Divya Sravani Bandi

A20397609

University Housing System (Assignment 1)

Principles of Database Design

**Report on**

**UNIVERSITY HOUSING SYSTEM**

**(WEB BASED DATABASE DESIGN)**



1. **Design and Development of the Database**

**Introduction:**

University Housing System is useful for the administration of a university in maintaining the housing provided by the university. It allows admin to manage the database of students who are willing to join the university housing. Students who fulfill all the requirements of the housing facilities are allotted with a room and those who do not have sufficient requirements will be rejected. When students book a room from their side, housing manager, who is the admin, validates the information provided by the them and determines weather or not to assign a room.

**Design of the database:**

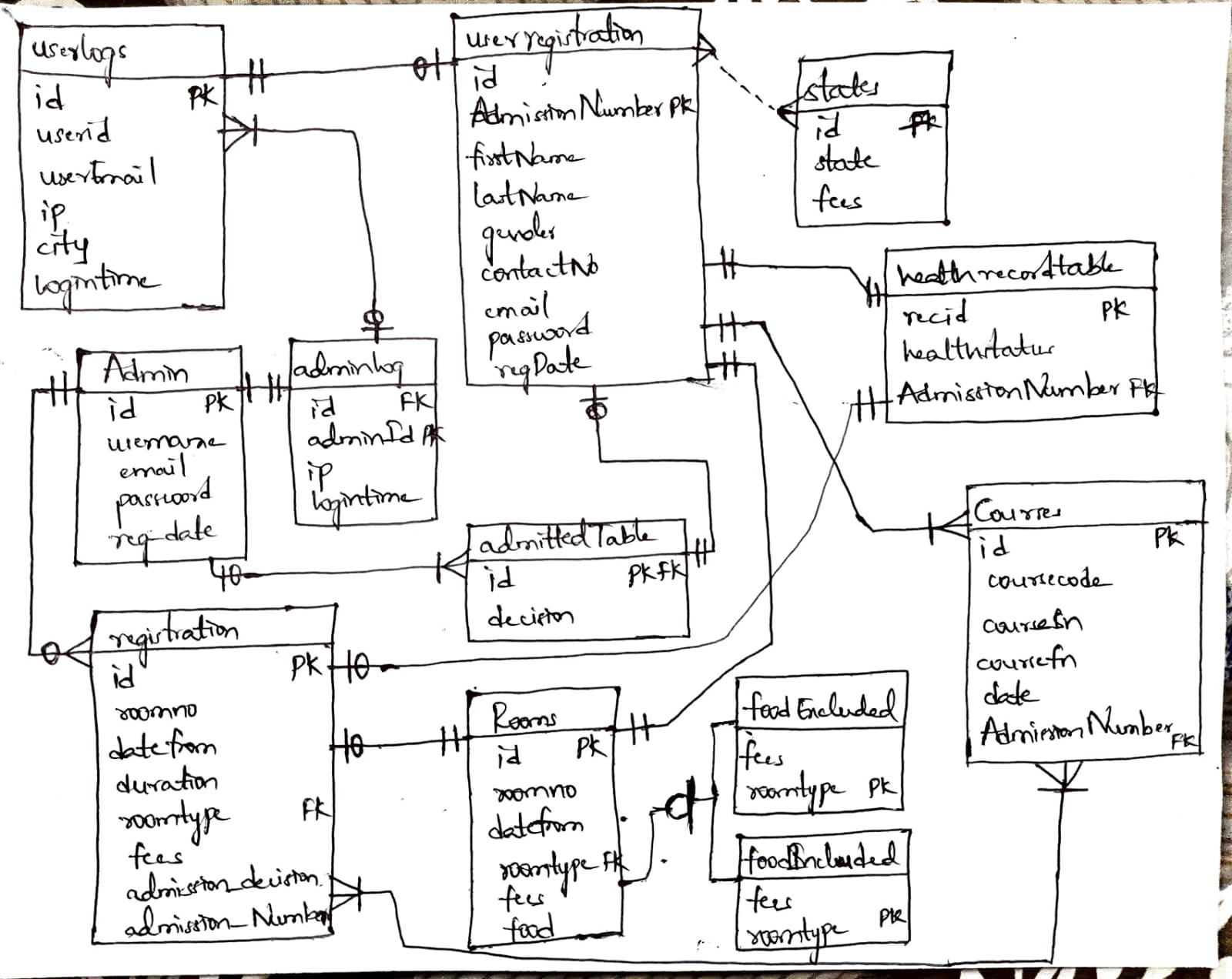
When a student wants to book a room in the university housing, they will have to contact either the housing manager or they can also book by self-registering directly from the University Housing website by filling the student registration form. Prior to booking the room, student has to have an admission from the university and must also clear health records hold. Student is given admission if only all the health certificates, vaccine certificates are provided to the university management. If the student wants to book a room through the housing manager, the housing manager immediately checks the required documents from the management, If all the holds are cleared, they will be allotted a room and allowed to register for courses, otherwise rejected or withheld. If the student wants to directly book the room using self-register option from the website, the student initially has to register in the website. Then the admin validates the information provided by the student along with his or her documents, certificates and also checks if the holds are cleared. If the student’s record is found to be sufficient then student will be allowed to book room otherwise rejected. Therefore, according to the design, student is allowed to book a room and register for courses only if accepted. Once prompted to book a room, students have to select one of the room types. And the fees is dependent on the type of the room selected. Later, students will be given a choice of selecting with food or without food where student can select their own choice. Depending on the option they choose, total fees for booking the room varies. If the user chooses with food, fees associated with meal plan and the fees for the room will be total fees. Stored procedures are created to store the log in details of the user and admin, getStudentlog and getAdminlog.

