HOSTEL ATTENDANCE AND LEAVE MANAGEMENT SYSTEM FOR AN EDUCATIONAL INSTITUTION

Abstract

Managing hostels is similar to running a hotel. It involves several tedious and manual tasks of maintaining student data, keeping track of attendance, students in-out, allocating rooms, and several aspects. Most schools/college hostel wardens allow their students to an outing during holidays/ week-offs. They usually maintain IN-OUT registers for the purpose. However, the system is heavily flawed, as the paper-based system can be manipulated easily, keeping the security of the system at stake. Hostel Leave management project is a web based system which can be accessed by all students of a particular university. This system is automated system for managing leave of students and approval of leaves. Every student is provided with unique user id and password for login to the system and send request for leave whereas in our project their login is being connected with Student Information System (SIS). Wardens will look after the request and they may accept or reject, seeing the students parent approval and the student's attendance details in case of working days. Before approving the leave different department people will look after the request through this system and take decisions. This system will update the process of leave management system inside the hostel by saving time and resources. This help the students to view their own leave balance, leave status, view past leave history and also helps wardens to review and approve leave applications. This Leave Management system is very useful for any hostel of any organization where there may be more number of people (students). It is a single click process. The Hostel Leave Management system also aims at linking academic and hostel activities, leave details to a unique Student Information System of a particular organization.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1. OVERVIEW

The process of hostel registration in many universities employs a parochial system that involves students going to the Students' Affairs Office to fill out a form for registration. This form seeks to find out details of the student. Thereafter, students will register for the new session by providing a bank teller receipt to show payment has been made and a new hostel would be allocated to the student. Registration, students' profiles would be transferred to the porter's lodge where students will receive a mattress, pillow, chairs, and tables. Unfortunately, these processes are carried out manually with pen and papers, an unreliable procedure which also wastes time.When hostel admissions are taken details of student Hostel Management System Project Report Hostel Management System Project. Since all possible completion times between the minimum and maximum durations for ever task have to be considered, there is not one but many critical paths, depending on the permutations of the estimates for each task. Hostel Management System written as a project Uses an access database to store. Find Code: Hostel Management System: Author: Street Life: The system titled "The RealschuleeXp 2005" is very flexible and can be changed at any point of time. This system is user friendly, because of GUI features. This system simplifies the maintenance of large amount of data. The speed of accessing the information about the member's is very useful. It minimizes the time and efforts of the user with efficiency. This project is developed such that any implementation or extension can be done easily. The "Hostel Management System" project is divided into two Schools are growing in India to match the pace of population growth. It has outsmarted even the population growth rate of India. A big school, like one dealt in this project, face a large number of problems involving data maintenance, storage and mining. The manual system of data processing and record keeping is error prone, labour intensive and thus costly. It requires a lot of paper handling which further requires proper storage facilities. Moreover, this modus operandi adversely affects the smooth functioning of the organization. This system also figures out the human engineering considerations (ergonomics) which, in turn, has resulted in a user friendly, menu-driven Graphical User Interface. This system

simplifies the upholding of large amount of data and the speed of processing is off the capacity of any manual system. It minimizes the time and efforts of the user with efficiency. It is possible at any point of time to extend the software to stand at ceremony. This project is developed in such a way that any implementation or extension can be done easily.

1.2. PROBLEMS STATEMENT

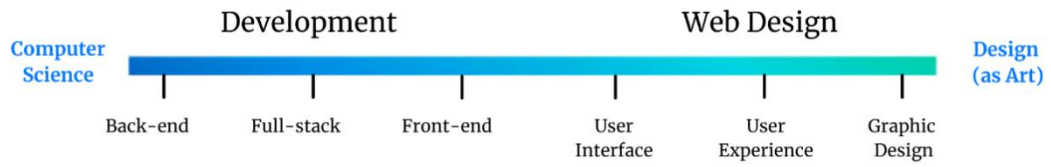
The identified problem revolves around the manual and inefficient process of hostel registration in universities. The current system requires students to physically visit the Students' Affairs Office to fill out forms, provide payment receipts, and then wait for hostel allocation. This process is not only time-consuming but also prone to errors and inefficiencies due to its reliance on pen and paper.

Moreover, the manual system leads to challenges such as:

- **Time wastage**: Both students and administrative staff spend significant time on paperwork and processing, which could be better utilized elsewhere.
- **Error-prone**: Manual data entry increases the likelihood of errors, leading to inaccuracies in student records and hostel allocations.
- **Limited scalability**: As the university population grows, the manual system becomes increasingly cumbersome and difficult to manage.
- 4 .**Inefficiency**: With the manual process, there's a lack of real-time updates and tracking, making it challenging to manage hostel occupancy effectively.
- **Resource consumption**: The manual system requires physical storage space for paperwork and additional manpower for data entry and processing.
- Overall, the identified problem is the inefficiency and ineffectiveness of the manual hostel registration process, which negatively impacts both students and administrative staff.

1.3. Web Design and Development

Web design and development is an umbrella term that describes the process of creating a website.



1.3.1. Web Design

Web design governs everything involved with the visual aesthetics and usability of a website color scheme, layout, information flow, and everything else related to the visual aspects of the UI/UX (user interface and user experience). Some common skills and tools that distinguish the web designer from the web developer are:

- Adobe Creative Suite (e.g., Photoshop, Illustrator) or other design software
- Graphic design
- UI design
- UX design
- Logo design
- Layout/format
- Placing call-to-action buttons
- Branding
- Wireframes, mock-ups, and storyboards
- Color palettes
- Typography

Web design is concerned with what the user actually sees on their computer screen or mobile device, and less so about the mechanisms beneath the surface that make it all work. Through the use of color, images, typography and layout, they bring a digital experience to life.

1.3.2. Web Development

Web development governs all the code that makes a website tick. It can be split into two categories—front-end and back-end. The front-end or client-side of an

application is the code responsible for determining how the website will actually display the designs mocked up by a designer. The back-end or server-side of an application is responsible for managing data within the database and serving that data to the front-end to be displayed. As you may have guessed, it's the front-end developer's job that tends to share the most overlap with the web designer. Some common skills and tools traditionally viewed as unique to the front-end developer are listed below:

- HTML/CSS/JavaScript
- CSS preprocessors (i.e., LESS or Sass)
- Frameworks (i.e., AngularJS, ReactJS, Ember)
- Web template design
- Libraries (i.e., jQuery)
- Git and GitHub
- On-site search engine optimization (SEO)

Front-end web developers don't usually create mock-ups, select typography, or pick color palettes—these are usually provided by the designer. It's the developer's job to bring those mock-ups to life. That said, understanding what the designer wants requires some knowledge of best practices in UI/UX design so that the developer is able to choose the right technology to deliver the desired look and feel and experience in the final product.

Back-end developers handle the business logic and data management on the back-end of an application. They write the APIs and routing that allow data to flow between the front and back end of an application. Programming languages and tools unique to back-end developers are listed below:

- Server-side programming languages (e.g., PHP, Python, Java, C#)
- Server-side web development frameworks (e.g., Ruby on Rails, Symfony, .NET)
- Database management systems (e.g., MySQL, MongoDB, PostgreSQL)
- RESTful APIs
- Authentication and security (e.g., OAuth, PassportJS)
- Servers (e.g., Linux, Apache, Express)

Web developers who possess a working knowledge across the frontend and backend of a technology stack are called full-stack developers

.CHAPTER 2
PROJECT DESCRIPTION

2.1. AIM AND OBJECTIVES

The aim and objectives of addressing the identified problem with the manual hostel registration process in universities could be outlined as follows:

Aim:

To streamline and automate the hostel registration process in universities, enhancing efficiency, accuracy, and user satisfaction.

Objectives:

- **Implement Automation**: Develop and implement a Hostel Management System to automate the registration process, reducing reliance on manual paperwork.
- **Improve Efficiency**: Reduce the time taken for hostel registration by implementing an efficient digital system that minimizes bureaucratic delays.
- **Enhance Accuracy**: Ensure accurate data entry and hostel allocation by eliminating manual errors associated with pen-and-paper processes.
- **Enhance User Experience**: Improve the experience for both students and administrative staff by providing a user-friendly interface for hostel registration and management.
- **Real-time Tracking**: Enable real-time tracking of hostel occupancy and availability to facilitate better decision-making and resource allocation.

- **Scalability**: Develop a system that can easily scale with the university's growing population and evolving needs, accommodating increased demand for hostel facilities.
- **Cost Reduction**: Reduce administrative costs associated with manual data entry, storage, and processing by transitioning to a digital system.
- **Ensure Data Security**:Implement robust security measures to safeguard student information and ensure compliance with data protection regulations.

2.2. SCOPE OF THE PROJECT

The scope of the project involves various aspects related to the automation and improvement of the hostel registration process in universities. Here are the key components within the scope:

- **Hostel Management System Development:** The development of a comprehensive software solution tailored to automate the entire hostel registration process. This includes features such as student registration, payment processing, room allocation, and data management.
- **User Interface Design**: Designing a user-friendly interface for both students and administrative staff to interact with the Hostel Management System. The interface should be intuitive, easy to navigate, and visually appealing.
- **Database Management:** Setting up and managing a database to store student information, payment records, room allocations, and other relevant data securely. The database should be scalable to accommodate the growing number of students and their information.
- **Integration with Existing Systems:** Ensuring seamless integration of the Hostel Management System with other university systems such as student databases, financial systems, and communication platforms to facilitate data exchange and process synchronization.
- **Security Measures:** Implementing robust security measures to protect student data and ensure compliance with data protection regulations. This includes encryption of sensitive information, access controls, and regular security audits.
- **Training and Support:** Providing training sessions and user manuals for students and administrative staff to familiarize them with the new system. Additionally, offering ongoing technical support and troubleshooting services to address any issues or questions that may arise.
- **Pilot Testing and Feedback**: Conducting pilot tests of the Hostel Management System with a small group of users to identify any potential

issues or areas for improvement. Gathering feedback from users and incorporating necessary changes before full-scale implementation.

- **Scalability and Future Enhancements:** Designing the system with scalability in mind to accommodate future growth and technological advancements. Additionally, planning for future enhancements and updates based on feedback from users and evolving needs.

