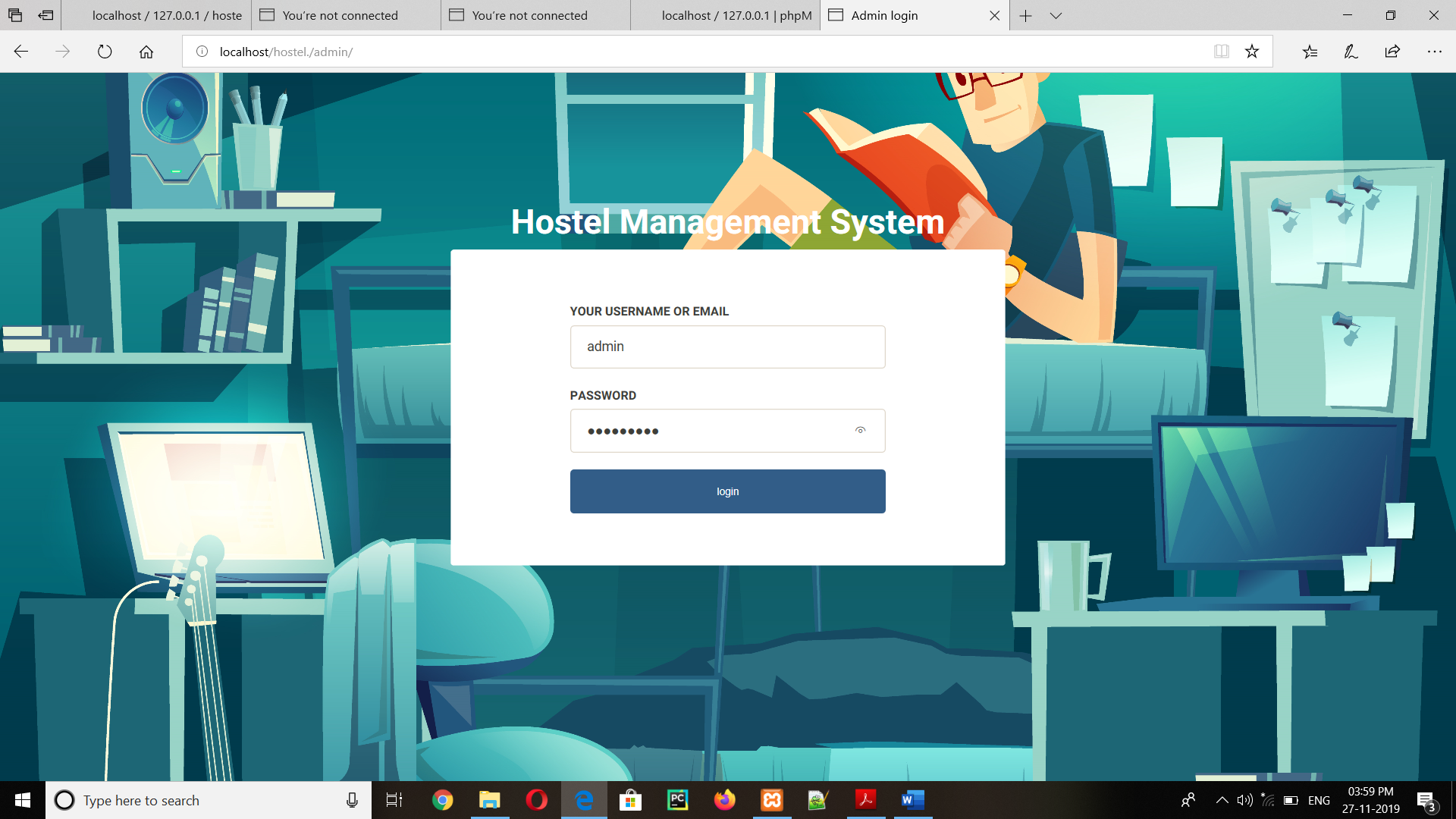


Fig 4.8: Admin Login page