Triggers: Once user registers, user can login to check the room details and the course details associated to his or her account. For this to take place, users’ login and password are checked with the existing data. If they belong to the existing data, user will be logged in. Admin login credentials are also checked in the same way with the already existing admin data. Admin is logged in only when appropriate credentials are entered. A trigger is created to store the logged details like ip address, time of admin in adminlog.

Subtype entities:The below ERDiagram shows the entity Rooms which is a supertype entity and the two entities foodincluded and foodexcluded are subtype entities. If it is selected as withfood, fees from the foodincluded entity will be considered whereas if is selected withoutfood, fees from the foodexcluded entity will be considered to calculate the total fees.

Weak Entity: The entity states depends on the useregistration completely. It has a foreign key id and other no primary key attributes. This entity deermines the fees based on the state as instate or outstate fees.

**ER Diagram:**



Tables:

Userregistration table:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | AdmissionNumber | firstName | lastName | gender | contactNo | email | password | regDate |
| int | Varchar | Varchar | Varchar | Varchar | Varchar | Varchar | Varchar | timestamps |

Admin table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | username | email | password | Reg\_date |
| int | Varchar | Varchar | Varchar | timestamps |

Rooms table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | roomno | datefrom | Roomtype | fees | food |
| int | int | date | int | Varchar | Varchar |

Healthrecords table:

|  |  |  |
| --- | --- | --- |
| recid | healthstatus | AdmissionNumber (FK) |
| int | Varchar | Varchar |

Registration table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id | roomno | datefrom | duration | Roomtype | fees | Admission\_decision | admissionNumber |
| int | int | date | Varchar | int | Varchar | Varchar | Varchar |

Courses table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | coursecode | coursesn | coursefn | date | AdmissionNumber |
| int | Varchar | Varchar | Varchar | date | Varchar (FK) |

Admitted table:

|  |  |
| --- | --- |
| Id (FK) | decision |
| int | Varchar |

States table:

|  |  |  |
| --- | --- | --- |
| Id (FK) | state | fees |
| int | Varchar | Varchar |

All the tables are in 1NF as each table satisfies atomicity. Partial dependency exists when a nonprime attribute is a proper subset of candidate key. Tables do not possess partial dependency hence the tables are in second normal form. For the tables to be in 3rd normal form, they have to be in second normal form and there shouldn’t be transitive dependency for nonprime attributes, i.e., nonprime attributes must not depend on other set of nonprime attributes in the table. The above tables are in second normal form and they also do not possess transitive dependency.

Tables are said to be in Boyce Codd Normal Form if they are in third normal form and right hand side of the functional dependency should depend on the super key of the table i.e., left hand side must be a proper subset of the superkey of the table. For example, in the userregistration table, all the non primary key attributes depend on the super key of that table like AdmissionNumber🡪firstName.