2.3. IMPORTANCE OF THE PROJECT

The importance of the project lies in its potential to significantly enhance the efficiency, accuracy, and overall experience of the hostel registration process in universities. Here are some key reasons why the project is important:

- **Efficiency Improvement:** By automating the hostel registration process, the project aims to reduce bureaucratic delays, paperwork, and manual labor involved in managing student accommodation. This efficiency improvement saves time for both students and administrative staff, allowing them to focus on more productive tasks**.**
- **Enhanced Accuracy:** Automation reduces the risk of human errors associated with manual data entry, leading to more accurate records and room allocations. This helps prevent issues such as double bookings, incorrect payments, and misplaced paperwork, ultimately improving the reliability of the hostel management system
- **Improved User Experience:** Implementing a user-friendly interface and streamlined processes makes the hostel registration experience more convenient and intuitive for students. They can easily access information, make payments, and manage their accommodation preferences, leading to higher satisfaction levels.
- **Optimized Resource Allocation:** Real-time tracking of hostel occupancy and availability enables better resource allocation and planning. University administrators can make data-driven decisions regarding room allocations, facility maintenance, and future infrastructure investments based on accurate occupancy data.
- **Cost Savings:** By reducing manual labor, paperwork, and administrative overheads, the project can result in cost savings for the university. This

includes savings on staff hours, printing costs, storage space, and potential penalties or fines associated with errors in the manual process.

- **Scalability and Adaptability:** A well-designed hostel management system can easily scale to accommodate the growing needs of the university and adapt to changes in student demographics, regulations, and technological advancements over time. This ensures that the system remains effective and relevant in the long term.
- **Data Security and Compliance:** Implementing robust security measures ensures the protection of sensitive student information and compliance with data protection regulations. This helps build trust among students and stakeholders and mitigates the risk of data breaches or unauthorized access.

CHAPTER 3

SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

In the existing system is very difficult to retrieve the information, there are hotel details, room availability detail and food details. The information generated by various transactions takes time and efforts to be stored at right place. Various changes to information like customer details are difficult to make as paper work is involved. Manual calculations are error prone and take a lot of time this may result in incorrect information. This becomes a difficult task as information is difficult to collect from various registers.

3.1.1. Disadvantages

- Difficult to clarify the doubts.
- Time Consuming.
- More manual work required.
- The manual error may occur severely in the exist

3.2. PROPOSED SYSTEM

In the proposed system is maintaining hostel information, booking rooms and food order through online. This would assure economic use of storage space and consistency in the data stored. The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires. In manual system there are many problems to store the largest amount of information. In this system, we can easily manage there hostel details, students records, room details, mess expenditure, mess bill calculation, easy way of allocation and hostel attendance.

3.2.1. ADVANATGES

- This system maintains user's personal info, address, and contact details.
- This system makes the overall project management much easier and flexible.
- Authentication is provided for this application.

CHAPTER 4

SYSTEM SPECIFICATION

4.1. Hardware Requirements:

- Processors: Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM
- Disk space: 320 GB
- Operating systems: Windows® 10, macOS*, and Linux*

4.2. Software Requirements:

- Server Side : PHP
- Client Side : HTML, CSS, Bootstrap
- Back end : MySQL 5.
- Server : WampServer 2i
- BC DLL : pyBlock, pyenv, pyFHE

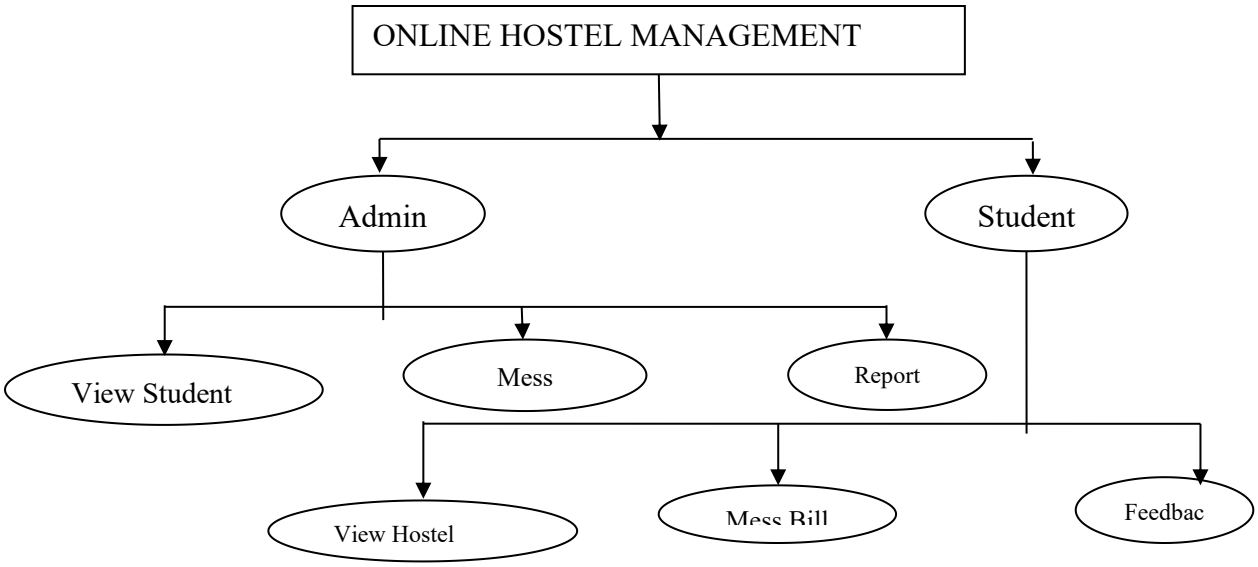
CHAPTER 5
SYSTEM DESIGN

5.1. SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

Various organizations define systems architecture in different ways, including:

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software
- The composite of the design architectures for products and their life-cycle processes.

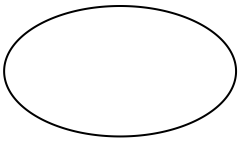


5.1. Fig 1: Online Hostel Management Diagram

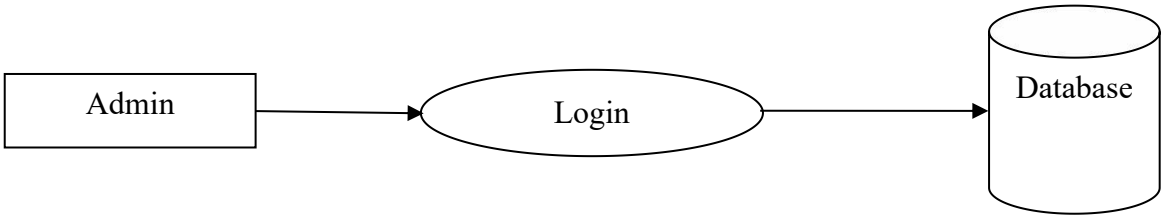
5.2. DATA FLOW DIAGRAM

A two-dimensional diagram that explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must (1) identify external inputs and outputs, (2) determine how the inputs and outputs relate to each other, and (3) explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

5.1. Data flow Symbols:

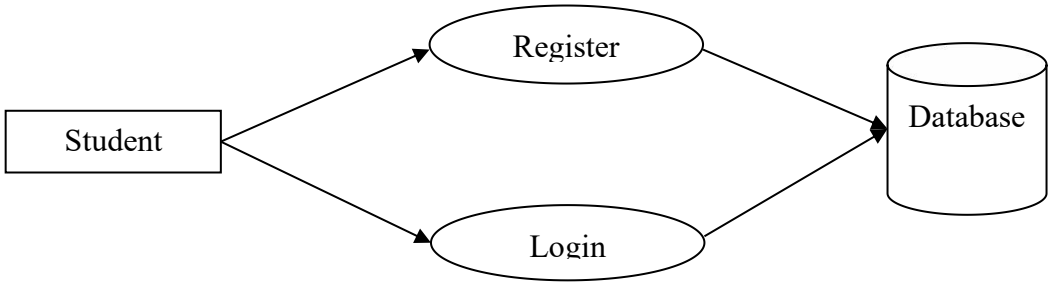
Symbol	Description
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.

5.2.1. LEVEL 0



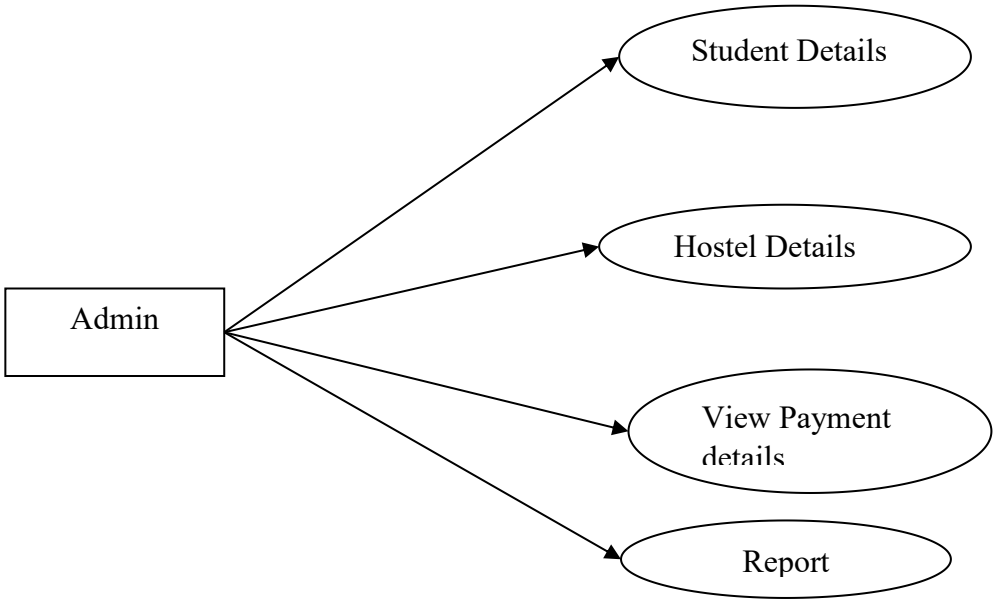
5.2. Fig 2: Data Flow Level Zero Diagram

