A PROJECT REPORT ON

“Student Management System”

Submitted By-

Shivanjali Chaurasia

(Web Development and Design Intern)

UNDER THE GUIDANCE OF

Er. Shivali Srivastava

(Software Engineer)

Triffort Technologies Private Limited

45A/1A/4 First Floor, Dayanand Marg (Thornhill Road),

Allahabad, Uttar Pradesh, INDIA

ACKOWLEGEMNENT

It is great pleasure to present this report on the project undertaken by me as part of our Web Development and Design Trainee.

I am thankful to our company **Triffort Technologies and Private Limited** for offering us such a wonderful challenging opportunity and I express my deepest thanks to all coordinators, for providing all the possible help and assistance and their constant encouragement.

It is a pleasure that we find ourselves penning down these lines to express our sincere thanks to the people who helped me along the way in completing our project. I find inadequate words to express our sincere gratitude towards them.

I would like to express our gratitude towards **Er. Shivali Srivastava** for placing complete faith and confidence in our ability to carry out this project and for providing us his time, inspiration, encouragement, help, valuable guidance, constructive criticism and constant interest.

TABLE OF CONTENTS

Content

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Objective of Project:

This project **"Student Management System"** provides us a simple interface for maintainance of student information.It can be used by educational institutes or colleges to maintain the records of 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. Throughout the project the focus has been on presenting information in an easy and intelligible manner. The project is very useful for those who want to know about Student Management Systems and want to develop various software or websites based on the same concept. The project provides facilities like online registration and profile creation of students, mentors, as well as university, thus reducing paperwork and automating the record generation process in an educational institution.

1.2 Need of the Project:

The project Student Management System has been developed to help the department maintaining the student details, mentor details, and college details. Earlier, the records where maintained manually, with the help of the package the concerned departments will be able to improve the productivity, reduce the time, cost factors associated with the system. The automation of the system will help the organization in proper maintenance of the record, less manpower, less man-days, less cost, proper & accurate functioning.

The basic need for the package was to automate the whole procedure of maintaining of student details, earlier it was all done manually. By developing this package lot of burden was removed from the department, which was maintaining students, mentors and university details. It improved the efficiency, reduced the cost, and reduced the time need to do the work manually. With the help

of this package the past details of the student and mentors can be assessed and reports can be generated on this details.

In brief, we can say this system was required to automate the processing of student details, which was done manually before the development of the package. Earlier all the information / data pertaining to the students was maintained manually or we can say it was on paper, hence it created a problem for the organization, how to manage it properly. With the help of this system the organization us able to maintain the data properly and accurately.

Why System was build?

- Earlier, data pertaining to students was maintained manually.
- Manual system was not efficient.
- Cost of maintaining data manually was bigger or huge.
- Large manpower was required.
- The procedure was error prone, it was not accurate.
- Manual system was not suited for electronic exchange of data.

Solution?

The solution for all this problem was to automate the system, automation of the students, mentors and university data maintenance would reduce the man power, man days will result in accurate data and above all increases the efficiency of the concerned department.

1.3 Features of the Project

- Role based login system
- Authenticate Mechanism
- Create, Read, Update and Delete Mechanism for all the entities
- Role Based Access Level
- Reports Management by Exporting Data in Excel Sheet
- Multiple Deletion of Data
- Profile Picture Upload Feature
- User Friendly and Interactive
- Minimum time required.
- Minimize manual data entry
- Security of data
- Proper control of higher officials
- Minimum time needed for the various processing
- Ensure data accuracy's.
- Better service

CHAPTER 2

FACILITIES REQUIRED

2.1 Hardware Requirements

Processor	:	Intel Pentium IV 2.0 GHz and above
RAM	:	512 MB and above
Hard disk	:	80GB and above
Monitor	:	CRT or LCD monitor
Keyboard	:	Normal or Multimedia
Mouse	:	Compatible mouse

2.2 Software Requirements

Client Side	:	XAMP
Language	:	PHP
Back End	:	PHP MyAdmin
Operation System	:	Windows XP or above
Browser	:	Any latest browser

CHAPTER 3

TECHNOLOGY OVERVIEW

3.1 Client Side:

- **HTML:** It is used to generate web page. HTML, an initial of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on.

- **CSS: C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

- **JAVASCRIPT:** JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. It is used for checking User information before sending to JavaScript is a scripting language most often used for client-side web development. It is dynamic, weakly typed, prototype-based language with first-class functions. Currently, "JavaScript" is an implementation of ECMA Script Standard.

- **JQUERY**- jQuery is a lightweight, "write less, do more", and JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. JQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

3.2 Server Side:

- **PHP-** PHP is a technology that lets you mix regular, static HTML with dynamically-generated HTML. Many Web pages that are built by CGI programs are mostly static, with the dynamic part limited to a few small locations. But most CGI variations, including servlets, make you generate the entire page via your program, even though most of it is always the same. PHP is a scripting language designed to fill the gap between SSI (Server Side Includes) and Perl, intended for the web environment. Its principal application is the implementation of web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the Open Source software movement. Its popularity derives from its C-like syntax, and its simplicity. PHP is currently divided into two major versions: PHP 4 and PHP 5, although PHP 4 is deprecated and is no longer developed or supplied with critical bug fixes. PHP 6 is currently under development. PHP was designed by Rasmus Lerdorf to display his resume online and to collect data from his visitors. PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "PHP: Hypertext Preprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications which can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL.

- **IIS SERVER /XAMP-** is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible. XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer, with the advantage a number of common add-in applications such as Wordpress and Joomla can also be installed with similar ease using Bitnami.

3.3 Backend Tool:

- **phpMyAdmin:** PHPMyAdmin is a free software tool written in PHP, intended to handle the Administration of MySQL over the Web. PHPMyAdmin supports a wide range of operations on MySQL, Maria DB and Drizzle. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc.) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

  PhpMyAdmin comes with a wide range of documentation and users are welcome to update our wiki pages to share ideas and how-to for various operations. The phpMyAdmin team will try to help you if you face any problem; you can use a variety of support channels to get help. PhpMyAdmin is also very deeply documented in a book written by one of the developers – Mastering phpMyAdmin for Effective MySQL Management, which is available in English and Spanish.

  The phpMyAdmin project is a member of Software Freedom Conservancy. SFC is a not-for-profit organization that helps promote, improve, develop, and defend Free, Libre, and Open Source Software (FLOSS) projects.

3.4 Framework:

- **CodeIgniter:** CodeIgniter is an open-source software rapid development web framework, for use in building dynamic web sites with PHP. CodeIgniter is loosely based on the popular model–view–controller (MVC) development pattern. While controller classes are a necessary part of development under CodeIgniter, models and views are optional. Codeigniter can be also modified to use Hierarchical Model View Controller (HMVC) which allows developers to maintain modular grouping of Controller, Models and View arranged in a sub-directory format. CodeIgniter is most often noted for its speed when compared to other PHP frameworks. In a critical take on PHP frameworks in general, PHP creator Rasmus Lerdorf spoke at frOSCon in August 2008, noting that he liked CodeIgniter "*because it is faster, lighter and the least like a framework.*"

CHAPTER 4

WORKING OF PROJECT

Student Information Management System can be used by education institutes to maintain the records of students, mentors and university 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.

1. **Registration Page:** The Registration Page consists of Fields such as First Name, Last Name, Email, Email Confirmation, Password, Password Confirmation, and User Type, which can be of three types: a) Student, b) Mentor, c) University.

   Once a user has registered with one email, he/she won't be able to register again with the same email. If any of the field doesn't matches or is empty, the application won't accept it and an error message will be displayed on site. Each and every field is required to complete the registration.