**Implementation of the database**

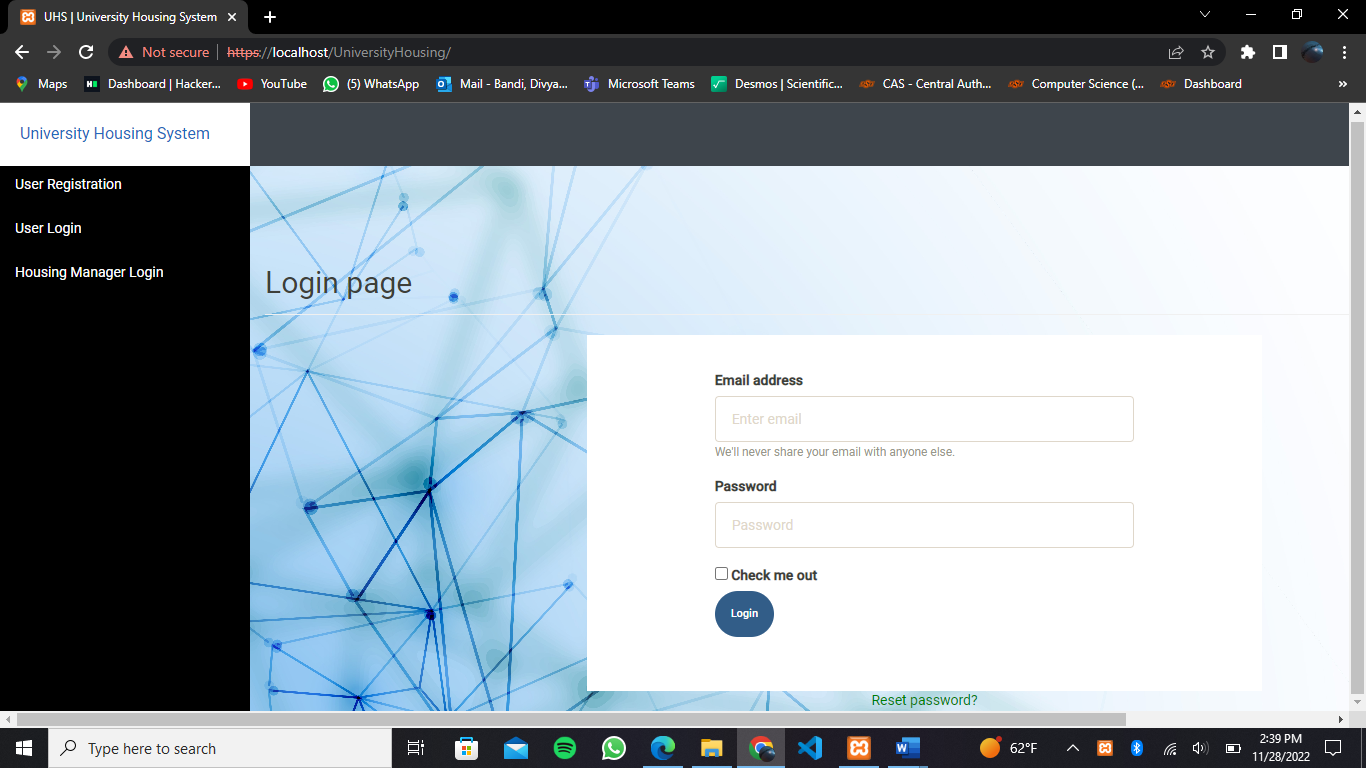
The database is implemented using PHP and the connection is established using xampp.

Frontend tools used: HTML, CSS, Javascript. HTML is a markup language used to structure the website. CSS is used to present the document written in markup language.