5.2.2. LEVEL 1



5.3. Fig 3: Data Flow Level One Diagram

5.2.3. LEVEL 3



5.4. Fig 4: Data Flow Level Two Diagram

5.3. DATABASE DESIGN

5.2. Table Name: register

Field Name	Datatype	Size	DESCRIPTION	Constraint
Id	Int	11	ID	Primary Key
Name	varchar	50	NAME	NOT NULL
Father Name	varchar	50	FATHER NAME	NOT NULL
Gender	varchar	30	GENDER	NOT NULL
Date of Birth	varchar	30	DATE	NOT NULL
Register Number	Int	11	REGISTER NUMBER	NOT NULL
Mobile Number	Int	11	MOBILE NUMBER	NOT NULL
Email ID	varchar	30	EMAIL ID	NOT NULL
User Name	varchar	30	USER NAME	NOT NULL
Password	varchar	30	PASSWORD	NOT NULL
Confirm Password	varchar	30	CONFIRM PASSWORD	NOT NULL

5.3 Table Name: Mess Bill

Field Name	Datatype	Size	DESCRIPTION	Constraint
Id	Int	11	ID	Primary Key
Register Number	varchar	50	REGISTER NUMBER	NOT NULL
Attendance Report	varchar	50	ATTENDANCE REPORT	NOT NULL
Bill	varchar	30	BILL	NOT NULL

5.4. Table Name: Room Allocate

Field Name	Datatype	Size	DESCRIPTION	Constraint
Id	Int	11	ID	Primary Key
Register Number	varchar	50	REGISTER NUMBER	NOT NULL
Student Name	varchar	50	STUDENT NAME	NOT NULL
Room Number	varchar	30	ROOM NUMBER	NOT NULL

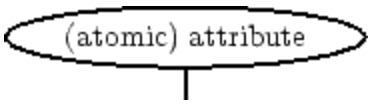
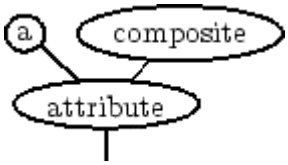
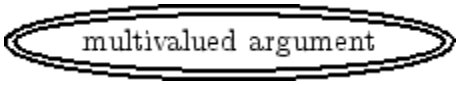
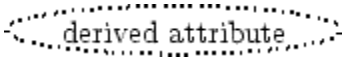
5.5. Table Name: Feedback

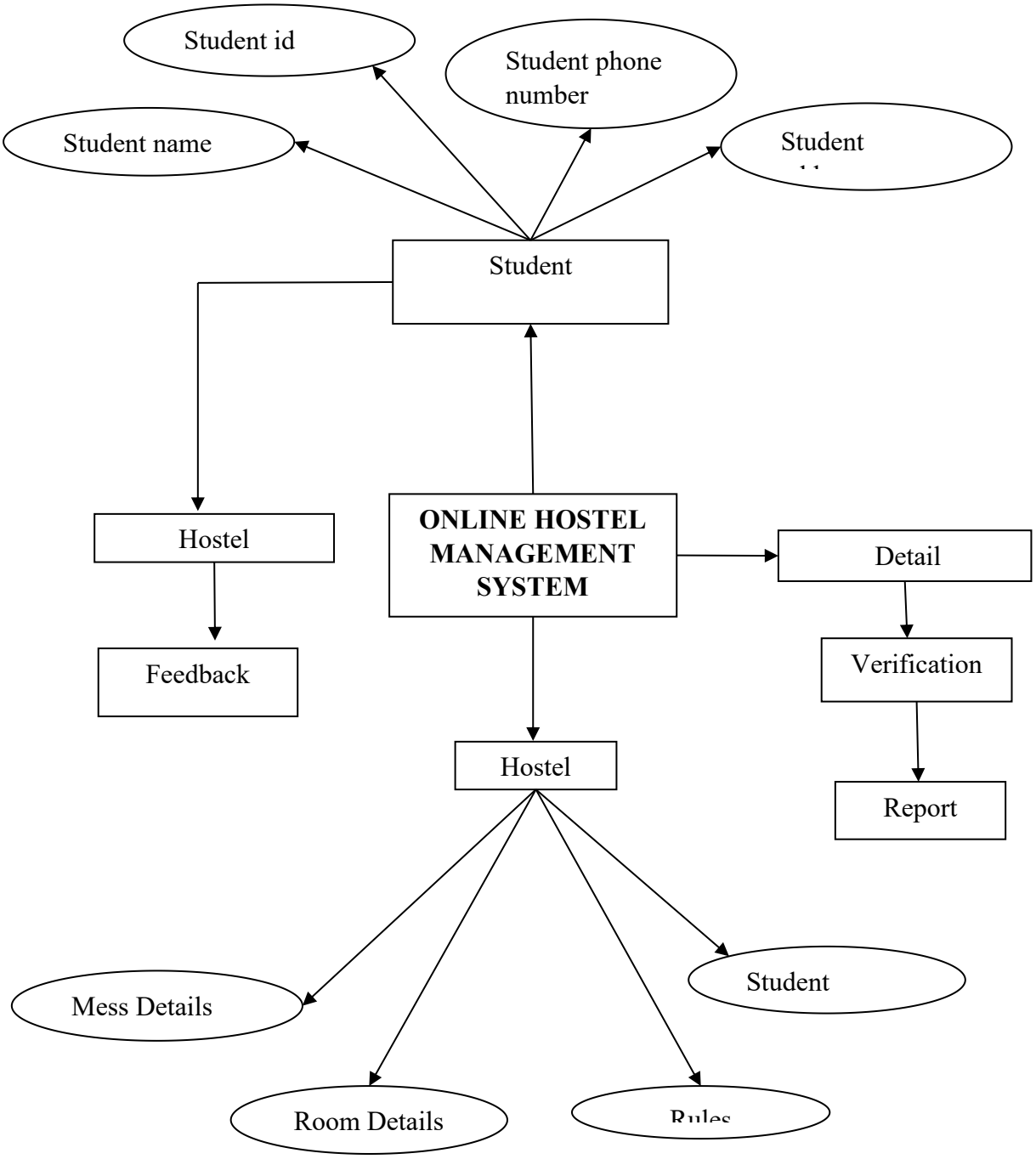
Field Name	Datatype	Size	DESCRIPTION	Constraint
Id	Int	11	ID	Primary Key
Register Number	varchar	50	REGISTER NUMBER	NOT NULL
Feedback	varchar	50	FEEDBACK	NOT NULL

5.4. Entity – Relationship Diagram

An entity–relationship model is the result of using a systematic process to describe and define a subject area of business data. It does not define business process; only visualize business data. The data is represented as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. The three schema approach to software engineering uses three levels of ER models that may be developed.

5.6. ER Diagram Symbol:

S.NO.	Diagram	Types
1.		Entity types
2.		Relationship Types
3.		Atomic attribute types
4.		Composite attribute types
5.		Multi valued attribute types
6.		Derived Attribute types



5.6. Fig 5: Online Hostel Manageent System Diagram

CHAPTER 6
SODTWARE DESCRIPTION

6.1. PHP

PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994 the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

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 specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

PHP - Overview

PHP is a recursive acronym for "PHP: Hypertext Preprocessor". PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically

used for developing web based software applications. This tutorial helps you to build your base with PHP.

PHP Objects

Basic object-oriented programming functionality was added in PHP 3 and improved in PHP 4. This allowed for PHP to gain further abstraction, making creative tasks easier for programmers using the language. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled like value types. The drawback of this method was that code had to make heavy use of PHP's "reference" variables if it wanted to modify an object it was passed rather than creating a copy of it. In the new approach, objects are referenced by handle, and not by value.

Implementations

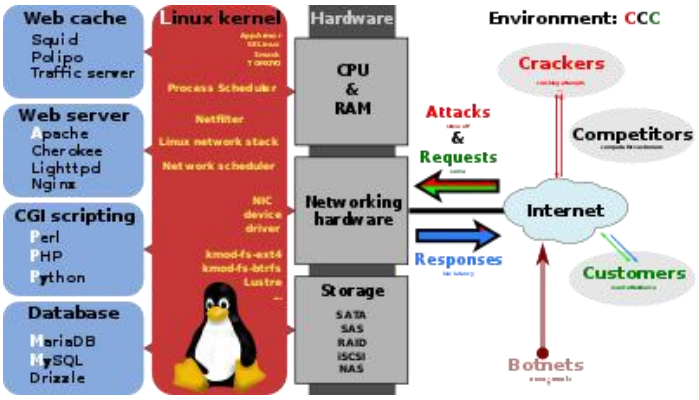
The only complete PHP implementation is the original, known simply as PHP. It is the most widely used and is powered by the Zend Engine. To disambiguate it from other implementations, it is sometimes unofficially called "Zend PHP". The Zend Engine compiles PHP source code on-the-fly into an internal format that it can execute, thus it works as an interpreter. It is also the "reference implementation" of PHP, as PHP has no formal specification, and so the semantics of Zend PHP define the semantics of PHP. Due to the complex and nuanced semantics of PHP, defined by how Zend works, it is difficult for competing implementations to offer complete compatibility.

Licensing

PHP is free software released under the PHP License, which stipulates that: Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo". This restriction on use of "PHP" makes the PHP License incompatible with the General Public License (GPL), while the Zend

License is incompatible due to an advertising clause similar to that of the original BSD license.

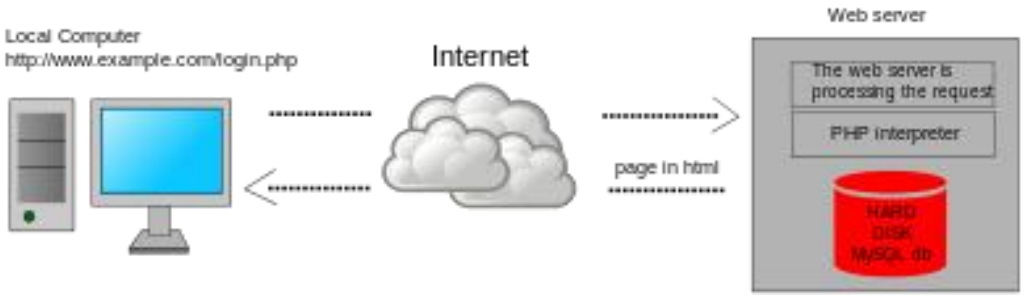
Use



6.1. Fig 6: BSD Licence

A broad overview of the LAMP software bundle, displayed here together with Squid

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). Most web hosting providers support PHP for use by their clients. 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.



