[SIDE-HUSTLE (PHP CAPSTONE)]

[Team 271-310]

THEME: Hostel Management System using PHP

Live Demo: https://hostel.trenalyze.com

Objective

* Problem Definition
* Problem Algorithm/Users Guide
* Developers Guide
* Design Specification
* Data Flow Diagram
* Summary
* Project Definition

We decided to handle this project having in mind that before now students will have to travel to the school after they receive their admission message, just to get a hostel early before the rooms will be occupied by others. But this is way too stressful, so we decided to come up with an idea to build a hostel Management Website. Which makes it easier for every student to visit their school Portal, register and then get their room number immediately from anywhere they are without having to travel to visit the school and cue up to get a hostel room.

Here is the live demo link of this project:

https://hostel.trenalyze.com

----------------------------------------------------------------------------

* Design Specification

1. A laptop or desktop.
2. Color SVGA.
3. An internet connection.

----------------------------------------------------------------------------

* Problem Algorithm

The school already has a database of all students given admission into the school, and also a database of all the available hostel rooms in the school. Every student must have a room space and pay for it and in order to have a room space you must register on the school portal.

The students visits the school web portal and registers with his/her Reg No (assigned to them by the school) and Email which they used during their admission process. If the reg no does not match the email address in the school database then the student cannot register but if they both match the registration will be successful and the system automatically fetches other data associated to that students such as first name and last name and fills into the users table then it will check for any available hostel room and assigns the room to that student and leads the student to his/her dashboard.

If the system could not find any available room or in other words all room all already occupied then the user cannot register at that time it will tell the user to check back later because there is no available room for the student

**OTHER FEATURES:** The students can students can still be able to edit their school details (email address) and site details (username and password) in their dashboard. There is also forgot password page where users request to change their password anytime they feel like.

* Developers Guide

This project was arranged in a well-defined manner it consists of 2 folders and some view files needed.

The assets folder contains all the styling, Javascript files and images used for this project.

The database folder contains the database query which will be imported to your database server and the configuration file which sets up the database Connection.

Other files contained in this projects are the pages needed which are the layout.php page containing the header layout of this website.

The layout\_footer.php page containing the footer layout of the site.

The register.php page contains the user/student registration Form.

The edit\_account.php page contains the form and link to edit the student’s login details.

The edit\_account\_c-password.php page contains the form and link to edit the user’s login password while logged in and this works by accepting the user’s valid old password and specifying the user’s new password.

The index.php page is the student’s dashboard where information are being displayed for the user to see. If a user is not logged in this index.php page will redirect the user to the login page.

The login.php page contains the form for the user to login to his/her dashboard.

The server.php page is where all php validations were done. This page contains functions which is called whenever a user wants to login, register, edit profile or password and variables to display some relevant information to the user are also declared and initialized here. hence every other pages includes the server.php at its top.

The success.php page contains some styles and information to display when a validation is successful and the error.php page does otherwise.

* Database Diagram (Normalization/Entity Relationship)

In the database design for this website we took note of normalization and entity relation in the tables so that a table will not have large data stored into it and takes much space than expected. The database consists of three main table the

Students Table: all students who exists in the school both the ones who have registered on the school portal and the ones who are no serious.

Users Table: all students who have registered on the school portal and gotten a room which is related to the students table

Rooms Table: which contains details of all rooms that are in the school both the ones available and the ones taken. This table is related to the users table

**DIAGRAMATIC REPRESENTATION**

|  |  |
| --- | --- |
|  | **Users** |
| **PK** | **id** |
| **FK** | **student\_id** |

|  |  |
| --- | --- |
|  | **Students** |
| **PK** | **id** |

|  |  |
| --- | --- |
|  | **Rooms** |
| **PK** | **id** |
| **FK** | **user\_id** |

