SCTR's Pune Institute of Computer Technology, Dhankawadi, Pune

AN INTERNSHIP REPORT ON

SOFTWARE RECORD MANAGEMENT SYSTEM USING FULL -STACK WEB DEVELOPMENT

SUBMITTED BY

Name: Durgesh Choudhary
Class: TE 4
Roll no: 31414

Under the guidance of

Dr . A.G. Phakatkar

DEPARTMENT OF COMPUTER ENGINEERING

ACADEMIC YEAR 2023-24

Acknowledgment

I would like to express my deepest gratitude to all those who have contributed to the successful completion of my internship report titled "**Software Record Management System " .**

This endeavor would not have been possible without the invaluable support, guidance, and encouragement I received from numerous individuals and organizations.

First and foremost, I extend my heartfelpreciation to my esteemed mentor **Dr . A. G. Phakatkar** . Their exceptional expertise, unwavering dedication, and continuous guidance were instrumental in shaping this internship project. Their profound knowledge and insightful feedback greatly enhanced my understanding of image processing and optimization techniques. I am truly grateful for their patience, encouragement, and invaluable suggestions throughout this journey.

I am thankful to our Head of Computer Engineering Department, **Dr. G. V. Kale**, for her indispensable support and suggestions throughout the internship work.

I would also like to take this opportunity to thank my internship guide **Prof.N.Y. Kapadnis** for giving me all the help and guidance needed. I am really grateful for his kind support and valuable suggestions that proved to be beneficial in the overall completion of this internship.

I would also genuinely like to express my gratitude to the Department Internship Coordinator, **Prof. P.P. Joshi**, for her constant guidance and support and for the timely resolution of the doubts related to the internship process.

Furthermore, I would like to acknowledge the support and assistance extended by the faculty members and staff of the Department of Computer Engineering. Their willingness to share their knowledge, resources, and time is greatly appreciated. Their valuable inputs and suggestions contributed to the success of this internship. Last but not least,

I would like to express my heartfelt gratitude to my family and friends for their unwavering support and encouragement throughout this internship. Their love, understanding, and motivation have been the driving force behind my achievements

`

Contents

1. Title 3
2. Introduction 4
3. Problem Statement 5
4. Objectives and Scope 6
5. Methodological Details 7
6. Modern Engineering Tools 9
7. Outcome / results of internship work (screenshots of work done) 11

`

List of Figures

Sr. no.	Name	Description
1	Fig. 1	Login Page
2	Fig. 2	Lab Dashboard
3	Fig. 3	Add a computer
4	Fig. 4	Add a Software License Record
5	Fig. 5	Admin Dashboard
6	Fig. 6	Categorizing Records by Software
7	Fig . 7	Categorizing Records by Department
8	Fig . 8	Categorizing Records by Date

`

1. Title

Software Record Management System using Full - Stack Web Development

Managing software licenses across multiple labs lacks centralized oversight, leading to non-compliance risks and reliance on error-prone manual methods. Real-time visibility into software installations is deficient, resulting in inaccurate inventories and compromised security. Absence of controls and policies contributes to software misuse risks, while expired licenses disrupt operations, incurring additional costs.

`

2. Introduction

1. **Centralized Platform for Streamlined Management:**
    - Your SRMS acts as a centralized hub where all software-related activities are managed. It eliminates the need for disparate tools or manual tracking methods, providing a unified platform for administrators to oversee software assets across multiple computer labs.
    - This centralization streamlines administrative tasks by consolidating license tracking, installation monitoring, and policy enforcement into one easy-to-use interface.
2. **Efficient License Tracking and Management:**
    - Tracking software licenses can be a complex and time-consuming process. Your SRMS automates this process, providing administrators with real-time insights into license usage, expiration dates, and compliance status.
    - By maintaining an accurate inventory of software licenses, your system helps institutions avoid over- or under-licensing, reducing unnecessary costs and ensuring compliance with vendor agreements.
3. **Real-time Installation Monitoring and Management:**
    - Managing software installations across multiple labs requires constant oversight. Your SRMS offers real-time monitoring capabilities, allowing administrators to track installations, updates, and usage patterns as they happen.
    - With proactive monitoring, administrators can quickly identify and resolve issues such as failed installations or unauthorized software deployments, minimizing disruptions and optimizing resource allocation.
4. **Enforcement of Usage Policies and Compliance:**
    - Ensuring compliance with usage policies and regulatory requirements is critical for institutions. Your SRMS enables administrators to define and enforce policies related to software usage, access permissions, and data security.
    - By automating policy enforcement and providing audit trails of user activities, your system helps institutions demonstrate compliance during audits and ensures adherence to industry standards and best practices.
5. **User-Friendly Interface for Enhanced Usability:**
    - The user-friendly interface of your SRMS enhances usability and adoption across institutions. Intuitive navigation, customizable dashboards, and role-based access control empower users to access the information and functionalities they need quickly and efficiently.
    - By providing a seamless user experience, your system minimizes the learning curve for administrators and IT personnel, increasing productivity and satisfaction.
6. **Advanced Features for Optimized Resource Management:**

- In addition to core functionalities, your SRMS offers advanced features such as automated workflows, integration with existing IT systems, and predictive analytics.
- These features enable institutions to streamline processes, optimize resource allocation, and anticipate future software needs accurately, ensuring that software resources are utilized efficiently and cost-effectively.