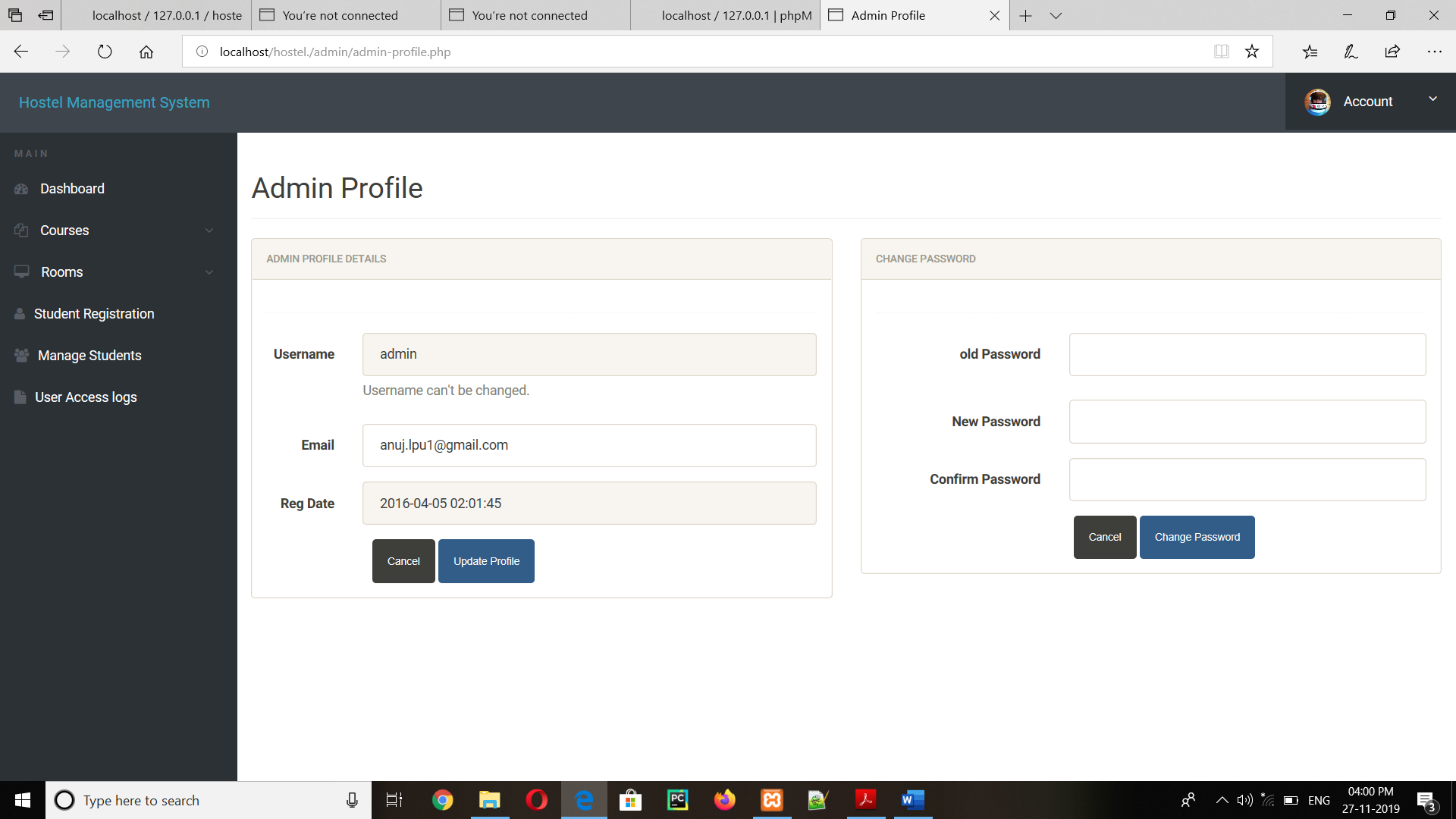


Fig 4.9: Admin Profile

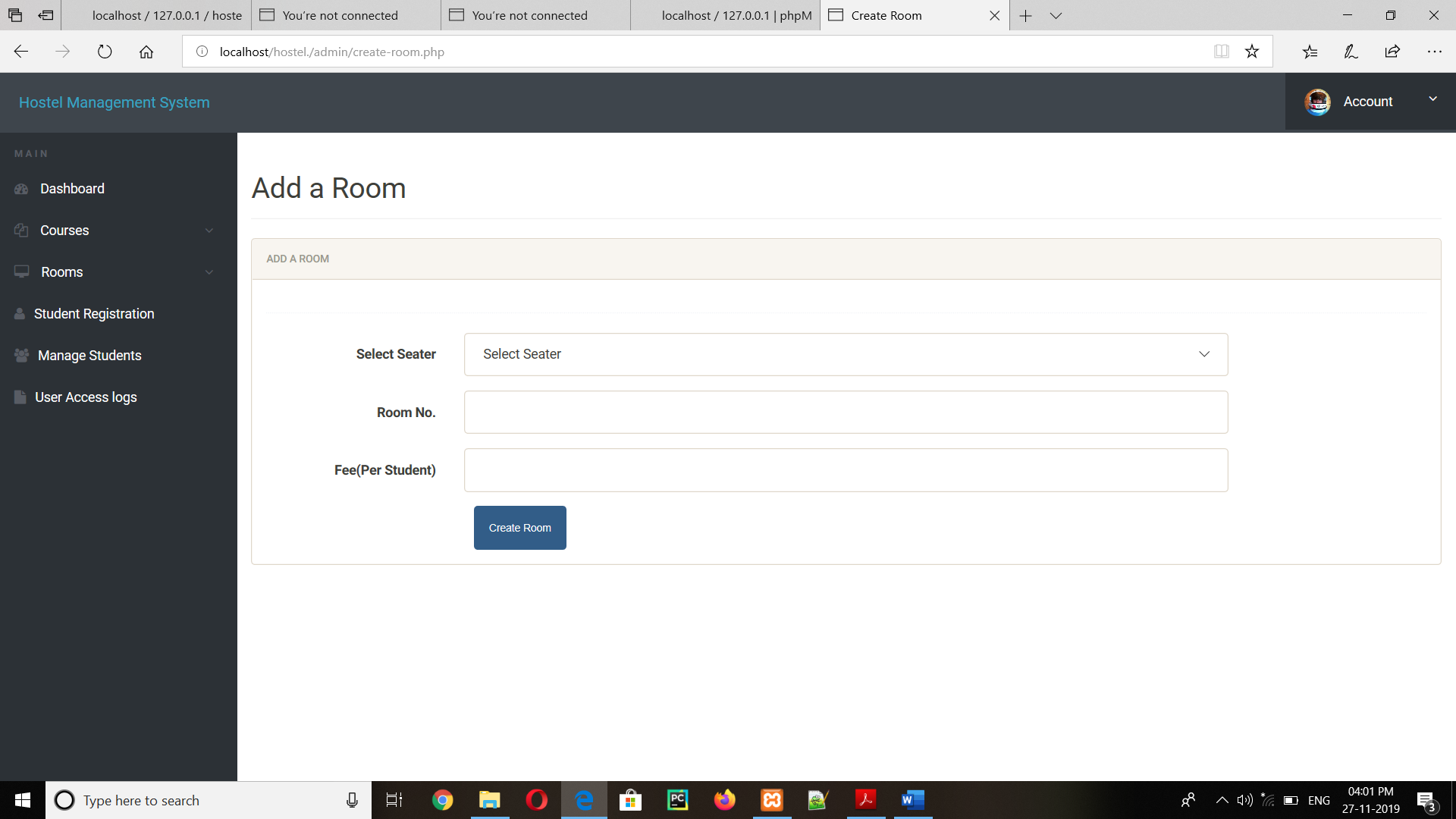