* Data Flow Diagram

Cart Item

Login

cart

Register

Select Item

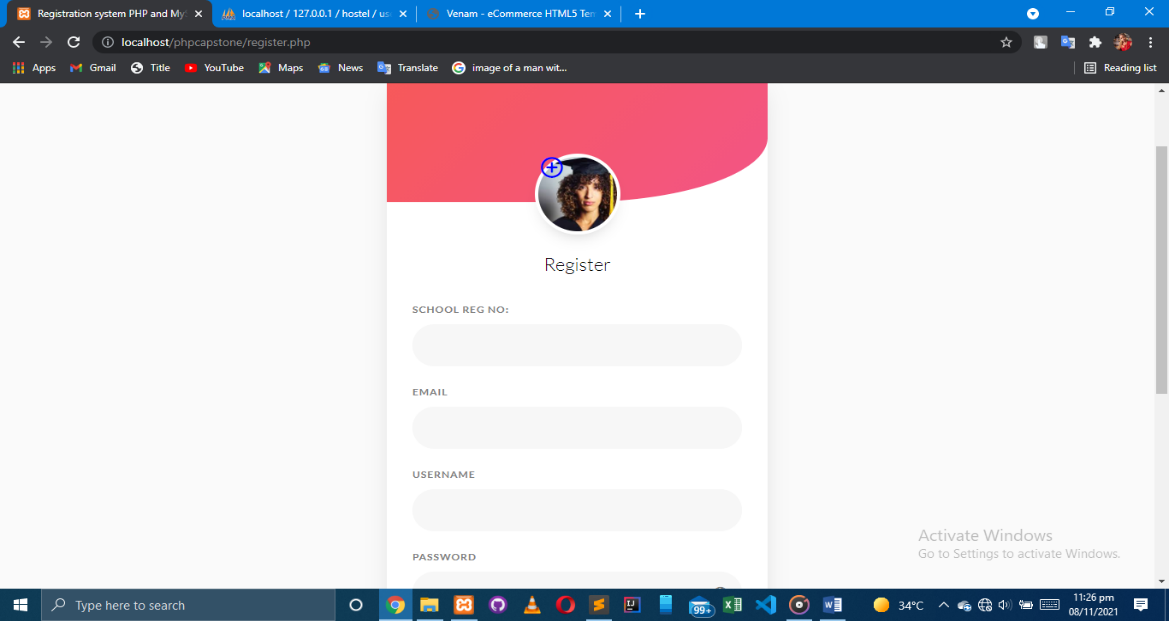
Student

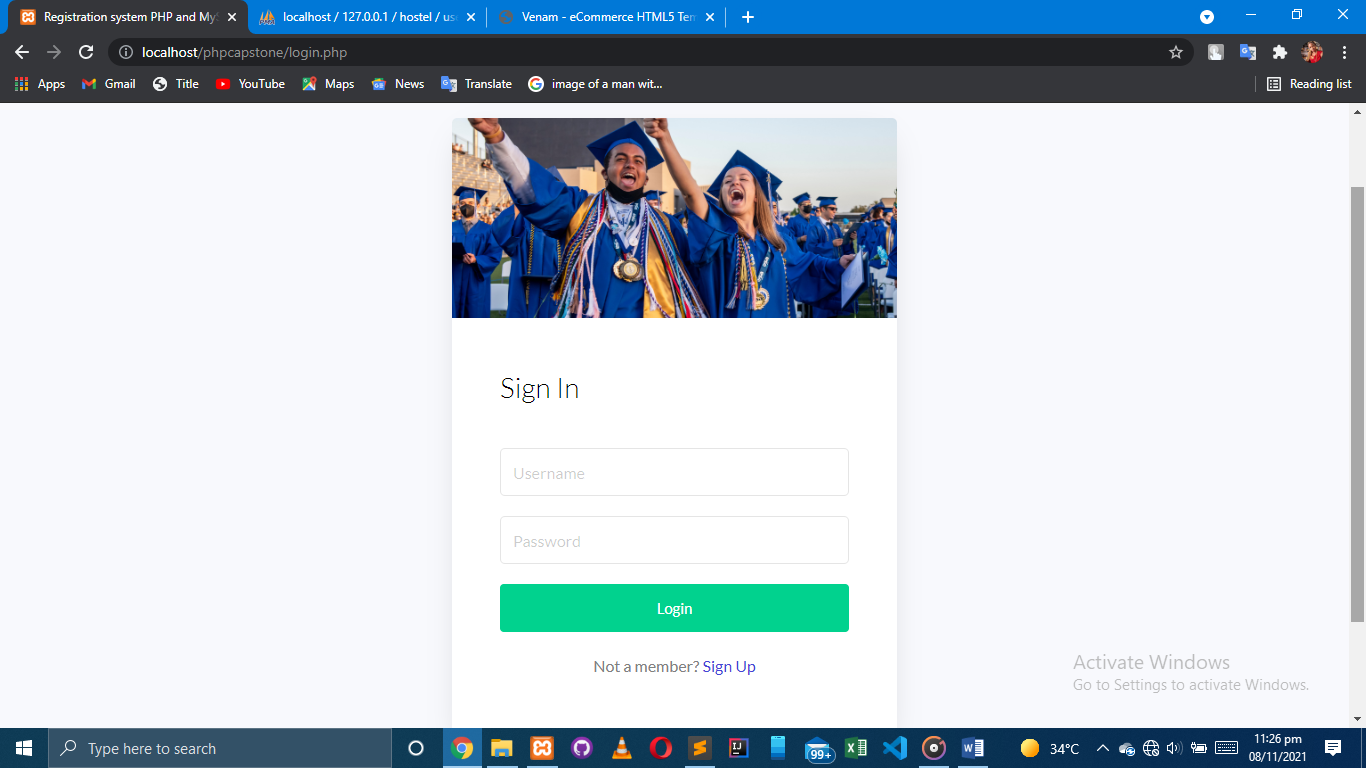