6.2. Fig 7: Database System

PHP acts primarily as a filter taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data. Most commonly the output will be HTML, although it could be JSON, XML or binary data such as image or audio formats. Since PHP 4, the PHP parser compiles input to produce bytecode for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's ASP.NET, Sun Microsystems' Java Server Pages, and mod_perl. PHP has also attracted the development of many software frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include PRADO, CakePHP, Symfony, CodeIgniter, Laravel, Yii Framework, Phalcon and Zend Framework, offering features similar to other web frameworks.

The LAMP architecture has become popular in the web industry as a way of deploying web applications. PHP is commonly used as the P in this bundle alongside Linux, Apache and MySQL, although the P may also refer to Python, Perl, or some mix of the three. Similar packages, WAMP and MAMP, are also available for Windows and macOS, with the first letter standing for the respective operating system. Although both PHP and Apache are provided as part of the macOS base install, users of these packages seek a simpler installation mechanism that can be more easily kept up to date.

As of April 2007, over 20 million Internet domains had web services hosted on servers with PHP installed and mod_php was recorded as the most popular Apache HTTP Server module. As of August 2019, PHP was used as the server-side programming language on 79.1% of websites, down from 83.5% previously, where the language could be determined. Web content management systems written in PHP include MediaWiki, Joomla, eZ Publish, eZ Platform, SilverStripe, WordPress, Drupal, and Moodle. Websites written in PHP, in back-end and/or user-facing portion, include Facebook, Digg, Tumblr, Dailymotion, and Slack.

For specific and more advanced usage scenarios, PHP offers a well-defined and documented way for writing custom extensions in C or C++. Besides extending the language itself in form of additional libraries, extensions are providing a way for improving execution speed where it is critical and there is room for improvements by using a true compiled language. PHP also offers well defined ways for embedding itself into other software projects. That way PHP can be easily used as an internal scripting language for another project, also providing tight interfacing with the project's specific internal data structures. PHP received mixed reviews due to lacking support for multithreading at the core language level, though using threads is made possible by the "pthreads" PECL extension. As of January 2013, PHP was used in more than 240 million websites (39% of those sampled) and was installed on 2.1 million web servers. A command line interface, php-cli, and two ActiveX Windows Script Host scripting engines for PHP have been produced. As of 2019, PHP 5 is most used on the web; which was last updated with security updates in January 2019, with PHP 5.6.40.

Why to Learn PHP?

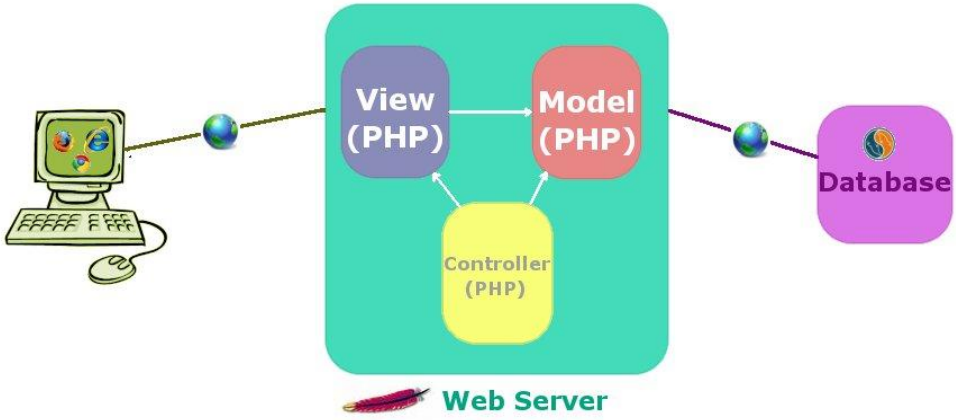
PHP started out as a small open source project that evolved as more and more people found out how useful it was. RasmusLerdorf unleashed the first version of PHP way back in 1994.

PHP is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning PHP:

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started,

executes even very complex queries with huge result sets in record-setting time.

- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.



6.3. Fig 8: Basic View of PHP

Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

Hello World using PHP.

Just to give you a little excitement about PHP, I'm going to give you a small conventional PHP Hello World program, You can try it using Demo link.

<html>

<head>

<title>Hello World</title>

</head>

<body>

<?php echo "Hello, World!";?>

</body></html>

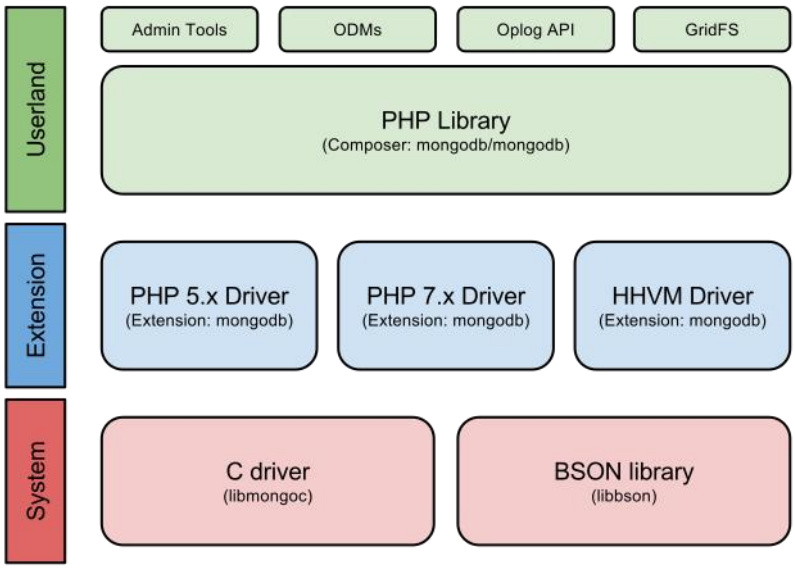
Applications of PHP

1.1.1.1.1.1.1.1.1 As mentioned before, PHP is one of the most widely used language over the web. I'm going to list few of them here:

1.1.1.1.1.1.1.1.2 PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.andcan handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.You add, delete, modify elements within your database through PHP and access cookies variables and set cookies.Using PHP, you can restrict users to access some pages of your website and encrypt data.

1.1.1.1.1.1.1.1.3 Architecture Overview

1.1.1.1.1.1.1.1.4 This section explains how all the different parts of the driver fit together. From the different language runtimes, through the extension and to the PHP libraries on top. This new architecture has replaced the old mongo extension. We refer to the new one as the *mongodb* extension.



6.4. Fig 9: Overview of PHP

At the top of this stack sits a pure » PHP library, which we will distribute as a Composer package. This library will provide an API similar to what users have come to expect from the old mongo driver (e.g. CRUD methods, database and collection objects, command helpers) and we expect it to be a common dependency for most applications built with MongoDB. This library will also implement common » specifications, in the interest of improving API consistency across all of the » drivers maintained by MongoDB (and hopefully some community drivers, too).Sitting below that library we have the lower level driver. This extension will effectively form the glue between PHP and our system libraries. This extension will expose an identical public API for the most essential and performance-sensitive functionality:

- Connection management
- BSON encoding and decoding
- Object document serialization (to support ODM libraries)
- Executing commands and write operations
- Handling queries and cursors

Prerequisites

1.1.1.1.1.1.1.1.5 Before proceeding with this tutorial you should have at least basic understanding of computer programming, Internet, Database, and MySQL etc is very helpful.

PHP started out as a small open source project that evolved as more and more people found out how useful it was. RasmusLerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

Common uses of PHP

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.Access cookies variables and set cookies. Using PHP, you can restrict users to access some pages of your website. It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

1.1.1.1.1.1.1.1.6 In order to develop and run PHP Web pages three vital components need to be installed on your computer system.

- **Web Server** − PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server. Download Apache for free here − https://httpd.apache.org/download.cgi
- **Database** − PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database. Download MySQL for free here − https://www.mysql.com/downloads/
- **PHP Parser** − In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web Browser. This tutorial will guide you how to install PHP parser on your computer.

PHP Parser Installation

1.1.1.1.1.1.1.1.7 Before you proceed it is important to make sure that you have proper environment setup on your machine to develop your web programs using PHP.

1.1.1.1.1.1.1.1.8 Type the following address into your browser's address box.

http://127.0.0.1/info.php

1.1.1.1.1.1.1.1.9 If this displays a page showing your PHP installation related information then it means you have PHP and Webserver installed properly. Otherwise you have to follow given procedure to install PHP on your computer.

1.1.1.1.1.1.1.1.10 This section will guide you to install and configure PHP over the following four platforms −

- PHP Installation on Linux or Unix with Apache
- PHP Installation on Mac OS X with Apache
- PHP Installation on Windows NT/2000/XP with IIS
- PHP Installation on Windows NT/2000/XP with Apache

Apache Configuration

1.1.1.1.1.1.1.1.11 If you are using Apache as a Web Server then this section will guide you to edit Apache Configuration Files.

1.1.1.1.1.1.1.1.12 Just Check it here − PHP Configuration in Apache Server

PHP.INI File Configuration

1.1.1.1.1.1.1.1.13 The PHP configuration file, php.ini, is the final and most immediate way to affect PHP's functionality.

1.1.1.1.1.1.1.1.14 Just Check it here − PHP.INI File Configuration

Windows IIS Configuration

1.1.1.1.1.1.1.1.15 To configure IIS on your Windows machine you can refer your IIS Reference Manual shipped along with IIS.

1.1.1.1.1.1.1.1.16 The main way to store information in the middle of a PHP program is by using a variable.

1.1.1.1.1.1.1.1.17 Here are the most important things to know about variables in PHP.

- All variables in PHP are denoted with a leading dollar sign ($).
- The value of a variable is the value of its most recent assignment.
- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.