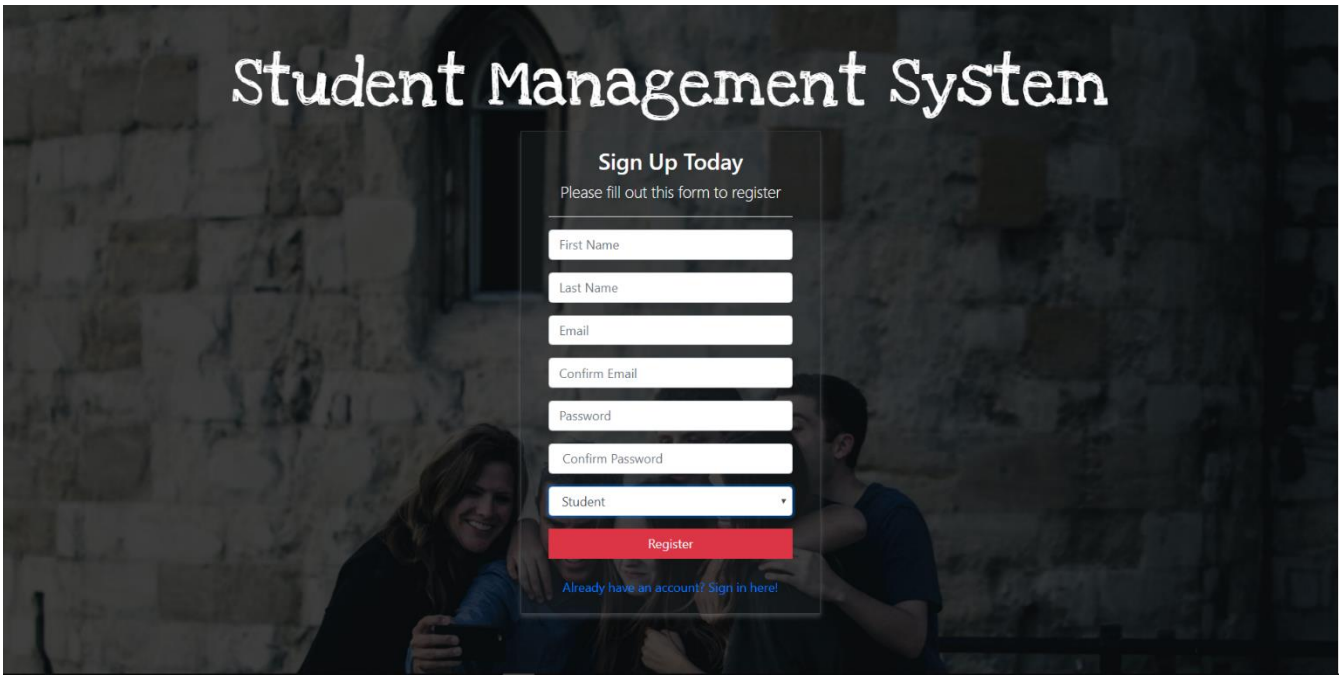


Fig 4.1: Registration Page

2. **Login Page:** With the help of the Login Page, User can log in to the system only if the user is already registered in the system, and if the email and the password given, matches to the existing database. And we all know, that can happen only when user is already registered. Else, it will display error messages.

   It also display an option to register, if you are not registered into the system, which will redirect us to the registration page.

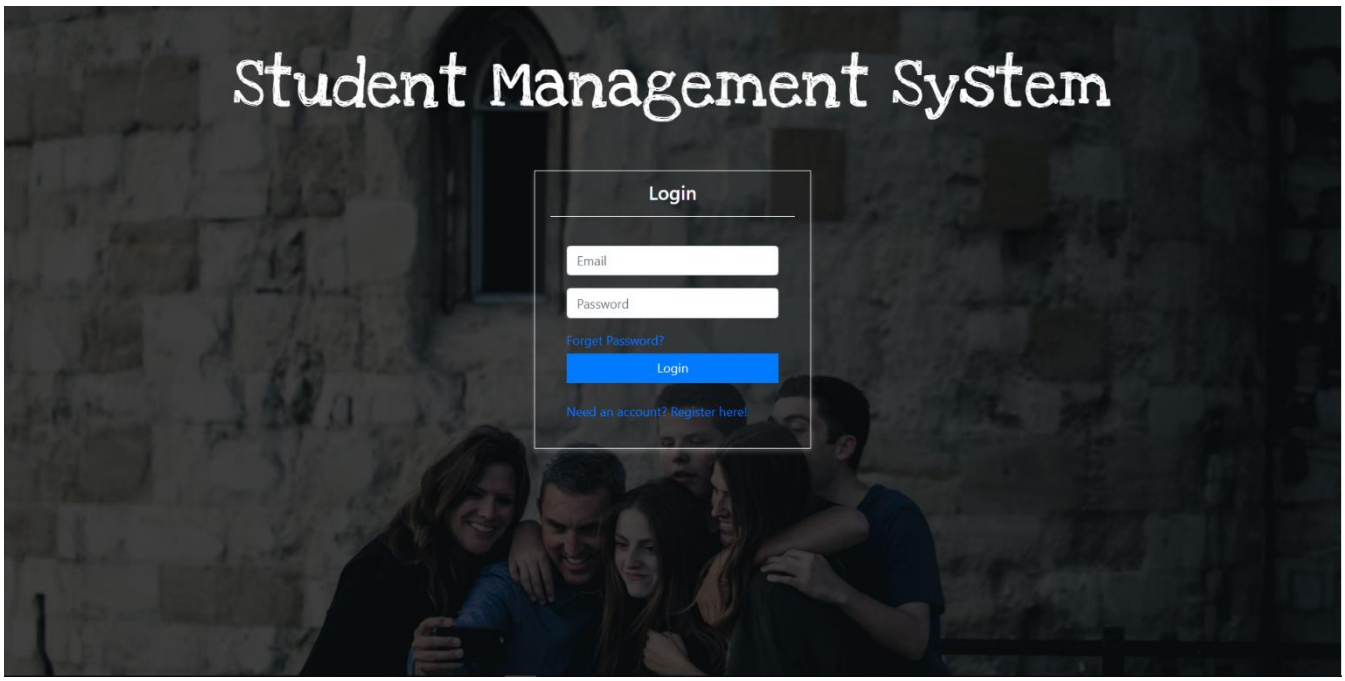


Fig 4.2: Login Page

3. **Student Dashboard (Submit):** As soon as you register, and select the user type as student, it will redirect you to the student dashboard page. If the user is visiting the page first time, and haven't filled any other details, then it will only display First Name, Last Name, and Email column filled, and will be disabled, so that user won't be able to edit it again.

   After entering dashboard, user will be asked few more details to fill to complete registration. User will be required to enter gender, Address that contains city/town, State/province/region, postal code and country. Country column comes as a list of all the available countries in the world, and the user will be needed to select one from that.

   Since, the user type is student, hence user needs to enter his/her field of study, year of passing, marks obtained, and the college in which he/she is conducting his course.

   It also consists of one checkbox that is proof of declaration that all the information provided above is true.

   If the user is entering his/her data for the first time, the update button will be disable for them, and they can only be submitting their data to the organization.