Gets a room

* Summary

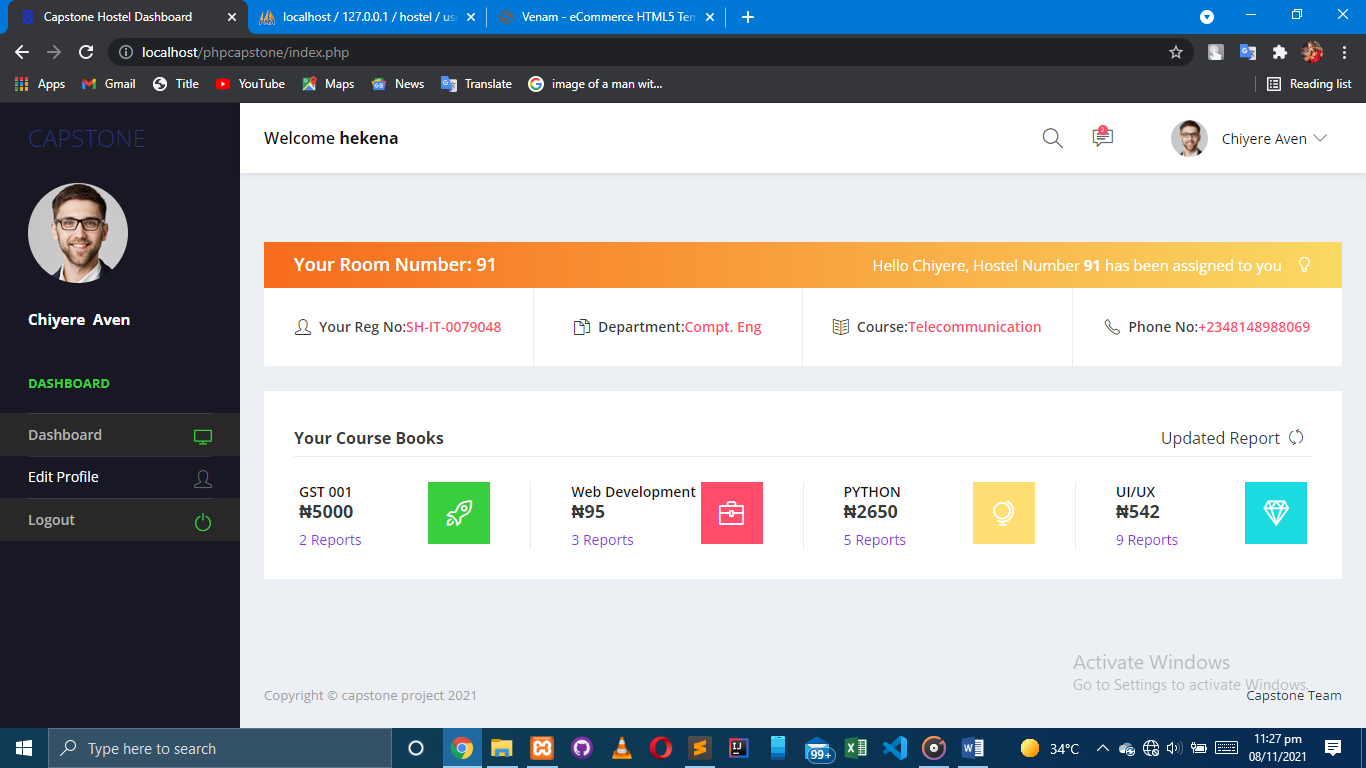
It was really fun handling this project here are some screenshots of our pages and how they look like