- Variables in PHP do not have intrinsic types - a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP does a good job of automatically converting types from one to another when necessary.
- PHP variables are Perl-like.

1.1.1.1.1.1.1.18 PHP has a total of eight data types which we use to construct our variables −

- **Integers** − are whole numbers, without a decimal point, like 4195.
- **Doubles** − are floating-point numbers, like 3.14159 or 49.1.
- **Booleans** − have only two possible values either true or false.
- **NULL** − is a special type that only has one value: NULL.
- **Strings** − are sequences of characters, like 'PHP supports string operations.'
- **Arrays** − are named and indexed collections of other values.
- **Objects** − are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources** − are special variables that hold references to resources external to PHP (such as database connections)

6.2. MYSQL

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals. MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

6.3. WAMPSERVER

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.

WAMPServer is a reliable web development software program that lets you create web apps with MYSQL database and PHP Apache2. With an intuitive interface, the application features numerous functionalities and makes it the preferred choice of developers from around the world. The software is free to use and doesn't require a payment or subscription.

4.4. BOOTSTRAP 4

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap

Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

Mobile-first approach: In Bootstrap, mobile-first styles are part of the core framework

Browser compatibility: Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera)

CHAPTER 7
SYSTEM IMPLEMENTATION

7.1. SYSTEM DESCRIPTION

Introduction

The Hostel Management System is a comprehensive software solution designed to streamline and automate the management of hostel facilities. It caters to the needs of hostel administrators, staff, and residents by providing efficient management of room allocation, resident details, billing, inventory, and other hostel-related operations.

Key Features

- User Management: The system allows administrators to manage user accounts, including staff members and residents. It supports functionalities such as user registration, login, profile management, and password reset.
- Room Management: Administrators can manage hostel rooms, including allocation, vacancy status, and room transfers. Residents can view available rooms, submit room transfer requests, and check their room assignments.
- Billing and Payments: The system facilitates billing and payment processing for hostel residents. It generates invoices for accommodation charges, tracks payment status, and sends payment reminders. It supports various payment methods and generates financial reports.
- Inventory Management: Administrators can manage hostel inventory, including stock tracking, replenishment, and item checkout. Residents can request items from the inventory and view their inventory usage.
- Reporting: The system provides reporting functionalities to generate various reports related to room occupancy, billing, inventory usage, and other hostel metrics. Reports can be exported in different formats for analysis and decision-making.
- Security: The system ensures data security and access control through user authentication and authorization mechanisms. It encrypts sensitive information and implements security best practices to protect against unauthorized access.

User Roles

- Administrator: Responsible for overall management of the hostel, including user management, room allocation, billing, inventory management, and reporting.
- Staff: Assist administrators in day-to-day hostel operations, including room allocation, billing, inventory management, and resident assistance.
- Residents: Stay in the hostel and utilize the system for room allocation, billing inquiries, inventory requests, and other hostel-related activities.

System Architecture

- The Hostel Management System is built using a web-based architecture, allowing access from any device with an internet connection.
- It is developed using PHP for server-side scripting, MySQL for the database, and HTML/CSS/JavaScript for the front-end user interface.
- The system follows a client-server architecture, with the web server hosting the application and serving requests from clients' web browsers.
- It incorporates secure authentication mechanisms, data encryption, and role-based access control to ensure system security.

Benefits

- Efficiency: Automation of manual tasks such as room allocation, billing, and inventory management improves operational efficiency.
- Accuracy: Centralized data management reduces errors and inconsistencies in hostel-related information.
- Transparency: Residents have access to their accommodation details, billing information, and inventory usage, promoting transparency and accountability.
- Cost Savings: Streamlined processes and reduced paperwork lead to cost savings in administrative overhead.
- Improved Communication: The system facilitates communication between administrators, staff, and residents through notifications, alerts, and messaging features.

Conclusion

The Hostel Management System offers a robust and efficient solution for managing hostel facilities, catering to the needs of administrators, staff, and residents. Its comprehensive features, user-friendly interface, and secure architecture make it an indispensable tool for hostel management.

7.2. SYSTEM FLOW

User Registration and Login

- New users register with the system by providing necessary details.
- Registered users log in using their credentials (username and password).

Room Booking

- Users select their desired room type and duration of stay.
- The system checks room availability and confirms the booking if a room is available.
- Confirmation details are provided to the user.

Check-in Process

- Users arrive at the hostel for check-in.
- Hostel staff verify the booking details and allocate a room to the user.
- Users receive keys or access cards for their allocated rooms.

Room Management

- Hostel staff manage room allocations and vacancies.
- Users can request room changes or extensions through the system.
- Cleaning and maintenance schedules are managed through the system.

Billing and Payments

- The system generates bills based on the user's stay duration and any additional services used (e.g., laundry, meals).
- Users can view and pay bills through the system.
- Payment confirmation is provided to the user.

Inventory Management

- Hostel staff manage inventory of items (e.g., toiletries, bedding) through the system.

- Stock levels are monitored, and reordering is done when necessary.
- Users can request items through the system.

Visitor Management

- Users can register their visitors through the system.
- Hostel staff verify visitor registrations and issue visitor passes.
- Visitor check-in and check-out times are recorded.

Security

- The system manages access control to hostel premises and rooms.
- Security incidents and emergencies are logged and managed through the system.
- CCTV cameras and other security measures may be integrated with the system.

Reporting

- The system generates reports on occupancy rates, revenue, inventory levels, etc.
- Reports are used for decision-making and monitoring hostel performance.

Check-out Process

- Users check out of their rooms at the end of their stay.
- Hostel staff verify room condition and check for any damages.
- Final bills are generated and settled, and keys or access cards are returned.

Feedback and Reviews

- Users can provide feedback and reviews of their stay through the system.
- Feedback is used to improve hostel services and operations.

System Administration

- Admins manage user accounts, permissions, and system settings.
- They oversee the overall functioning of the system and address any technical issues.

Data Security and Backup

- The system ensures data security through encryption and access controls.
- Regular backups of the data are taken to prevent data loss.

7.3. MODULES DESCRIPTION

Modules:

Student

- Registration
- Login
- Hostel Details
- View Mess Bill
- View Room
- Feedback

Admin

- Login
- Student Admission
- Room Allocate
- Mess Bill Preparation
- Student Report
- To receive the staff permission

Staff

- Register
- Login
- Given permission to student

Student

- **Registration**

The registration page is useful for the new student to register themselves by giving their valid details such as e-mail id, user name, and phone number. The user has to fill all the details else message is displayed to the user. Once all the fields are filled the user clicks the Register button, which submits the data to the database.

- **Login**

Login modules can be user implemented it allowed the register person only. We have to use this module in security purpose related the details.

- **Hostel Details**

A Hostel large room with separate beds, a shared bathroom, and a communal kitchen. Some hostels have private rooms, but the lower-cost ones generally offer bunk beds. But they've grown in popularity and you can find them.

- **View Mess Bill**

Administrator is responsible for providing the all bills. The bills provided by administrator are checked by secretary.

- **Feedback**

The user can provide their feedback about the entire process whether it may be positive or negative and the feedback can be viewed by the admin.

Staff

- **Registration**

The registration page it's for the new staff to register themselves by giving their valid details such as e-mail id, user name, and phone number. The user has to fill all the details else message is displayed to the user. Once all the fields are filled the user clicks the Register button, which submits the data to the database.

- **Login**

This module describes the details of login. In this module the parents have an own unique login id and password. Login is used for protect the unauthorized access.

- **Given permission to student**

Staff given permissions, by the student, allow to meet, discuss about topic or class are a record of standing permissions from other terms. Institutions staff given approve to send warden. Selected permissions student to meet staff that are a record store in data set

Admin

- **Login**:

This module describes the details of login. In this module the admin and admin have an own unique login id and password. Login is used for protect the unauthorized access.

- **Student Record:**

Students have to provide their needed information like name, reg no, attendance and so on to obtain username and password with which the student can login and perform necessary actions that has to done throughout the hostel management system.

- **Mess Bill Preparation:**

Admin providing the all bills. Mess bill i for the food you will eat in the hostel whereas the hostel fees

- **Allotting Room:**

Admin provide the room number for all user. Using this room number user can also check status of the room details.

- **Receive the staff permission**

Staff given permissions, by the student, allow to meet, discuss about topic or class are a record of standing permissions from other terms. Institutions staff given approve to send warden. Selected permissions student to meet staff that are a store record in data set

SOURCECODE

Packages

```
<?php
$rdate=date("Y/m/d");
if (isset($_POST['signup-submit'])) {
  require 'config.inc.php';
  $idno = $_POST['idno'];
  $Fname = $_POST['Fname'];
  $Lname = $_POST['Lname'];
  $Mob_no = $_POST['Mob_no'];
  $Dept = $_POST['Dept'];
  $Pwd = $_POST['Pwd'];
  $i="insert into staff(idno,Fname,Lname,Mob_no,Dept,Pwd,rdate)value('$idno','$Fname','$Lname','$Mob_no','$Dept','$Pwd','$rdate')";
   $qry=mysqli_query($conn, $i);
        if($qry)
                                {
                                //header("location:addhearingdate.php");
            ?>
            <script language="javascript">
                alert("Added Successfully");
                window.location.href="sregister.php";
                </script>
                <?php
```

```
}
else
{
$msg="Could not Registered";
}

}

?>
```