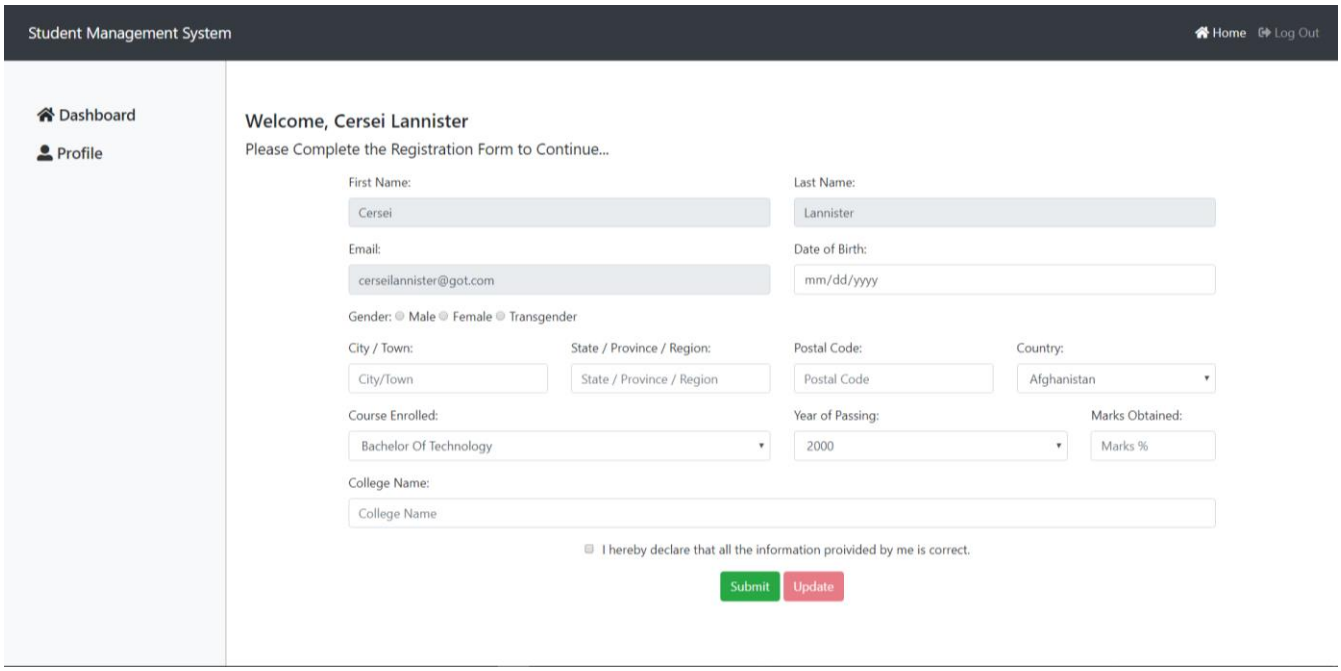


Fig 4.3: Student Dashboard (Submit)

4. **Student Dashboard (Update):** Once the student has submitted the data, then for the next time, he/she will not be able to update his/her date of birth, course, year of passing, and marks obtained. All these columns will be disabled, and user will only be able to update his address and college name.

   Also, the submit button will be disabled, and the user will only be able to update his/her data.

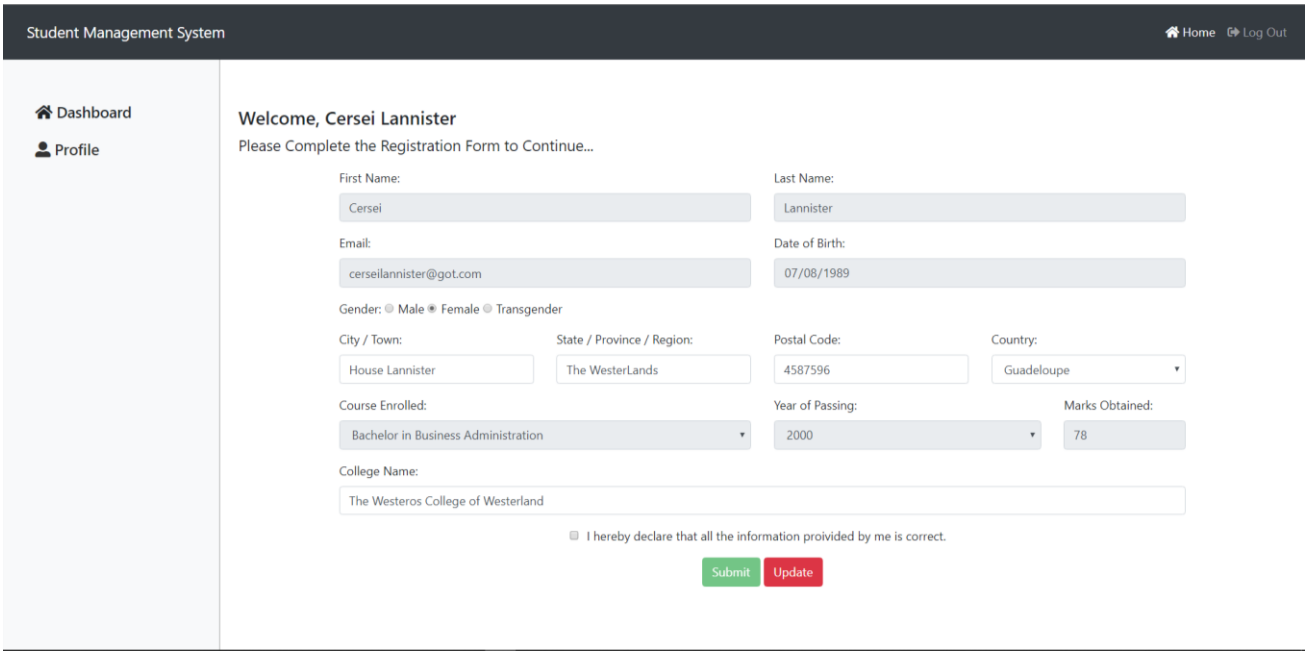


Fig 4.4 Student Dashboard (Update)

5. **Student Profile (Default):** On the student profile page, student can view his/her complete details along with an option of a profile picture, which by-default will be nothing. Student profile also contains three buttons in them. First one is of upload image, by which the image can be updated, second is to update details, which will redirect us to the student dashboard, and the third option is to delete profile, through which the profile can be deleted.

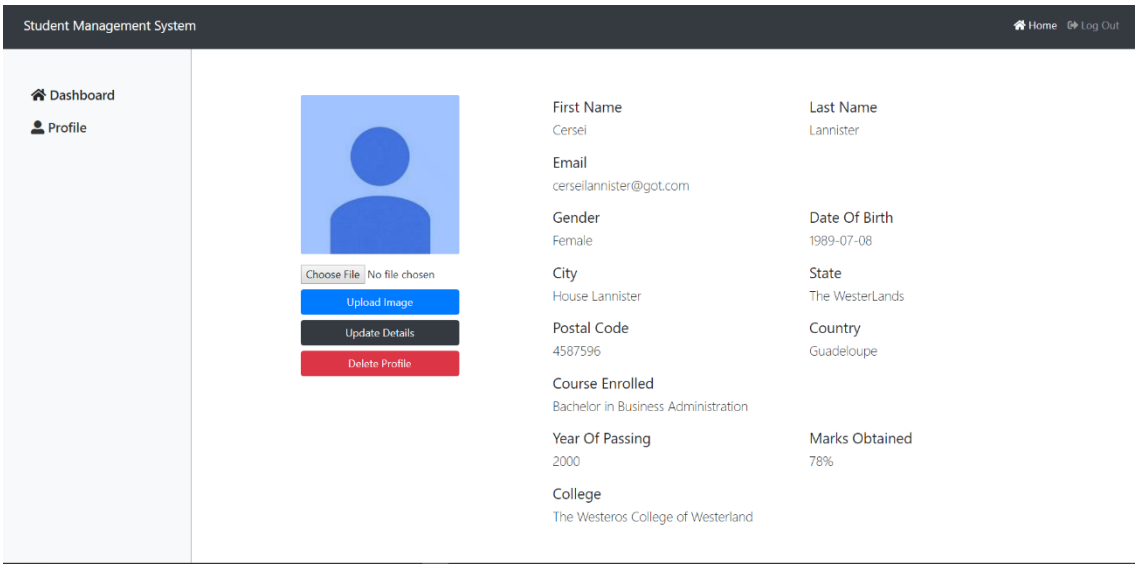


Fig 4.5 Student Profile (Default)

6. **Student Profile (Update):** After updating the profile picture, the new pic will be displayed.