**The Registration Page:**

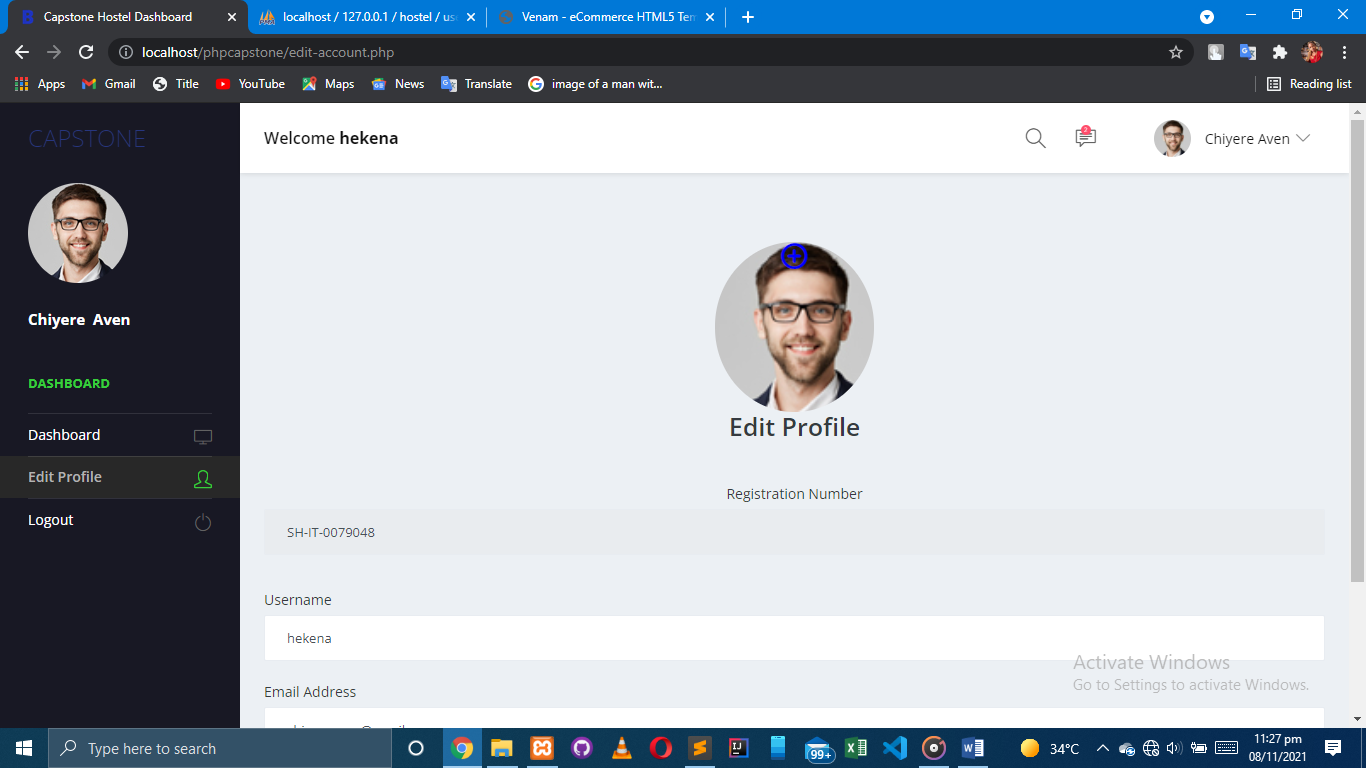


**The Login Page**

**The Students Dashboard**

****

**The Edit Account**

****