STUDENT HOME

```
<?php
session_start();
$student_roll_no=$_SESSION['student_roll_no'];
 require 'config.inc.php';
?>
<!DOCTYPE html>
<html lang="en">
<head>
<title>HMS</title>

<!-- Meta tag Keywords -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta charset="utf-8">
<meta name="keywords" content="Intrend Responsive web template, Bootstrap Web Templates, Flat Web Templates, Android Compatible web template,
```

```
Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web design" />
<script type="application/x-javascript">
    addEventListener("load", function () {
        setTimeout(hideURLbar, 0);
    }, false);

    function hideURLbar() {
        window.scrollTo(0, 1);
    }
</script>
<!--// Meta tag Keywords -->

<link href="web_home/css_home/slider.css" type="text/css" rel="stylesheet" media="all">

<!-- css files -->
<link rel="stylesheet" href="web_home/css_home/bootstrap.css"> <!-- Bootstrap-Core-CSS -->
<link rel="stylesheet" href="web_home/css_home/style.css" type="text/css" media="all"/> <!-- Style-CSS -->
<link rel="stylesheet" href="web_home/css_home/fontawesome-all.css"> <!-- Font-Awesome-Icons-CSS -->
<!-- //css files -->

<!-- testimonials css -->
```

```
        <link rel="stylesheet" href="web_home/css_home/flexslider.css" type="text/css" media="screen" property="" /><!-- flexslider css -->

        <!-- //testimonials css -->


        <!-- web-fonts -->

        <link href="//fonts.googleapis.com/css?family=Poiret+One&amp;subset=cyrillic,latin-ext" rel="stylesheet">

        <!-- //web-fonts -->


</head>


<body>


<!-- banner -->

        <div class="banner" id="home">

                <div class="cd-radial-slider-wrapper">


<!--Header-->

<header>

        <div class="container agile-banner_nav">

                <nav class="navbar navbar-expand-lg navbar-light bg-light">


                        <h1><a class="navbar-brand" href="home.php">HMS <span class="display"></span></a></h1>
```

```
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse justify-content-center" id="navbarSupportedContent">
<ul class="navbar-nav ml-auto">
<li class="nav-item active">
<a class="nav-link" href="home.php">Home <span class="sr-only">(current)</span></a>
</li>
<li class="nav-item">
<a class="nav-link" href="services.php">Hostels</a>
</li>
<li class="nav-item">
<a class="nav-link" href="contact.php">Contact</a>
</li>
<li class="nav-item">
<a class="nav-link" href="message_user.php">Message Received</a>
</li>
<li class="nav-item active">
```

```
                                        <a class="nav-link"
href="pay.php">Bill</a>
                                    </li>
                                <li class="dropdown nav-item">
                                    <a href="#" class="dropdown-toggle
nav-link" data-toggle="dropdown"><?php echo $student_roll_no;?>
                                        <b class="caret"></b>
                                    </a>
                                    <ul class="dropdown-menu
agile_short_dropdown">
                                        <li>
                                            <a
href="profile.php">My Profile</a>
                                        </li>
                                        <li>
                                            <a
href="includes/logout.inc.php">Logout</a>
                                        </li>
                                    </ul>
                                </li>
                            </ul>
                        </div>
                    </nav>
                </div>
            </header>
            <!--Header-->
```

```
<ul class="cd-radial-slider" data-radius1="60" data-radius2="1364" data-centerx1="110" data-centerx2="1290">
<li class="visible">
<div class="svg-wrapper">
<svg viewBox="0 0 1400 800">
<title>Animated SVG</title>
<defs>
<clipPath id="cd-image-1">
<circle id="cd-circle-1" cx="110" cy="400" r="1364"/>
</clipPath>
</defs>
<image height='800px' width="1400px" clip-path="url(#cd-image-1)" xlink:href="web_home/images/1.png"></image>
</svg>
</div> <!-- .svg-wrapper -->
<div class="cd-radial-slider-content">
<div class="wrapper">
<div class="text-center">
<h2>Hostel Monitoring </h2>
<h3> System </h3>
</div>
</div>
```

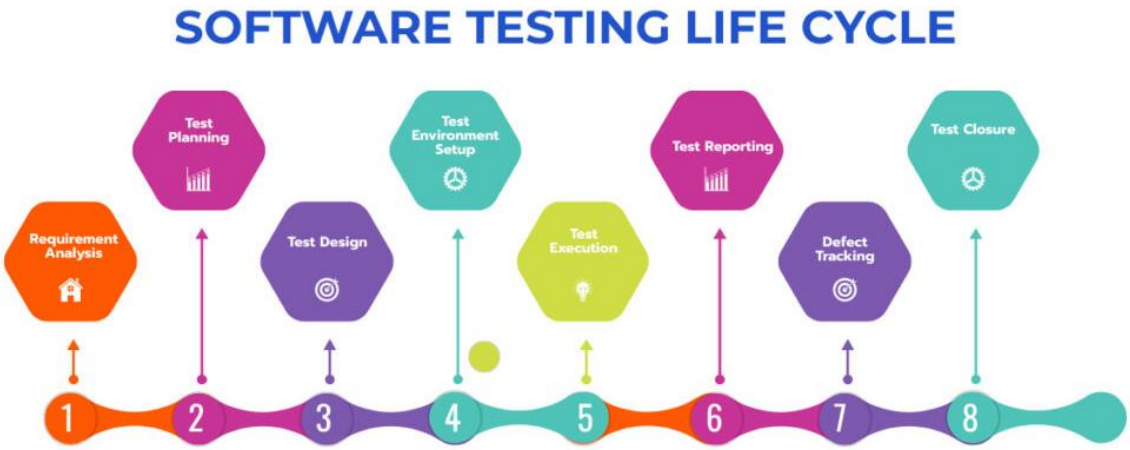
```
                                    <a class="nav-link"
href="message_hostel_manager.php">Messages Received</a>
                                </li>

                                <li class="nav-item">
                                    <a class="nav-link"
href="contact_manager.php">Contact</a>
                                </li>
                                <li class="dropdown nav-item">
                                    <a href="#" class="dropdown-toggle
nav-link" data-toggle="dropdown"><?php echo $_SESSION['username']; ?>
                                        <b class="caret"></b>
                                    </a>
                                    <ul class="dropdown-menu
agile_short_dropdown">
                                        <li>
                                            <a
href="includes/logout.inc.php">Logout</a>
                                        </li>
                                    </ul>
                                </li>
                            </ul>
                        </div>
                    </nav>
                </div>
            </header>
```

CHAPTER 8

SYSTEM TESTING

8.1. SOFTWARE TESTING

Software testing is a process of evaluating and verifying that a software application or system meets specified requirements and functions correctly. The primary goal of software testing is to identify defects or errors in the software and ensure its quality and reliability.



8.1. Fig 10: Software Testing Life Cycle

SDLC stands for Software Development Life Cycle, which is a systematic process or set of phases used in software development projects to design, develop, test, and maintain high-quality software. The SDLC aims to produce a software product that meets or exceeds customer expectations, is delivered on time and within budget, and is maintainable and scalable.

Unit Testing

- Test individual components (functions, methods, classes) of the PHP code.
- Use PHP Unit or similar testing frameworks.
- Test cases should cover all possible scenarios and edge cases.
- Verify input validation, data manipulation, and error handling.

Integration Testing

- Test the integration of PHP modules with other components (database, external APIs, etc.).

- Ensure proper communication and data exchange between modules.
- Verify compatibility and consistency of data formats.

System Testing

- Test the entire Hostel Management System as a cohesive unit.
- Validate system functionalities from end to end.
- Test user interactions, workflows, and system responses.
- Verify system behavior against requirements.

User Interface (UI) Testing

- Test the graphical user interface (GUI) elements of the system.
- Ensure proper rendering of web pages and forms.
- Validate user input handling and form submission.
- Verify accessibility and usability of the UI.

Database Testing

- Test database operations (CRUD operations, transactions, queries).
- Verify data integrity and consistency.
- Test database schema changes and migrations.
- Validate indexing and query optimization.

Security Testing

- Test for vulnerabilities such as SQL injection, XSS, CSRF, etc.
- Validate authentication and authorization mechanisms.
- Test session management and data encryption.
- Verify compliance with security best practices and standards.

Performance Testing

- Test system performance under normal and peak load conditions.
- Measure response times, throughput, and resource utilization.
- Identify performance bottlenecks and optimize code as needed.
- Use tools like Apache JMeter or Siege for load testing.

User Acceptance Testing (UAT)

- Involve stakeholders (administrators, staff, residents) in testing.
- Validate system functionalities against user requirements.
- Gather feedback and address any usability issues.
- Ensure the system meets the needs of end-users.

Regression Testing

- Re-run previously passed tests to ensure new changes haven't introduced regressions.
- Automate regression tests wherever possible to streamline testing efforts.
- Validate backward compatibility with previous versions of the system.

Continuous Integration (CI) Testing

- Integrate testing into the CI/CD pipeline.
- Automate test execution and result reporting.
- Ensure code quality and stability with each code commit.

8.2. TEST CASES

Test Case Structure

- Explanation of the structure and format of test cases.
- Components of a test case (Test ID, Description, Preconditions, Test Steps, Expected Results, Actual Results, Pass/Fail).

Unit Test Cases

Test Case 1: UserRegistrationTest

Description: Test the functionality of user registration.

Preconditions: None.

Test Steps:Provide valid user registration data.

Submit registration form.

Expected Results: User should be successfully registered and added to the database.

Actual Results: User is successfully registered.

Pass/Fail: Pass.

Test Case 2: RoomAllocationTest

Description: Test the functionality of room allocation.

Preconditions: Available rooms in the hostel.

Test Steps:Select a resident.

Choose a vacant room.

Allocate the room to the resident.

Expected Results: Room should be allocated to the resident in the database.

Actual Results: Room is successfully allocated to the resident.

Pass/Fail: Pass.

Integration Test Cases

Test Case 3: UserLoginIntegrationTest

Description: Test the integration of user login functionality.

Preconditions: User account exists in the database.

Test Steps:

Enter valid login credentials.

Submit login form.

Expected Results: User should be logged in successfully.

Actual Results: User is logged in successfully.

Pass/Fail: Pass.

Test Case 4: BillingIntegrationTest

Description: Test the integration of billing functionality.

Preconditions: Resident with billable services.

Test Steps:Generate billing for the resident.

Verify invoice details.

Expected Results: Billing should be generated accurately.

Actual Results: Billing is generated accurately.

Pass/Fail: Pass.

System Test Cases

Test Case 5: RoomTransferSystemTest

Description: Test the system behavior for room transfer.

Preconditions: Resident with existing room allocation.

Test Steps:

Select a resident for room transfer.

Choose a vacant room.

Transfer the resident to the new room.

8.3. TEST REPORT

Testing Environment

- Description of the testing environment setup.
- Hardware and software requirements.
- Testing tools and frameworks used.