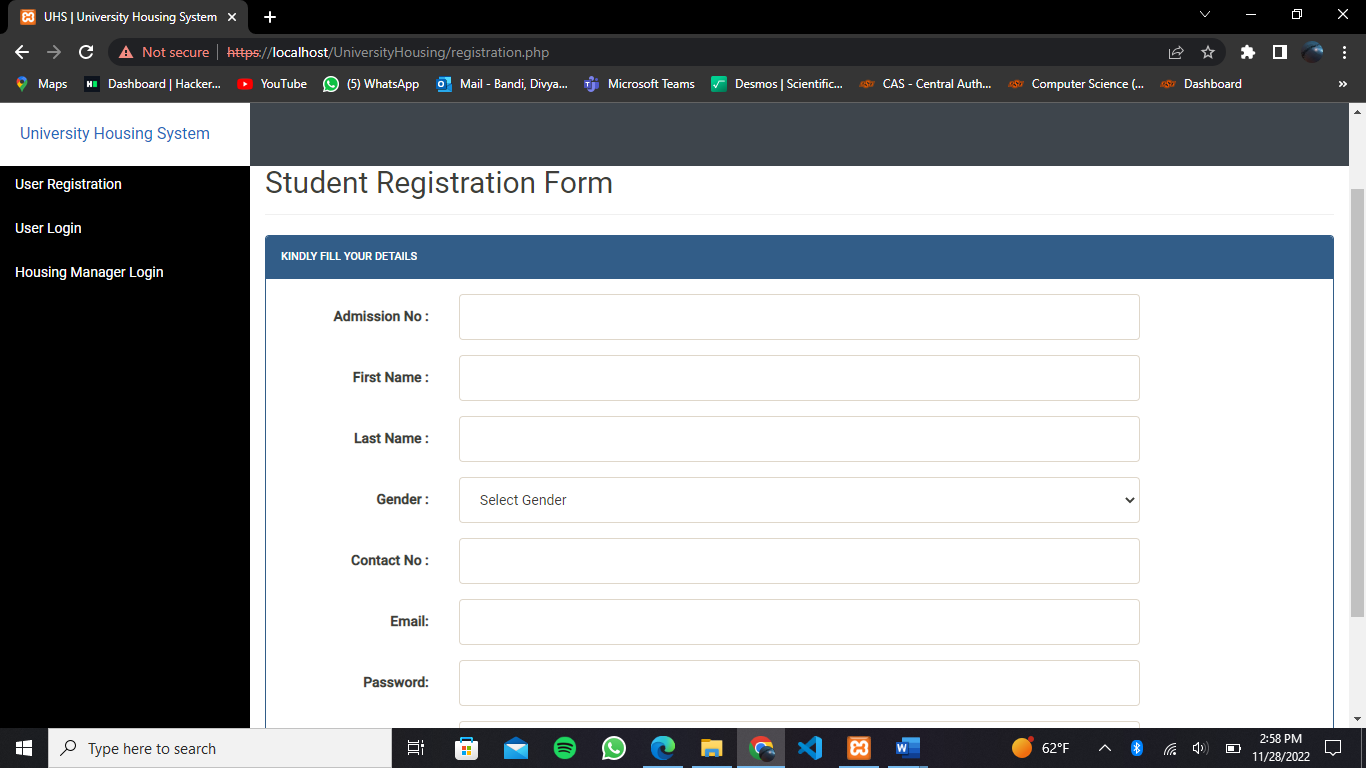
Backend: PHP is used, which is a high-level server-side scripting language.

Software: XAMPP, a fully functional PHP/MySQL server environment installed by using PHP Triad. PHP, MySQL, Apache, and PhpMyAdmin are installed. On installing xampp, PhpMyAdmin is used to manage the databases. PhpMyAdmin allows users to create databases, tables, views, stored procedures, etc. It allows to insert data into the tables, retrieve data, write SQL queries and execute them. Connection has to be established before viewing the data which is all coded using Php. Before deleting, PHP can look through databases, tables, views, fields, and indexes. PHP allows to create databases, tables, fields, and indexes; keep servers, databases, and tables up to date; execute, modify, and bookmark any SQL query, including batch queries. tables with text file data.