Fig 4.11: Add room window

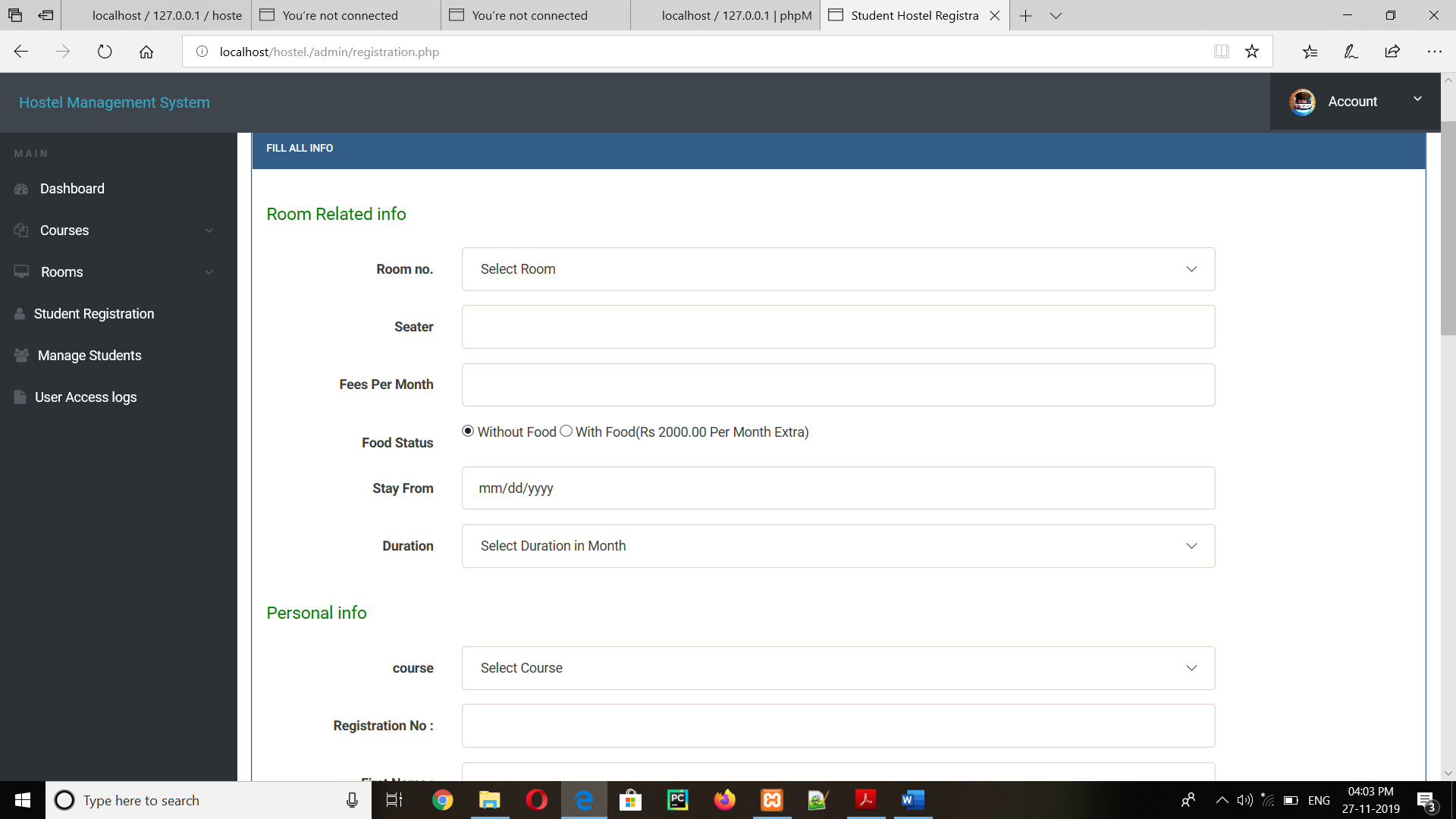