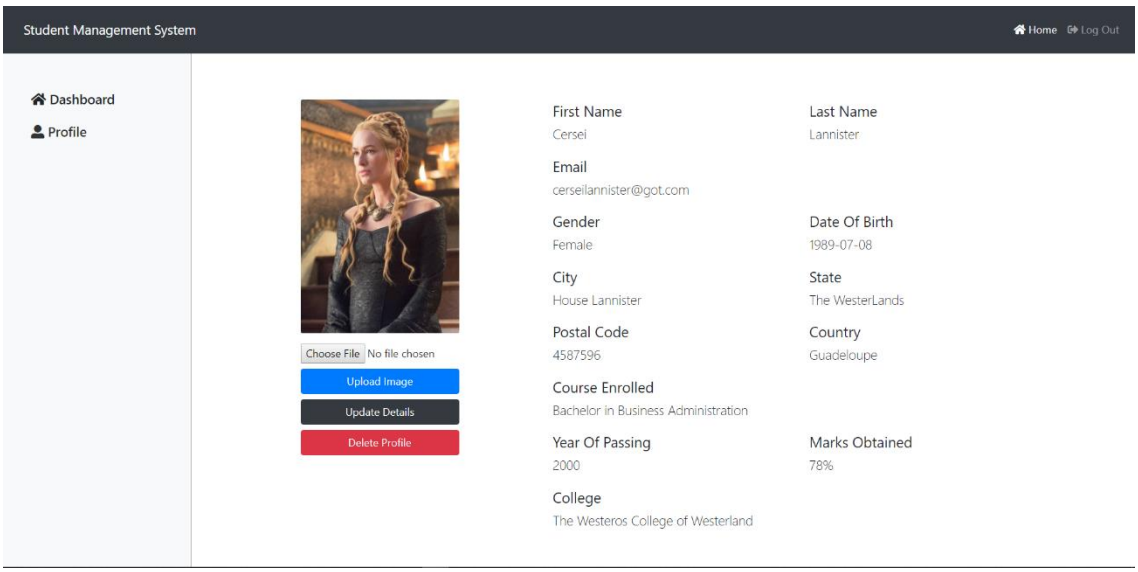


Fig 4.6 Student Profile (Update)

7. **Mentor Dashboard:** As soon as you register, and select the user type as mentor, it will redirect you to the mentor dashboard page. If the user is visiting the page first time, and haven't filled any other details, then it will only display First Name, Last Name, and Email column filled, and will be disabled, so that user won't be able to edit it again.

   If the user is already registered and logging in, then he won't be able to update his designation as well as date of birth too.

   After entering dashboard, user will be asked few more details to fill to complete registration. User will be required to enter gender, Address that contains city/town, State/province/region, postal code and country. Country column comes as a list of all the available countries in the world, and the user will be needed to select one from that.

   Since, the user type is mentor, hence user needs to enter his/her designation too.

   It also consists of one checkbox that is proof of declaration that all the information provided above is true.

   If the user is entering his/her data for the first time, the update button will be disable for them, and they can only be submitting their data to the organization.

   On sidebar, in addition to student, mentor has also an option to view the list of students.

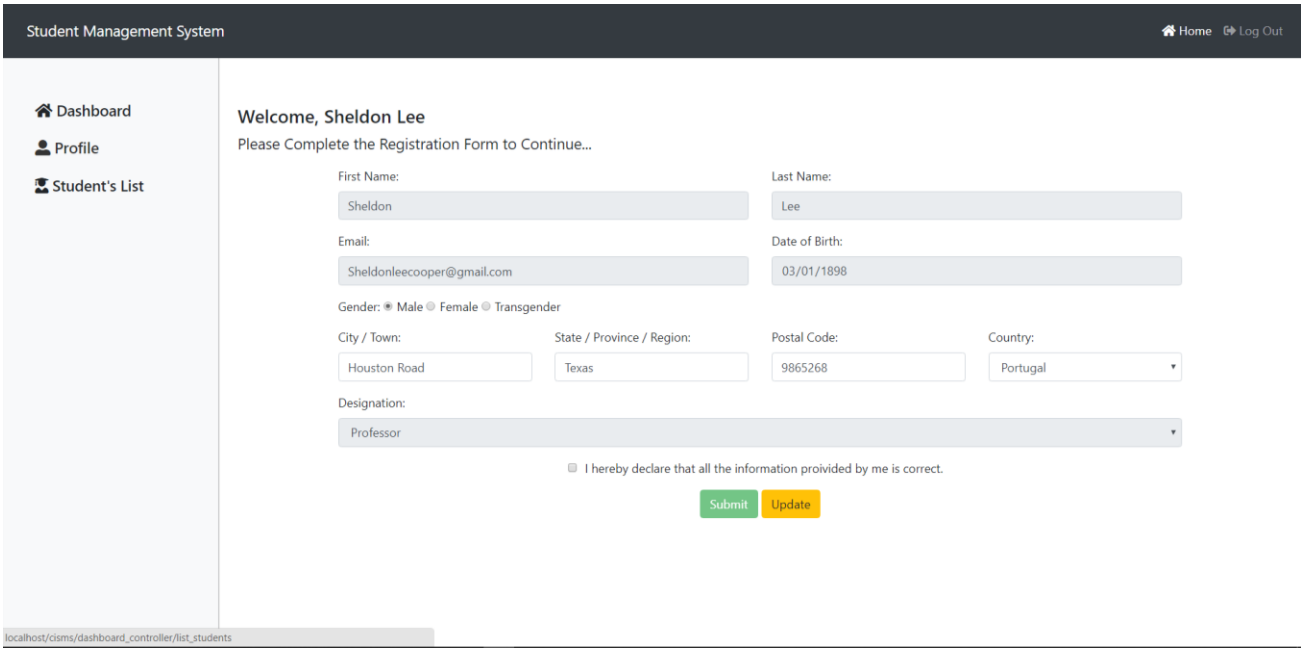


Fig 4.7 Mentor Dashboard

8. **Mentor Profile:** On the mentor profile page, mentor can view his/her complete details along with an option of a profile picture, which by-default will be nothing, but if the user has updated it then it will show his latest profile picture. Mentor profile also contains three buttons in them. First one is of upload image, by which the image can be updated, second is to update details, which will redirect us to the mentor dashboard, and the third option is to delete profile, through which the profile can be deleted.

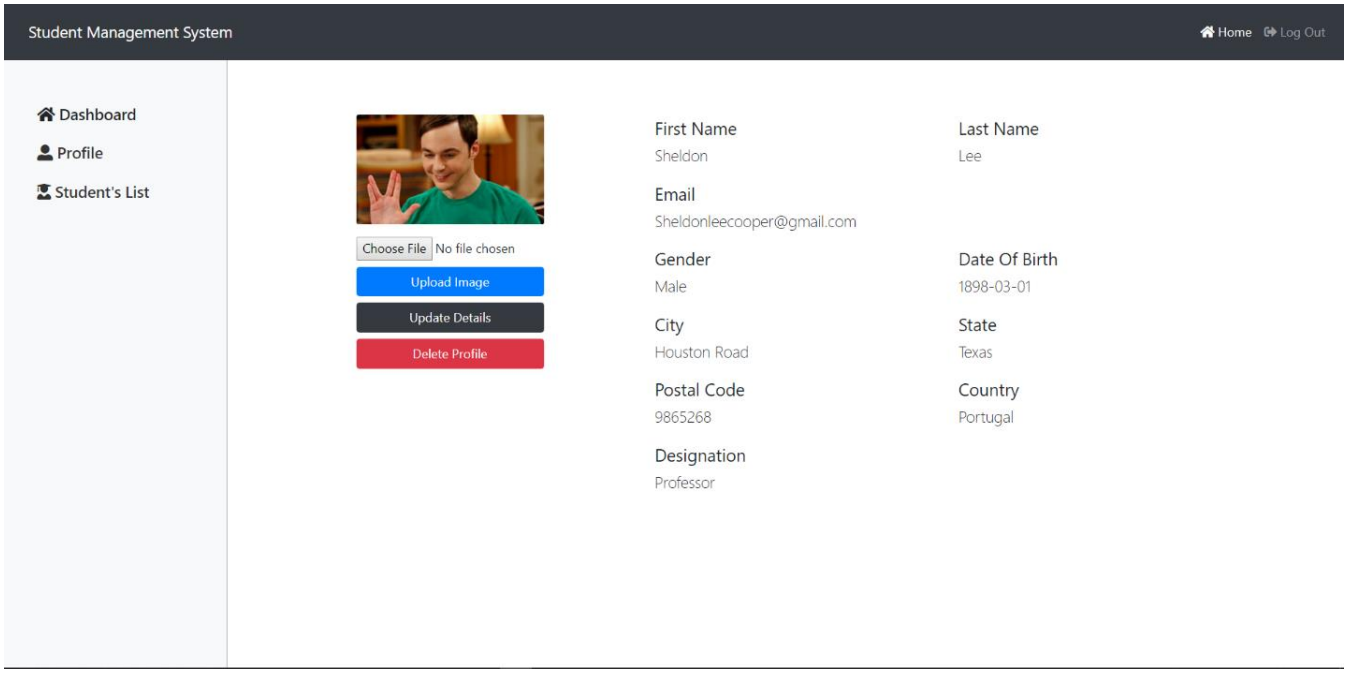


Fig 4.8 Mentor Profile

9. **Student's List (Mentor's Profile):** The student's list provides the table that contains the list of students enrolled, along with their profile picture, name, email, date of birth, college, course, year of passing, marks, city and state. It also contains a delete sign in each row through which you can delete each row individually. There's also an option of multiple deletion with the help of checkbox.