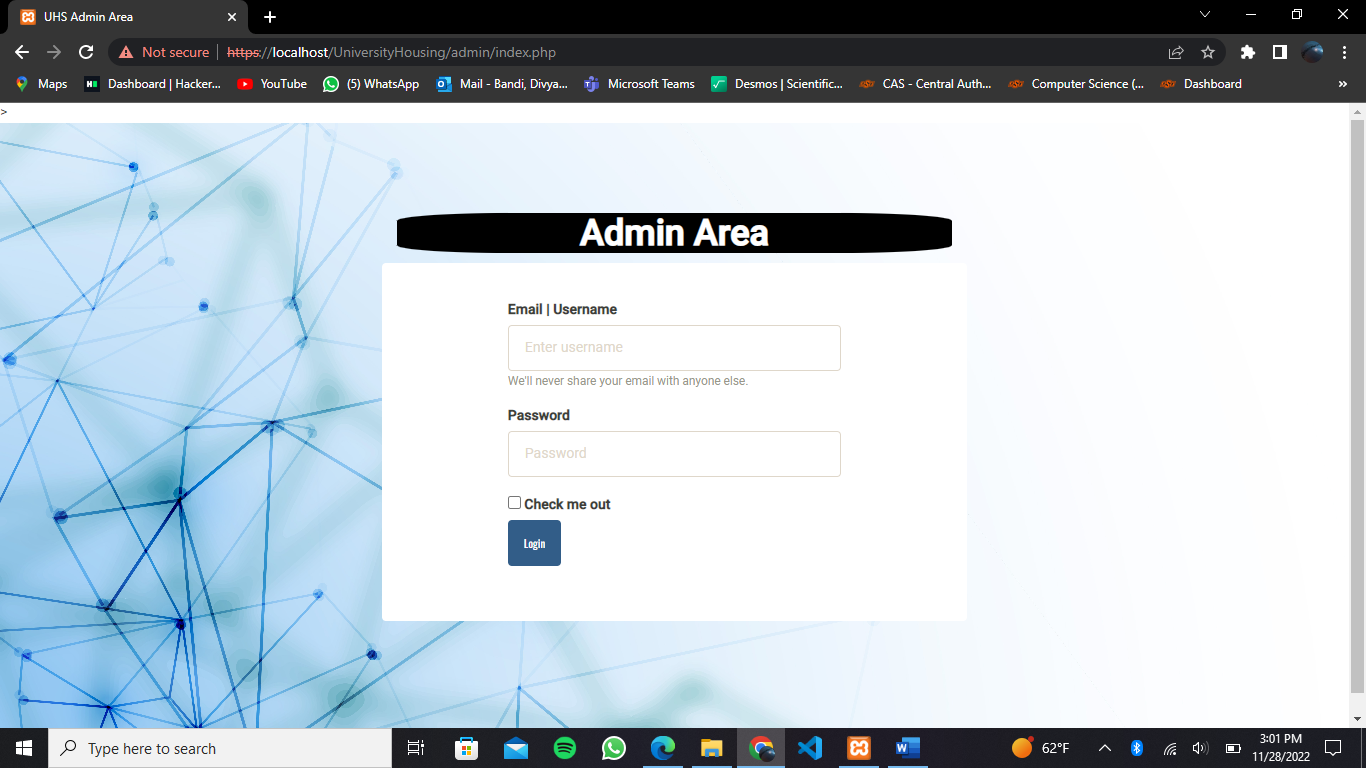
**Appendix**

****

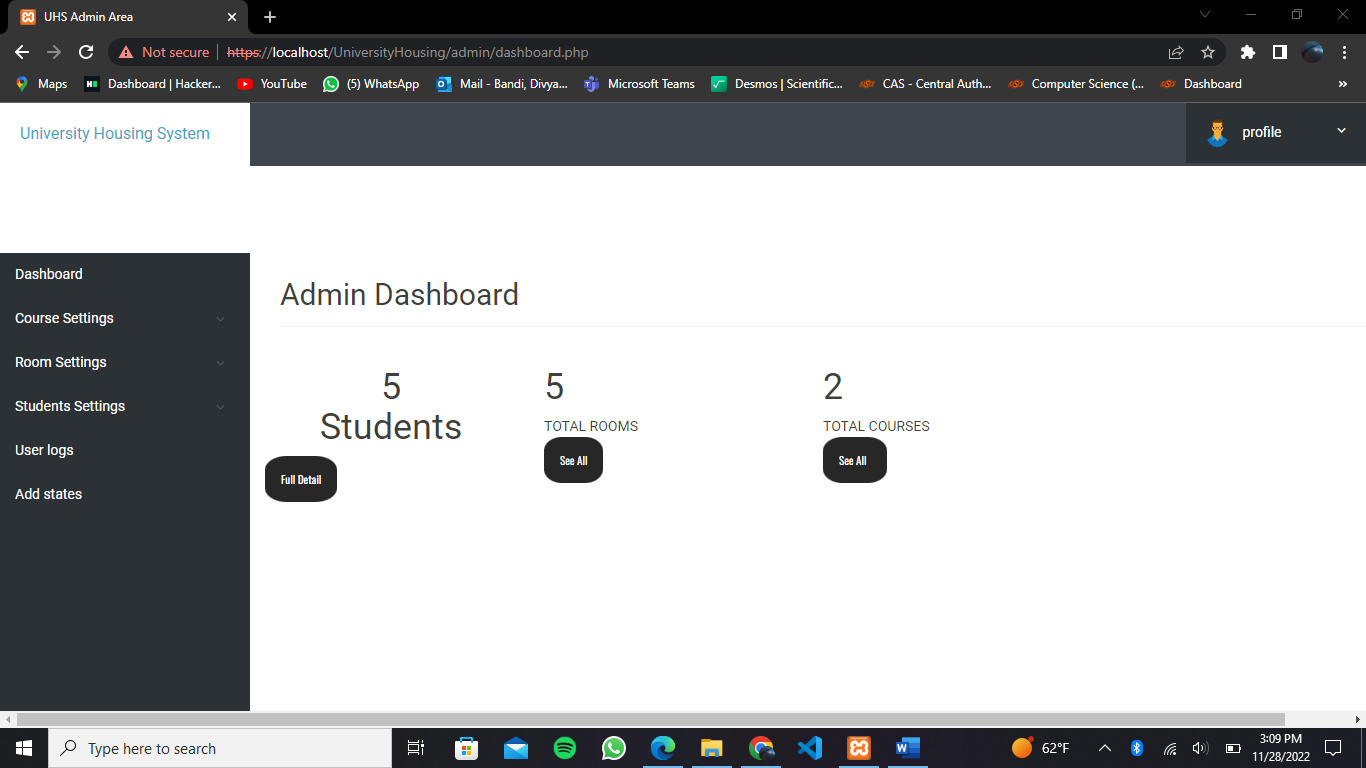
The above picture shows the login page of the website that first appears when the website is opened.