Fig 4.11: Registration form for students by admin

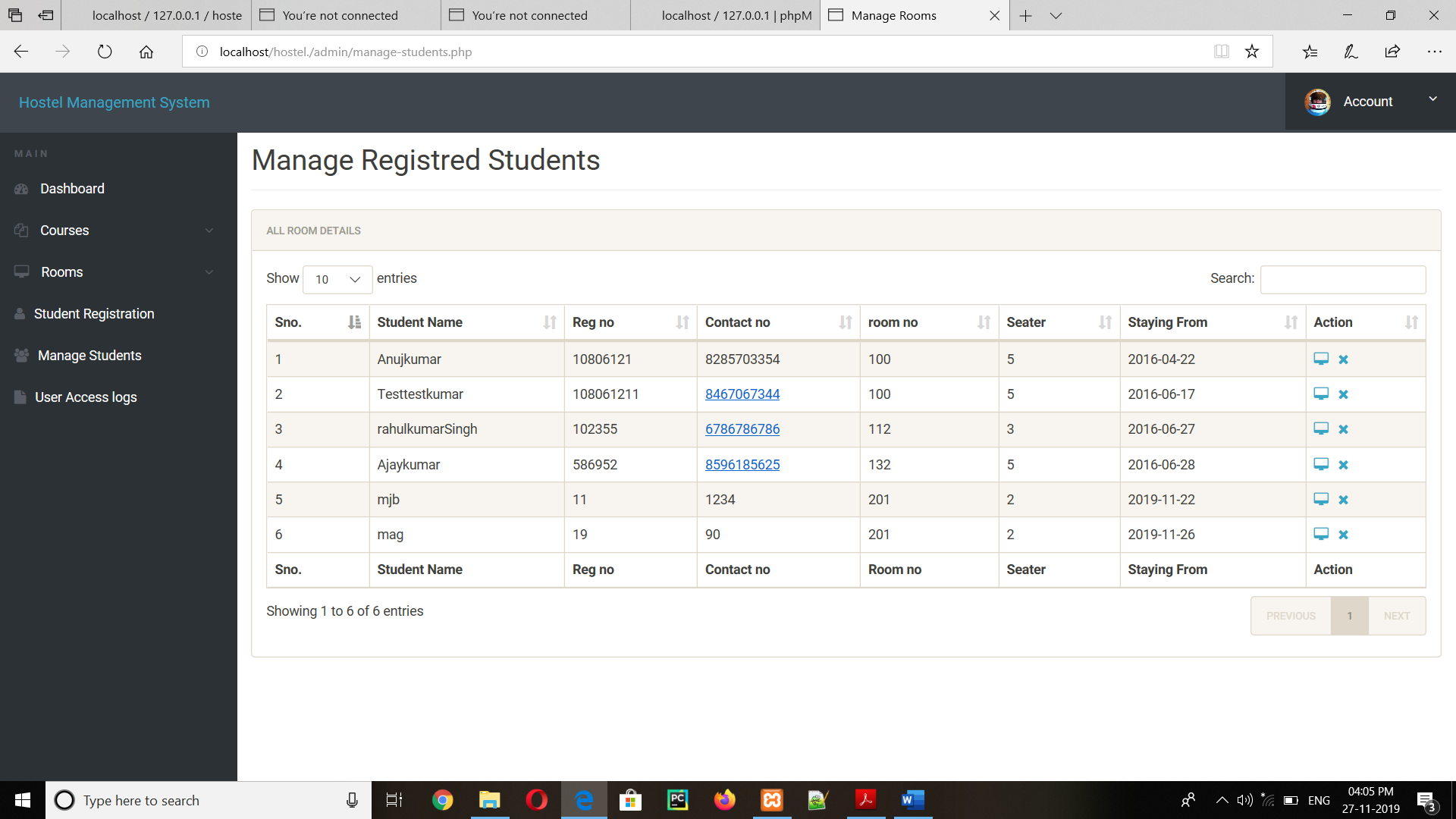


Fig 4.13: Manage Student details