   Student list also contains an option of export data, which creates an excel file that contains the list of students in excel sheet. As soon as you will click the button, an excel file will be downloaded.

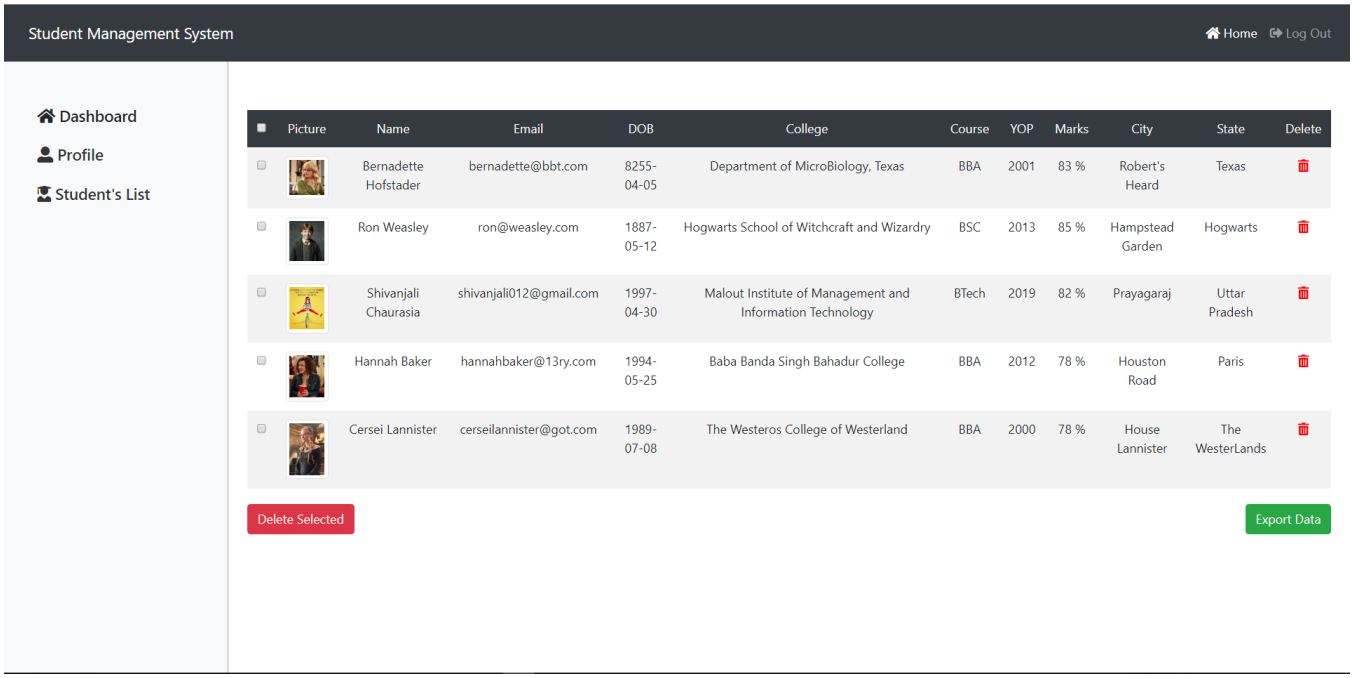


Fig 4.9 Student's List (Mentor's Profile)

10. **University's Dashboard:** As soon as you register, and select the user type as university, it will redirect you to the university dashboard page. If the user is visiting the page first time, and haven't filled any other details, then it will only display First Name, Last Name, and Email column filled, and will be disabled, so that user won't be able to edit it again.

    If the user is already registered and logging in, then he/she won't be able to update his/her founder's day as well as university registration number too.

    After entering dashboard, user will be asked few more details to fill to complete registration. User will be required to enter founder's day, List of colleges Associated, number of seats available and university registration number.

    It also consists of one checkbox that is proof of declaration that all the information provided above is true.

    If the user is entering his/her data for the first time, the update button will be disable for them, and they can only be submitting their data to the organization.

    On sidebar, in addition to mentor, university has also an option to view the list of students, list of mentors, as well as list of college associated with it.

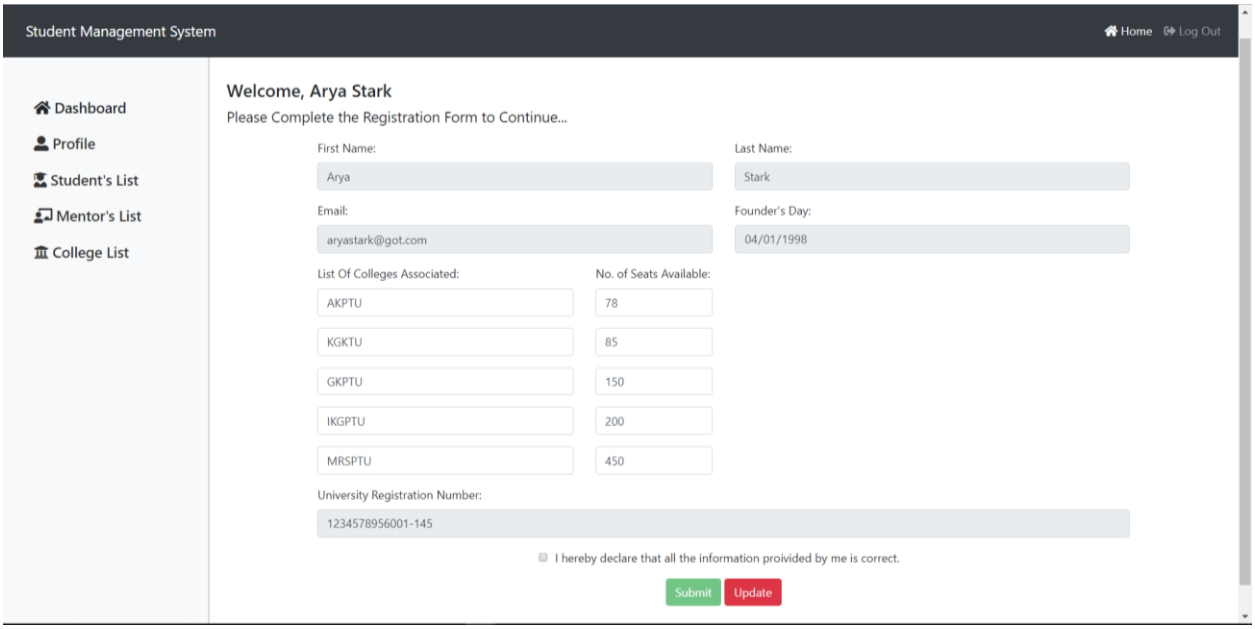


Fig 4.10 University Dashboard

11. **University Profile:** On the university profile page, university authority member can view his/her complete details along with an option of a profile picture, which by-default will be nothing, but if the user has updated it then it will show his latest profile picture. University profile also contains three buttons in them. First one is of upload image, by which the image can be updated, second is to update details, which will redirect us to the mentor dashboard, and the third option is to delete profile, through which the profile can be deleted.