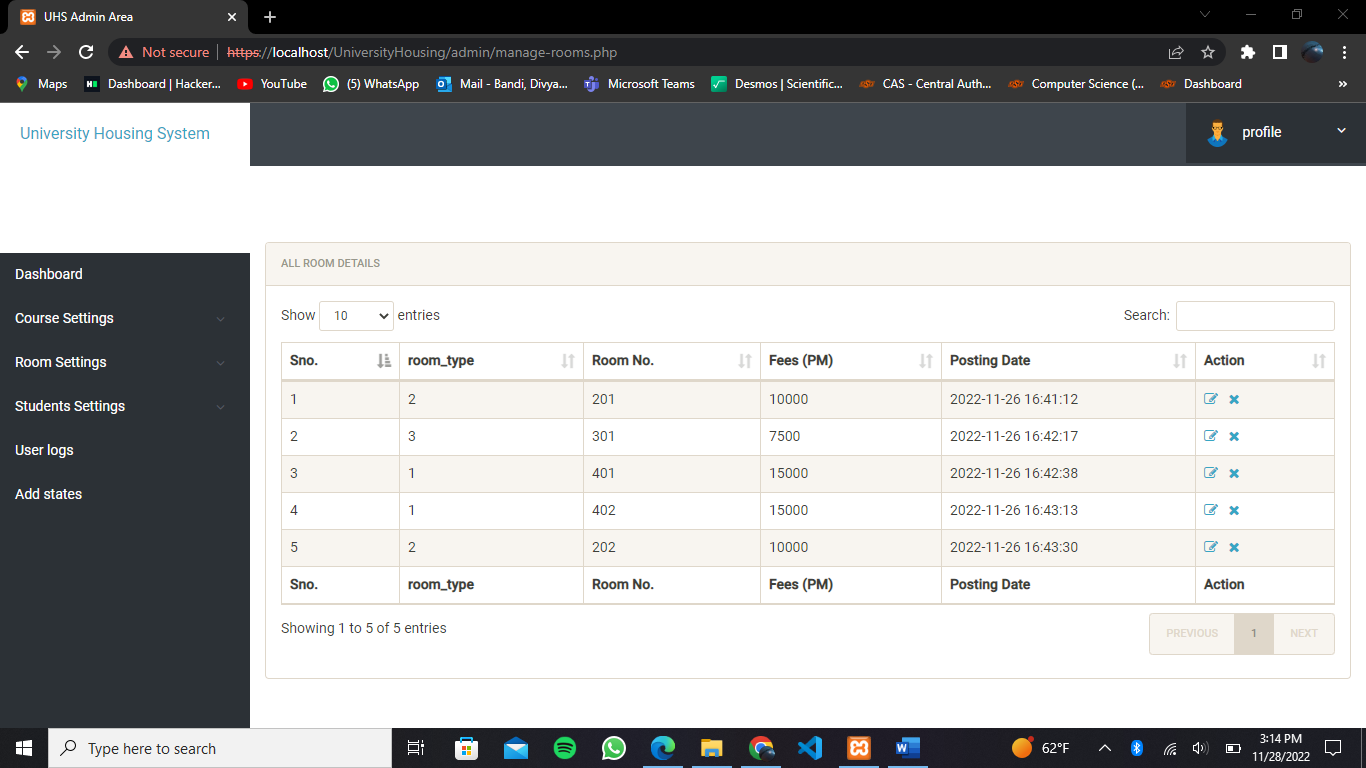
The above image shows student registration form, where the student can directly self-register to book a room, using this page.