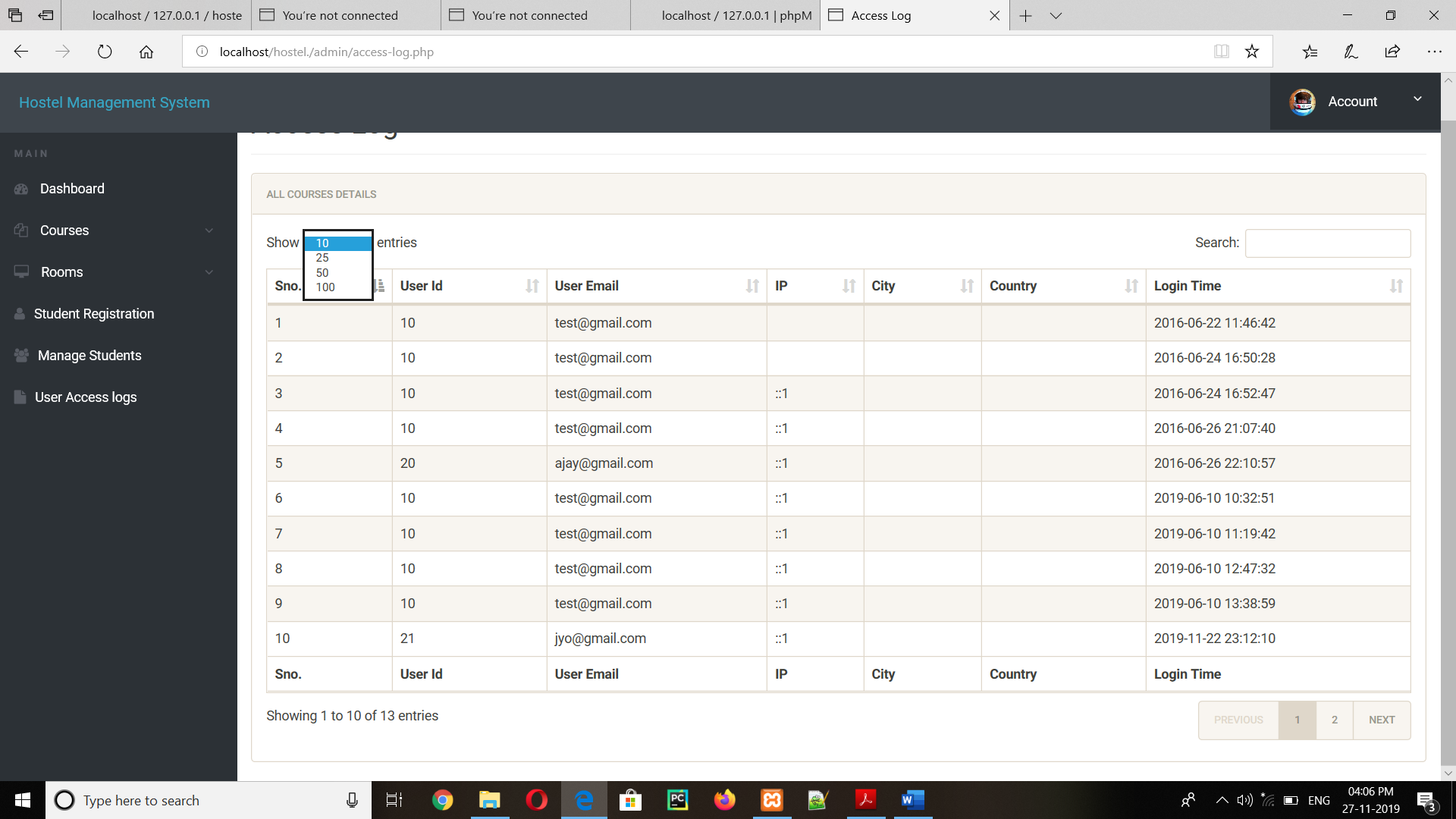


Fig 4.14: The access log of user/student