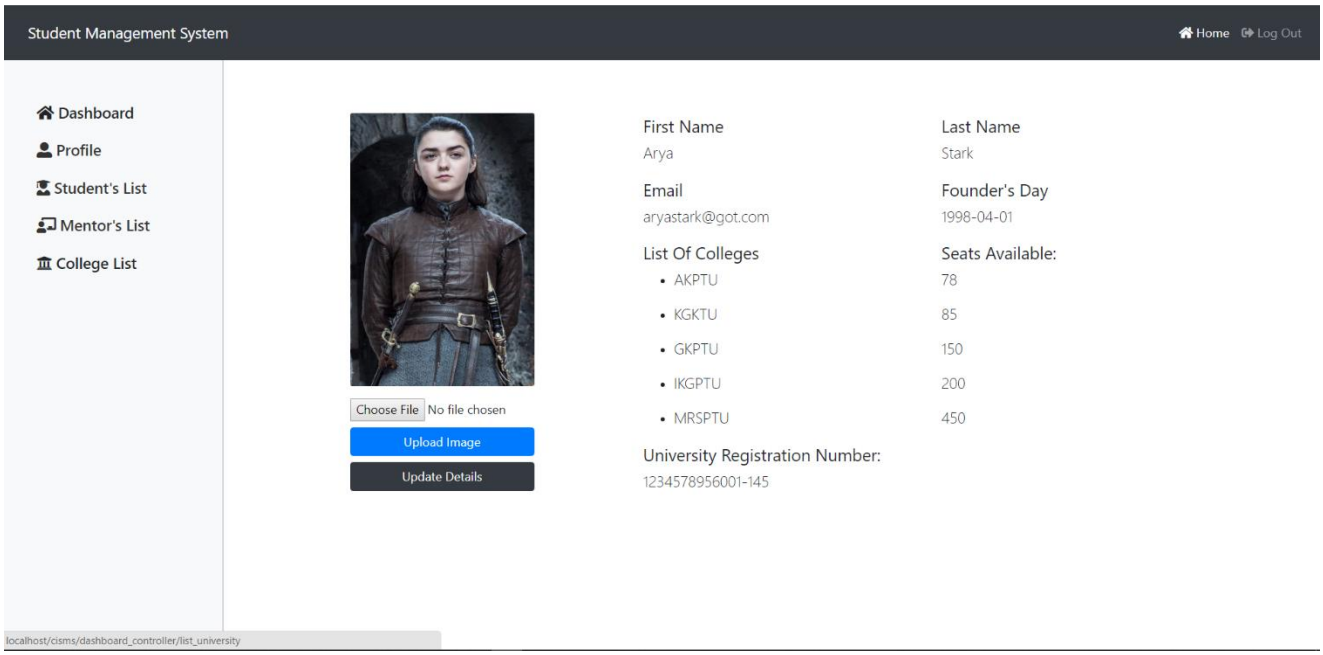


Fig 4.11 University Profile

12. **Student's List (University's Profile):** The student's list provides the table that contains the list of students enrolled, along with their profile picture, name, email, date of birth, college, course, year of passing, marks, city and state. It also contains a delete sign in each row through which you can delete each row individually. There's also an option of multiple deletion with the help of checkbox.

    Student list also contains an option of export data, which creates an excel file that contains the list of students in excel sheet. As soon as you will click the button, an excel file will be downloaded.

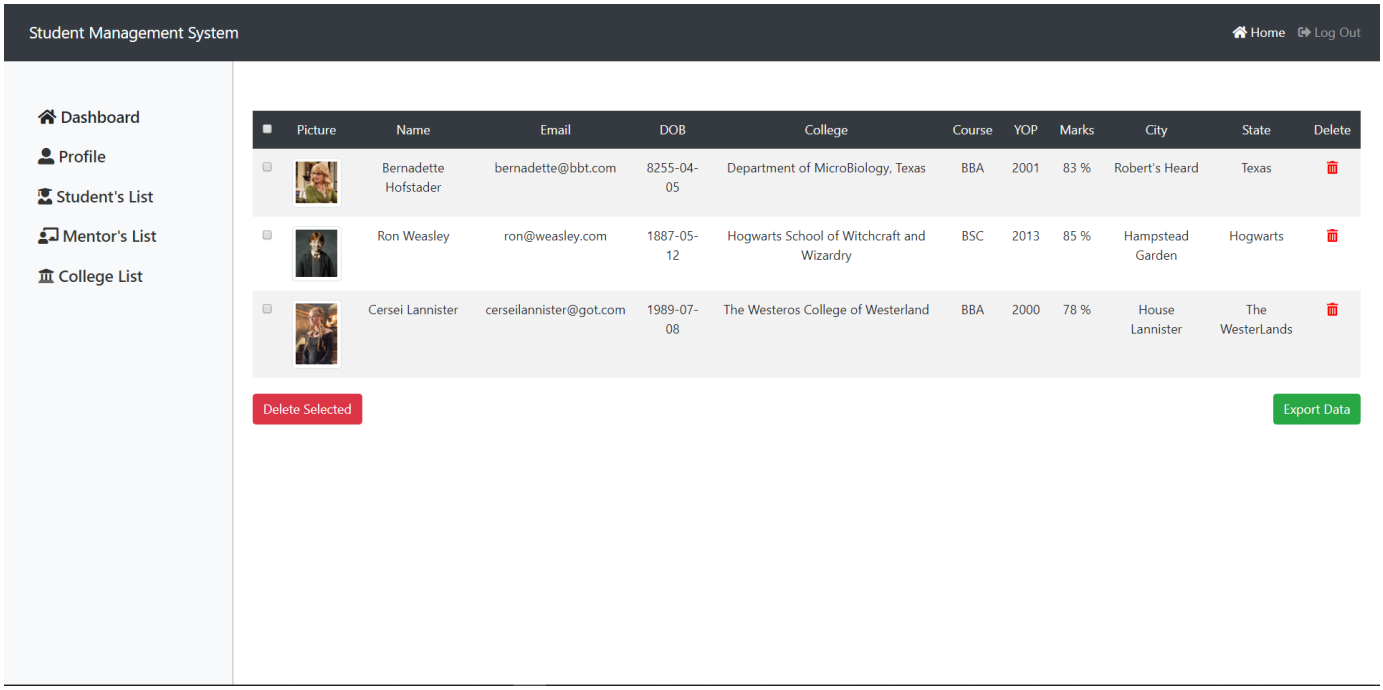


Fig 4.12 Student's List (University's Profile)

13. **Mentor's List:** The mentor's list provides the table that contains the list of mentors enrolled, along with their profile picture, name, email, date of birth, designation city and state. It also contains a delete sign in each row through which you can delete each row individually. There's also an option of multiple deletion with the help of checkbox.
    Mentor list also contains an option of export data, which creates an excel file that contains the list of students in excel sheet. As soon as you will click the button, an excel file will be downloaded.

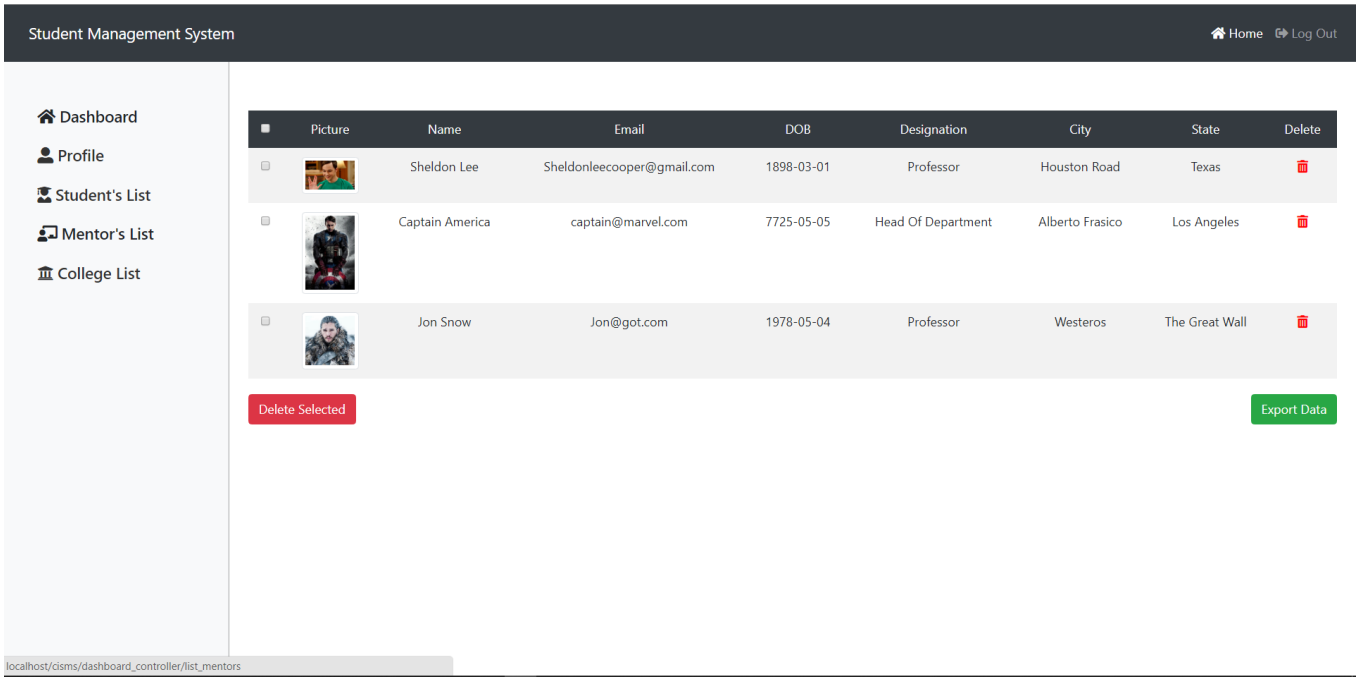


Fig. 4.13 Mentor's List

14. **College List:** The college list provides a list of college affilated and the seats available in each college in a table. There's also an option of update details, which will redirect us to the university dashboard page.