Test Coverage

- Summary of the areas covered by testing.
- Breakdown of testing types (unit testing, integration testing, system testing, etc.).
- Explanation of the rationale behind the test coverage.

Test Strategy

- Description of the testing approach adopted.
- Explanation of the testing methodologies used.
- Justification for the selected testing strategies.

Test Cases

- Detailed description of test cases developed for each testing type.
- Inputs, expected outcomes, and actual outcomes for each test case.
- Coverage of various scenarios and edge cases.

Test Execution

- Overview of the test execution process.
- Summary of test results, including pass/fail status.
- Identification of any issues encountered during testing.

Defect Management

- Documentation of identified defects.
- Classification and prioritization of defects.
- Description of defect resolution efforts.

Performance Testing Results

- Summary of performance testing outcomes.
- Key performance metrics such as response times, throughput, and resource utilization.
- Identification of performance bottlenecks and recommendations for optimization.

Security Testing Results

- Summary of security testing outcomes.
- Documentation of identified vulnerabilities.
- Recommendations for improving system security.

User Acceptance Testing (UAT) Results

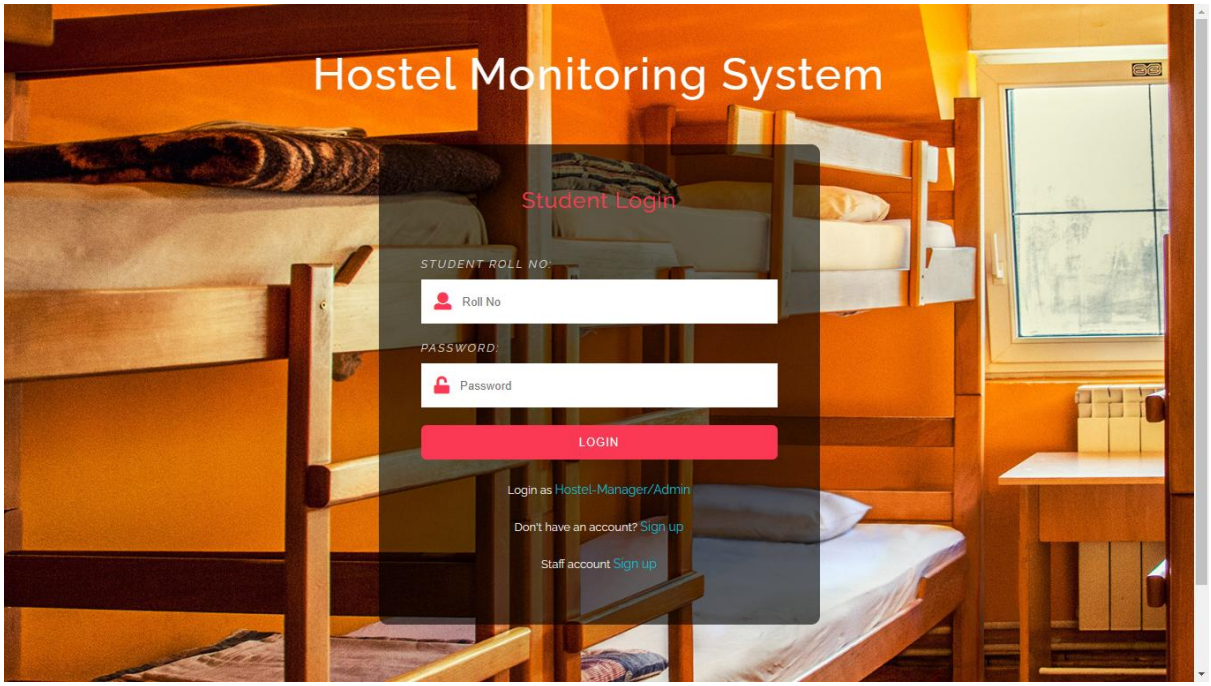
- Summary of UAT outcomes.
- Feedback from stakeholders/users.

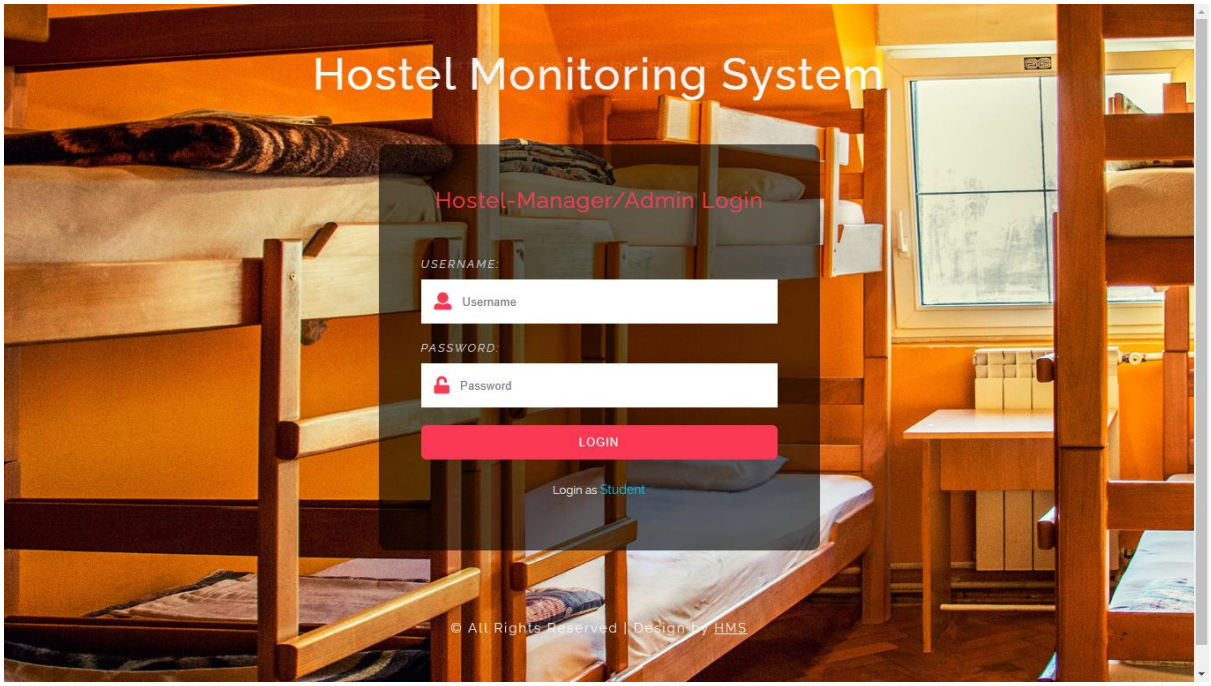
- Confirmation of system readiness for deployment based on UAT results.

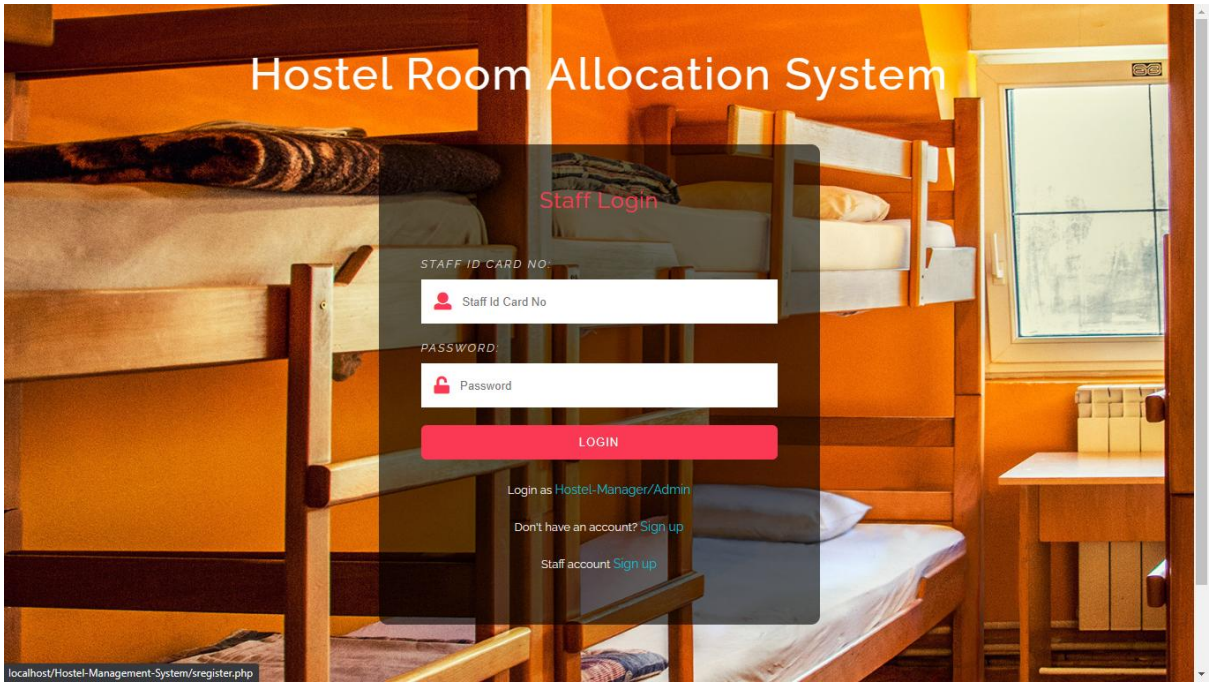
Test Conclusion:.