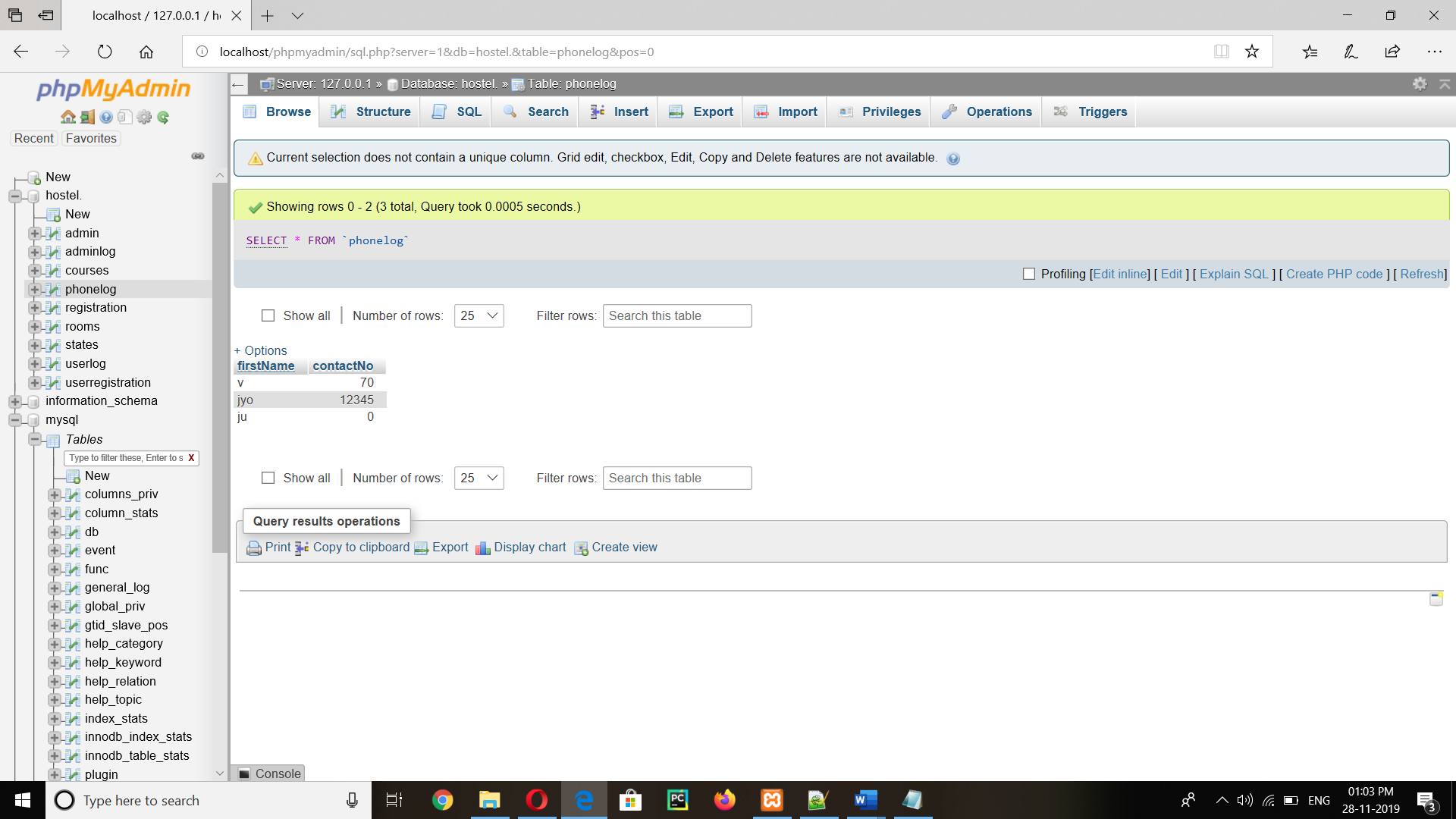


Fig 4.15:Back-end picture of phonelog(trigger)