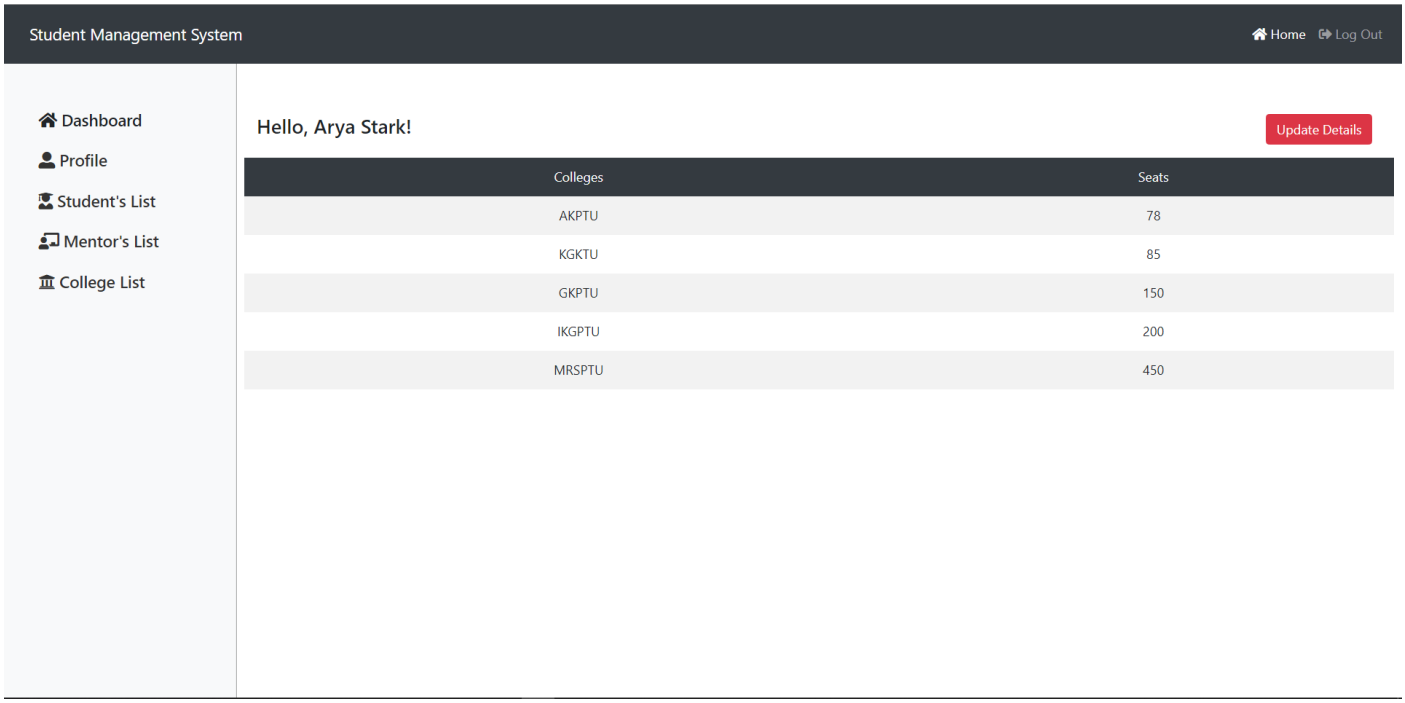


Fig. 4.14: College List

15. **Export Student Data:** After clicking on the export student, the excel file is generated, which contains all the list of students registered. The name of the file begins with Student Details along with the current date and time.

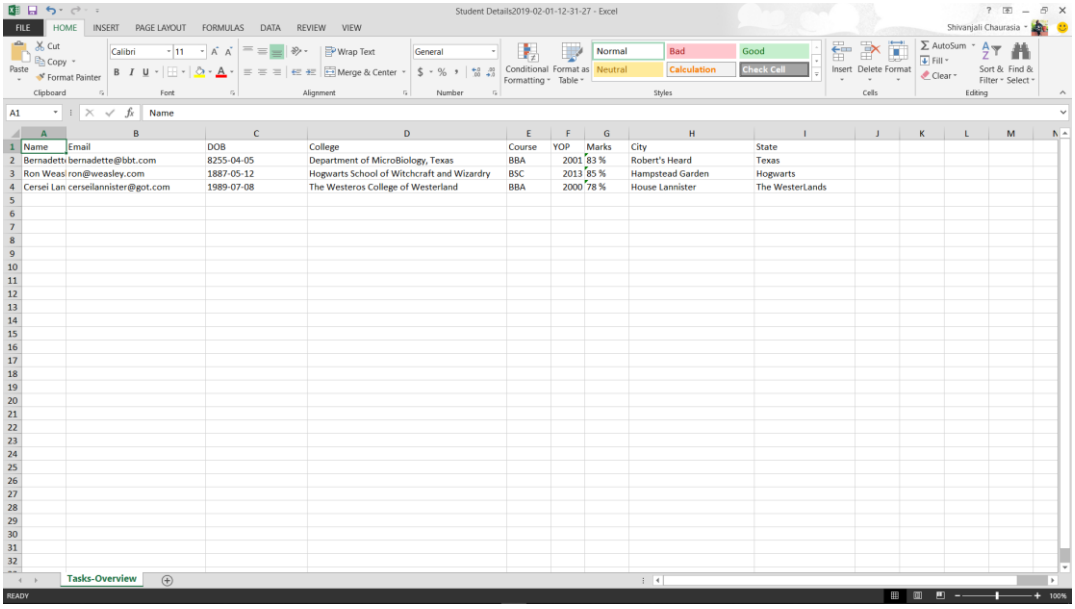


Fig. 4.15 Export Student Data

16. **Export Mentor Data:** After clicking on the export mentor, the excel file is generated, which contains all the list of mentors registered. The name of the file begins with Mentor Details along with the current date and time.

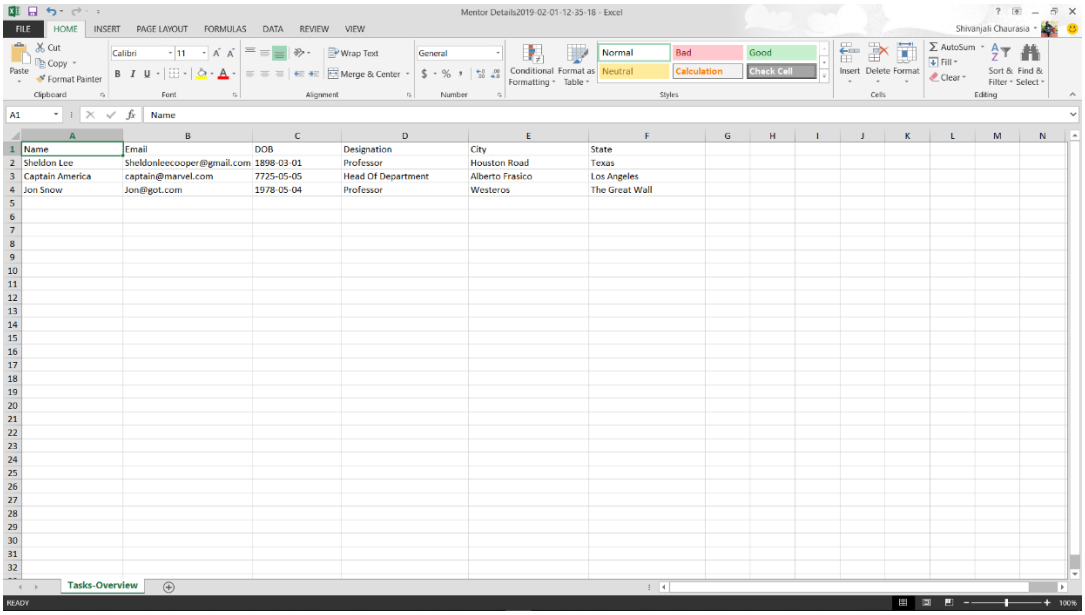


Fig. 4.16 Export Mentor Data

17. **Database:** The name of the database created here is “cisms”, and it contains four tables in it. Users, Student, Mentor and University.