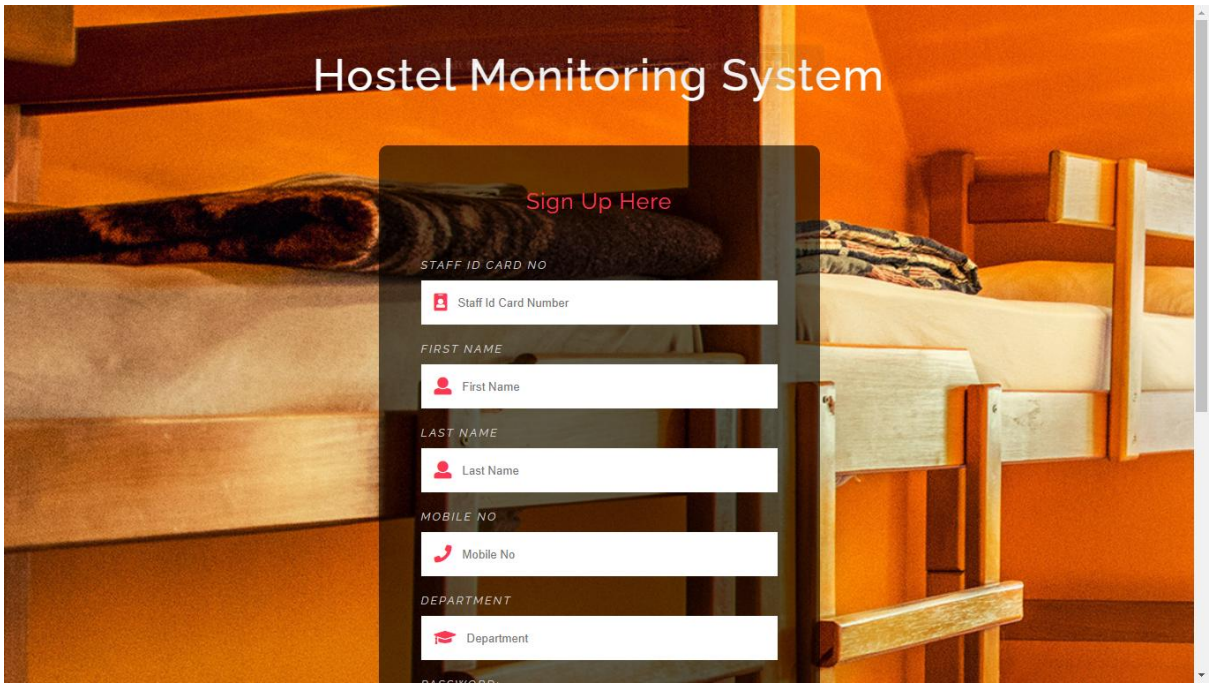
This PHP Test Report chapter provides a comprehensive overview of the testing process for the Hostel Management System, including test coverage, test strategy, test cases, test execution results, defect management, performance testing results, security testing results, UAT results, and conclusions. It serves as a valuable document for stakeholders and future reference.

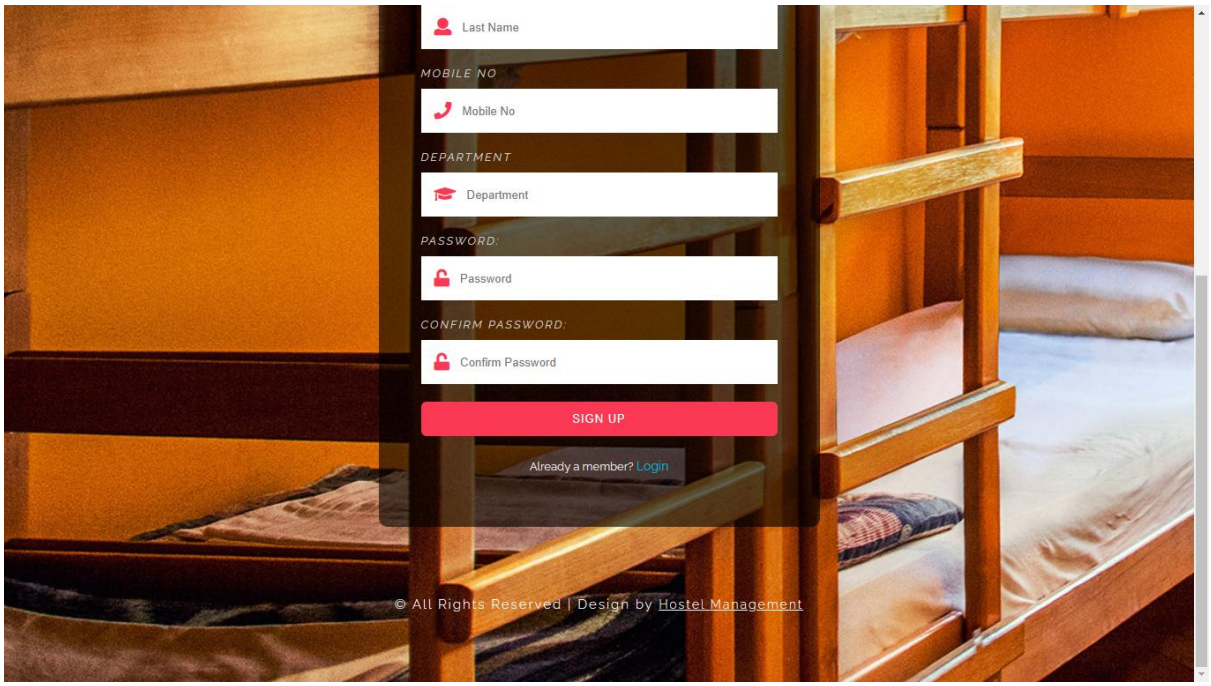
CHAPTER 9
SCREENSHOTS






Hostel Room Allocation System
Staff Login
STAFF ID CARD NO:
Staff Id Card No
PASSWORD:
Password
LOGIN
Login as Hostel-Manager/Admin
Don't have an account? Sign up
Staff account Sign up
localhost/Hostel-Management-System/sregister.php


Hostel Monitoring System
Sign Up Here
STAFF ID CARD NO
Staff Id Card Number
FIRST NAME
First Name
LAST NAME
Last Name
MOBILE NO
Mobile No
DEPARTMENT
Department


Last Name
MOBILE NO
Mobile No
DEPARTMENT
Department
PASSWORD:
Password
CONFIRM PASSWORD:
Confirm Password
SIGN UP
Already a member? Login
© All Rights Reserved | Design by Hostel Management

CHAPTER 10
CONCLUSION

In conclusion, the proposed Hostel Management System offers a modern and efficient solution to the challenges of managing a hostel. The system is designed to automate and streamline various tasks, such as managing room allocations, calculating mess bills, and maintaining attendance records, making it more accurate and convenient than the existing manual system. The proposed system will also offer enhanced accessibility, real-time reporting, and improved guest registration processes, providing a better experience for staff, students, and hostel administrators. By implementing this system, hostels can improve their overall efficiency, save time and effort, and provide a better living experience for their residents. Therefore, the proposed Hostel Management System is a significant step towards modernizing and optimizing the hostel management process. The introduction of an electronic system of registration focuses on saving cost, improving the efficiency of the processes involved in both registration and management of hostels and makes the overall procedure stress free. The hostel management system is aimed at streamlining the registration and management process of hostels for both students and the administrators in charge of the procedures involved. It is to eliminate unnecessary administrative tasks and reduce or even avoid paper work. This system will help improve productivity and reliability of the hostel registration and management process in a more efficient manner.

CHAPTER 11
FUTURE ENHANCEMENT

1. **Mobile Application Integration:**

Develop mobile applications for Android and iOS to allow students and wardens to manage leave requests on the go. This would increase accessibility and convenience for users.

2. Biometric Authentication:

Implement biometric authentication (e.g., fingerprint or facial recognition) to enhance security and ensure that only authorized users can access the system.

3. Real-Time Notifications:

Integrate real-time notifications via SMS, email, or push notifications to inform students and wardens of the status of leave requests and any updates immediately.

4. Automated Alerts and Reminders:

Set up automated alerts and reminders for students to submit leave requests and for wardens to review pending requests, ensuring timely processing and reducing delays.

5. Integration with Access Control Systems:

Link the leave management system with hostel access control systems to automatically update permissions for students' entry and exit based on approved leave requests.

6. Advanced Reporting and Analytics:

Develop advanced reporting and analytics features to provide insights into leave patterns, attendance trends, and system usage, helping wardens and administrators make informed decisions.

7. Parent Portal:

Create a dedicated portal for parents to monitor their child's leave requests, approval status, and leave history, ensuring better communication and transparency.

8. AI-Powered Leave Recommendation:

Incorporate artificial intelligence to analyze students' attendance, academic performance, and other factors to provide recommendations on whether leave should be granted or denied.

REFERENCES

1. "Hostel Management System" by Vigneshwaran N and Karthikeyan K, Journal of Advanced Research in Computer Science and Software Engineering, Vol. 9, Issue February 2019.
2. ."Development of Hostel Management System" by Abubakar Sadiq Yakubu and Iliyasu Bala Muhammad, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 6, Issue 2, February 2018.
3. Design and Implementation of Hostel Management System" by Oyelakin Olatunji, Adeniyi Adeniran, and Taiwo Oluwadare, International Journal of Computer Science and Mobile Computing, Vol. 8, Issue 3, March 2019.
4. ."A Web-Based Hostel Management System" by Md. Humayun Kabir, Md. Anwarul Islam, and Md. Ashraful Islam, International Journal of Computer Applications, Vol. 129, Issue 5, November 2015.
5. ."Hostel Management System with Biometric Authentication" by Saurabh B. Verma, International Journal of Innovative Technology and Exploring Engineering, Vol. 8, Issue 7S, May 2019.
6. Whitten, Bentley, and Dittman. 2004. System Analysis and Design Methods (5th ed). McGraw-Hill: New York, NY.
7. PerlScriptsJavaScripts.com. 2006. “MySQL Tutorial, Database Commands, Beginners Guide”. http://www.perlscriptsjavascripts.com/tutorials/mysql/index.html. Accessed 2/3/06.

12.1. Book References

1. HTML Elements. W3schools. Retrieved 3/16/15.
2. CSS Introduction. W3schools. Retrieved 3/16/15.
3. Xhtml.com. Retrieved on 5/16/12.
4. MySQL: Project Summary. Black Duck Software.Retrieved 9/17/12.
5. PHP Manual: www.php.net

12.2. Web References

1. PHP for Windows: php.net. Retrieved 10/29/13.https://wikipedia.org/wiki/PHP
2. https://wikipedia.org/wiki/HTML
3. https://wikipedia.org/wiki/CSS
4. https://wikipedia.org/wiki/MySQL
5. Segun O. Olatinwo and et al. 2014. “Development of an Automated Hostel Facility Management System”. Journal of Science and Engineering.