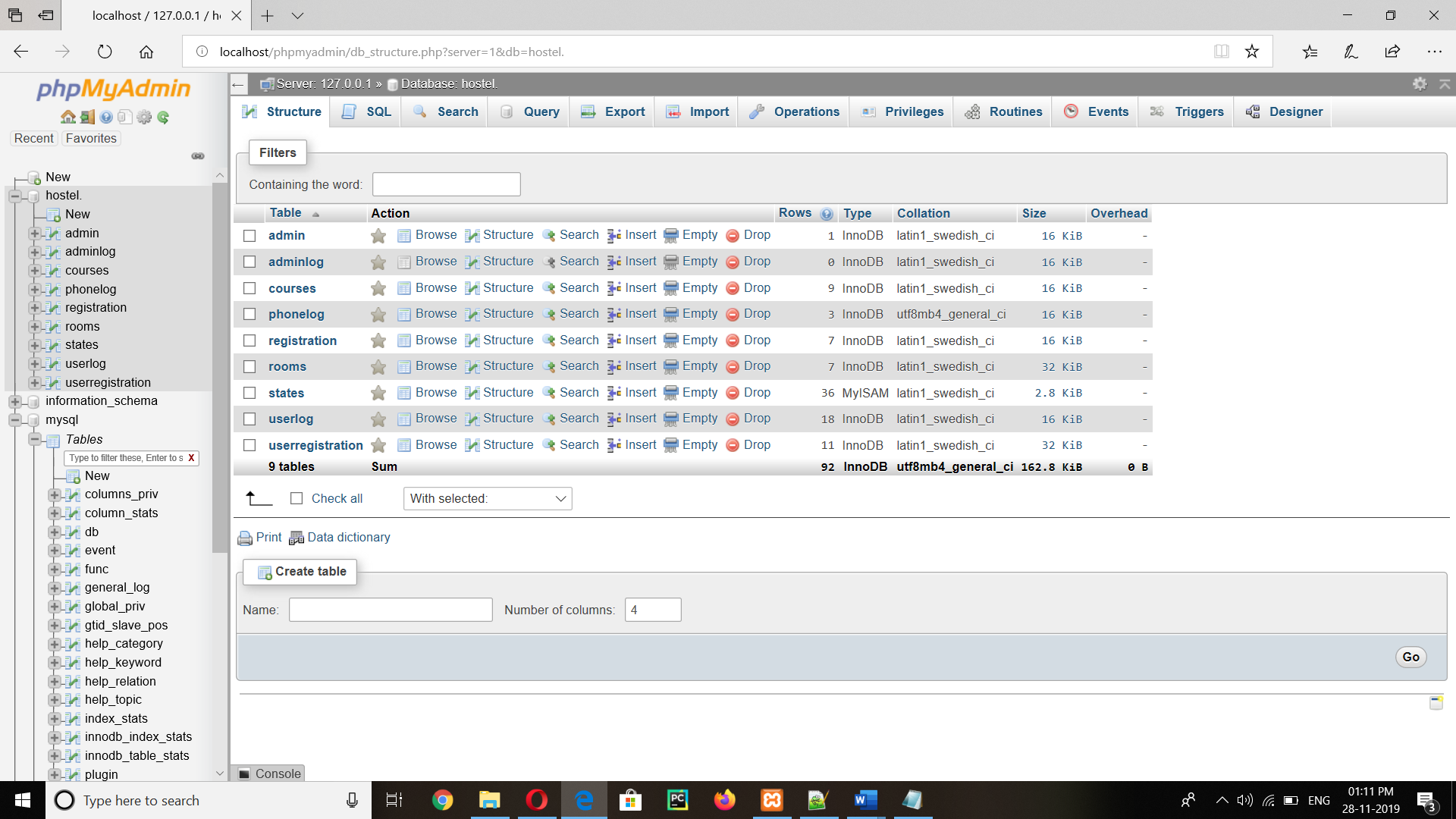


Fig 4.16:Back-end picture of hostel tables

**Chapter 5**

**Conclusion and Future Enhancement**

Hostel Management System is a Customized and User-friendly software for Hostel .It has

been designed to automate, manage and look after the over-all processing of even very

large hostel. It is capable of managing Enquiry details, Student Details, Payment Details

etc.

Hostel Management System is a perfect way to maintain the hostel with the help

of the admin and decrease the time consuming process like standing in the queue for

booking the hostel It also decreases the paper work as the details of the students are

automatically stored into the Database and provides a proper security for the information

of the students. This how the Software Hostel Management System can be useful.

Hostel Management System software is offering a maximum of stability, costeffectiveness

and usability .It provides the most flexible and adaptive standards

management System Software solutions for the hostel. It is easy to extend the system that

we have proposed. A person could see any of the issued Unissued or all the rooms

accordingly to his/her will. In future we can implement some features for “HOSTEL

***MANAGEMENT*** SYSTEM” project.

**References**

[1] www.w3schools.com

[2] www.monstercrawler.com

[3] www.stackoverflow.com

[4] www.info.com

[5] www.phpbuddy.com

[6] www.google.com

[7] www.mysql.com