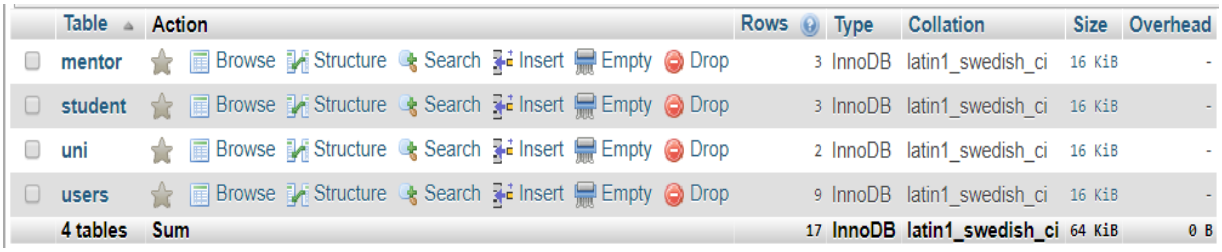
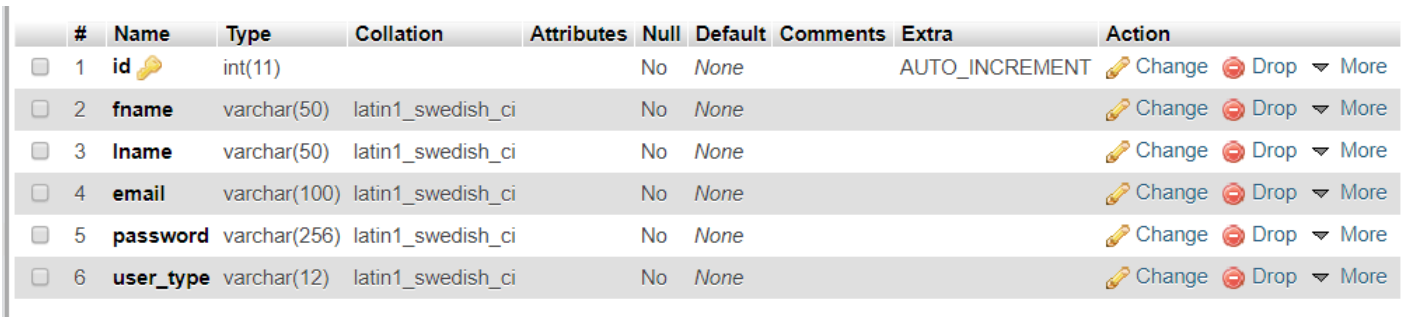


	Table	Action	Rows	Type	Collation	Size	Overhead
☐	mentor	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
☐	student	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
☐	uni	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
☐	users	Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16 KiB	-
	4 tables	Sum	17	InnoDB	latin1_swedish_ci	64 KiB	0 B

Fig. 4.17 Database

18. **Users Table Structure:** The user table consists of six columns named as id that contains the id of a user, it is also the primary key of the table, fname that contains the first name of user, lname that contains the last name of the column, email that contains the email of user, password that contains the password of user and then the user_type at the end, that contians the type of user which can be either student, mentor or university.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	fname	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	lname	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
☐	4	email	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
☐	5	password	varchar(256)	latin1_swedish_ci		No	*None*			Change Drop More
☐	6	user_type	varchar(12)	latin1_swedish_ci		No	*None*			Change Drop More

Fig 4.18 User Table Structure

19. **Student's Table Structure:** The student table consists of 13 columns named as id that contains the id of a user that matches the user table, email, date of birth as dob, gender, city, state, postal, country, course, yop as year of passing, marks, college, and stu_pic that contains the file name of the profile picture uploaded.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id** 🔑	int(11)			No	*None*			Change Drop More
☐	2	**email**	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	**dob**	date			No	*None*			Change Drop More
☐	4	**gender**	varchar(2)	latin1_swedish_ci		No	*None*			Change Drop More
☐	5	**city**	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
☐	6	**state**	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
☐	7	**postal**	int(15)			No	*None*			Change Drop More
☐	8	**country**	text	latin1_swedish_ci		No	*None*			Change Drop More
☐	9	**course**	varchar(10)	latin1_swedish_ci		No	*None*			Change Drop More
☐	10	**yop**	int(4)			No	*None*			Change Drop More
☐	11	**marks**	float			No	*None*			Change Drop More
☐	12	**college**	text	latin1_swedish_ci		No	*None*			Change Drop More
☐	13	**stu_pic**	varchar(256)	latin1_swedish_ci		No	profile.jpg			Change Drop More

Fig. 4.19: Student's Table Structure

20. **Mentor's Table Structure:** The mentor table consists of 10 columns named as id that contains the id of a user that matches the user table, email, date of birth as dob, gender, city, state, postal, country, designation and ment_pic that contains the file name of the profile picture uploaded.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	None			Change Drop More
☐	2	email	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
☐	3	dob	date			No	None			Change Drop More
☐	4	gender	varchar(2)	latin1_swedish_ci		No	None			Change Drop More
☐	5	city	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	6	state	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	7	postal	int(11)			No	None			Change Drop More
☐	8	country	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
☐	9	designation	varchar(5)	latin1_swedish_ci		No	None			Change Drop More
☐	10	ment_pic	varchar(256)	latin1_swedish_ci		No	profile.jpg			Change Drop More

Fig. 4.20: Mentor's Table Structure

21. **University's Table Sturcture: Structure:** The university table consists of 15 columns named as id that contains the id of a user that matches the user table, email, founders _day, five colleges and seats associated with them, and uni_pic that contains the file name of the profile picture uploaded.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	None			Change Drop More
☐	2	email	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	3	founder_day	date			No	None			Change Drop More
☐	4	college1	varchar(256)	latin1_swedish_ci		No	None			Change Drop More
☐	5	seats1	int(7)			No	None			Change Drop More
☐	6	college2	varchar(256)	latin1_swedish_ci		No	None			Change Drop More
☐	7	seats2	int(7)			No	None			Change Drop More
☐	8	college3	varchar(256)	latin1_swedish_ci		No	None			Change Drop More
☐	9	seats3	int(7)			No	None			Change Drop More
☐	10	college4	varchar(256)	latin1_swedish_ci		No	None			Change Drop More
☐	11	seats4	int(7)			No	None			Change Drop More
☐	12	college5	varchar(256)	latin1_swedish_ci		No	None			Change Drop More
☐	13	seats5	int(7)			No	None			Change Drop More
☐	14	reg_no	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
☐	15	uni_pic	varchar(256)	latin1_swedish_ci		No	profile.jpg			Change Drop More

Fig. 4.21 University's Table Structure

CHAPTER 5

BIBLIOGRAPHY

Web Links:

[1] https://www.codeigniter.com/userguide3/database/query_builder.html

[2] https://www.codeigniter.com/user_guide/libraries/sessions.html

[3] https://www.codeigniter.com/user_guide/libraries/form_validation.html

[4] https://www.codeigniter.com/user_guide/libraries/form_validation.html#callbacks-your-own-validation-methods

[5] https://www.codeigniter.com/user_guide/libraries/file_uploading.html

[6] https://www.codeigniter.com/user_guide/general/helpers.html

DECLARATION

I hereby declare that the project entitled **"Student Management System"** submitted for the Web Development and Design Internship is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of Intern:

Shivanjali Chaurasia