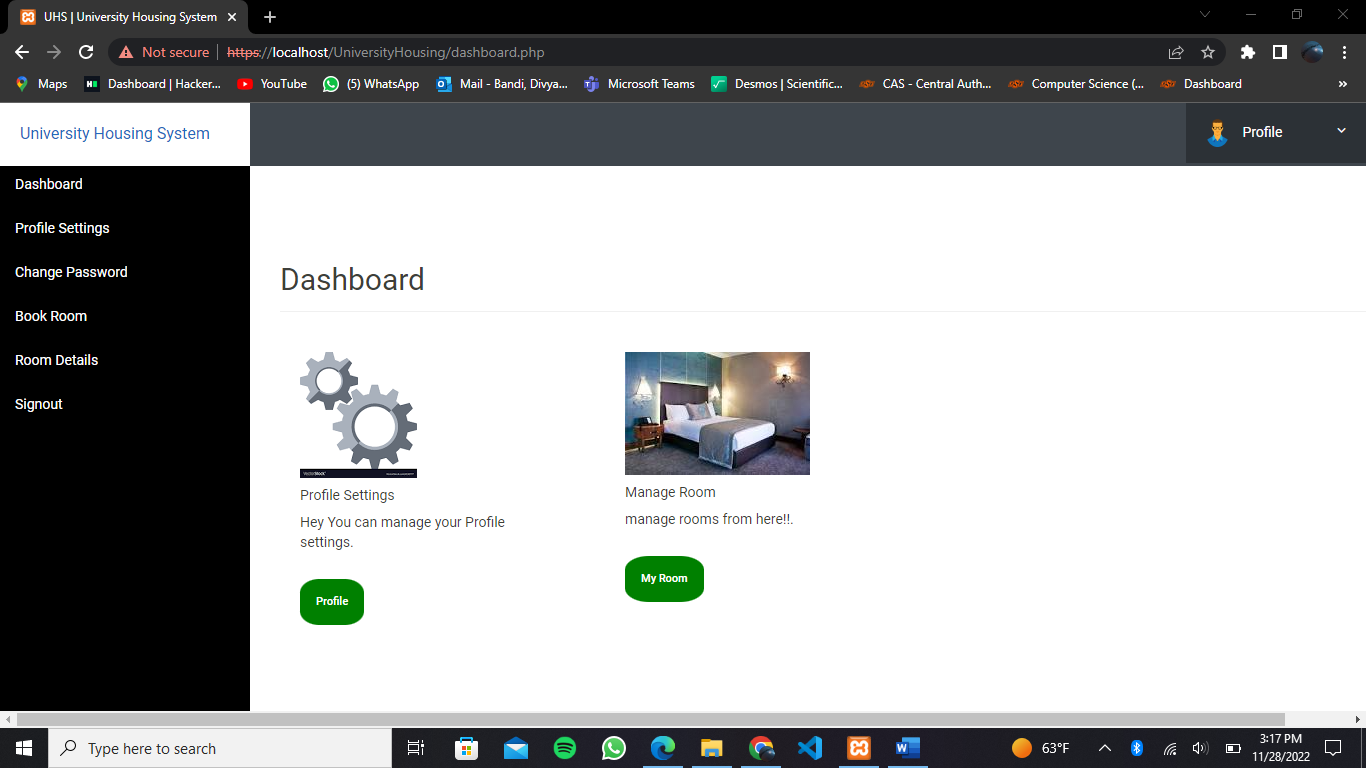


The above image shows a page where admin can login using the appropriate login credentials.



The above picture shows the dashboard of the admin side. It shows number of students, number of rooms and courses. Admin can also manage room, students’ data. Admin has the access to add and delete courses, rooms, students. Admin can register a student and can also view the userlogs.



The above picture shows manage rooms page where admin can check the rooms availability, roomtype and fees.

The above picture shows the dashboard of user login after registration, user can check the room that is booked or book a room if not booked yet. User can also change password or update profile.