7. **Robust Security Measures for Data Protection:**
    - Security is a top priority in your SRMS, with robust measures in place to protect sensitive data and prevent unauthorized access.
    - Encryption of data, role-based access control, regular security audits, and compliance with industry standards ensure that institutions can trust your system to safeguard their valuable software assets and sensitive information.

`

3. Problem Statement

- The Managing licenses across multiple labs poses difficulty due to the lack of centralized oversight, leading to non-compliance risks and reliance on error-prone manual record-keeping methods.
- Real-time visibility into software installations is lacking, resulting in inaccurate inventories that compromise security, exacerbated by the reliance on outdated manual tracking methods.

- Absence of controls and clear policies contributes to the risk of software misuse and unauthorized installations, exposing institutions to security risks and operational disruptions.
- Expired licenses disrupt operations, necessitating reactive handling that incurs additional costs, while inadequate resource allocation undermines optimization efforts, impacting efficiency and budgetary constraints.

`

4. Objectives and Scope

Objectives:

1. **Efficient Lab Resource Management:** Enable lab administrators to effectively manage lab resources by providing tools to add, remove, or update computer information within their respective labs. This includes details such as computer specifications, peripherals, and location.
2. **Accurate Software Installation Tracking:** Facilitate the tracking of software installations on each computer across all labs. Capture essential details such as software name, version, installation date, license information, and current status (e.g., active, expired, pending renewal).
3. **Comprehensive License Management:** Implement features to manage software licenses comprehensively. This includes tracking license expiration dates, renewal requirements, usage limits, and compliance with licensing agreements to mitigate the risk of legal or financial penalties.
4. **Centralized Administration and Oversight:** Provide a centralized administrative interface for system administrators to oversee the entire Software Record Management System. This includes managing user accounts, assigning roles and permissions, generating reports, and ensuring uniformity and consistency across all labs.
5. **User-Friendly Interface:** Design an intuitive and user-friendly interface accessible to both administrators and lab users. The interface should facilitate easy data entry, retrieval, and navigation, promoting user adoption and satisfaction.

`

5. Methodological Details

1. **Requirement Analysis:** Document functional and non-functional requirements, prioritizing features based on user needs and system constraints Develop use cases and user stories to capture system interactions and user workflows.

2. **System Design:** Design a database schema to store department , lab , computer , softwares list . Also to establish proper relationships among the tables .

3. **User Interface Design:** Create an intuitive interface with features like adding , deleting and editing details of the entities accordingly . Also displaying the details of the entities on the based selected choice .

4. **Backend Development:** Build backend functionality for user authentication, records management, license recording , laboratory management, and using appropriate technologies.

5. **Frontend Development:** Develop responsive frontend components using HTML, CSS, and JavaScript to provide a seamless user experience.

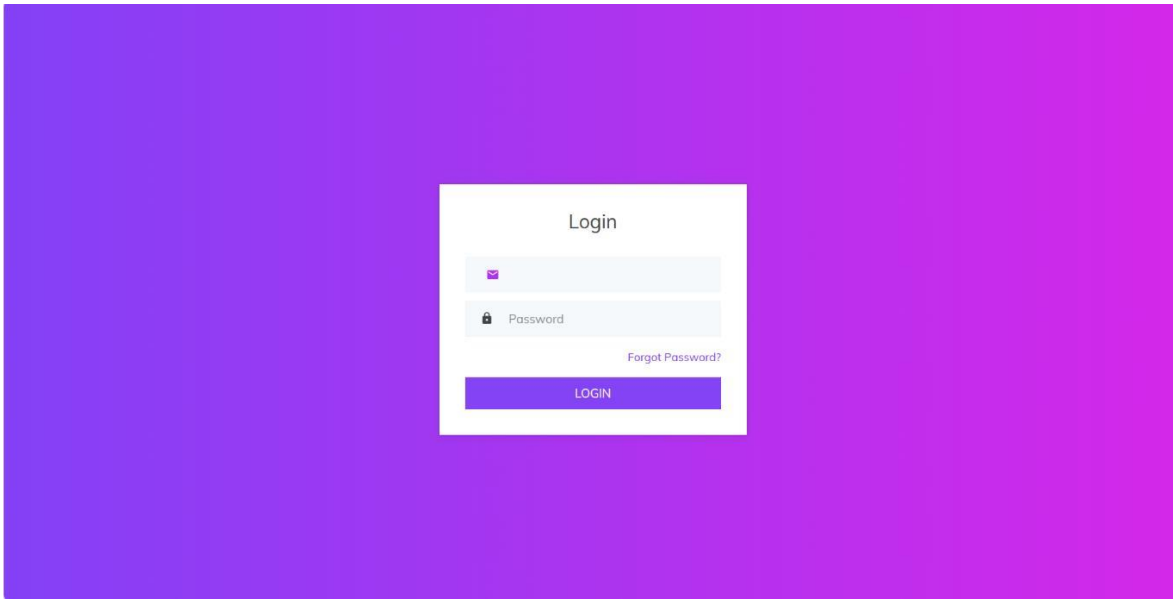
6. **Integration and Testing:** Integrate system components and conduct thorough testing to ensure functionality, performance, and security.

7. **Deployment and Maintenance:** Deploy the system, provide user training and support, and establish maintenance procedures for regular updates and improvements.

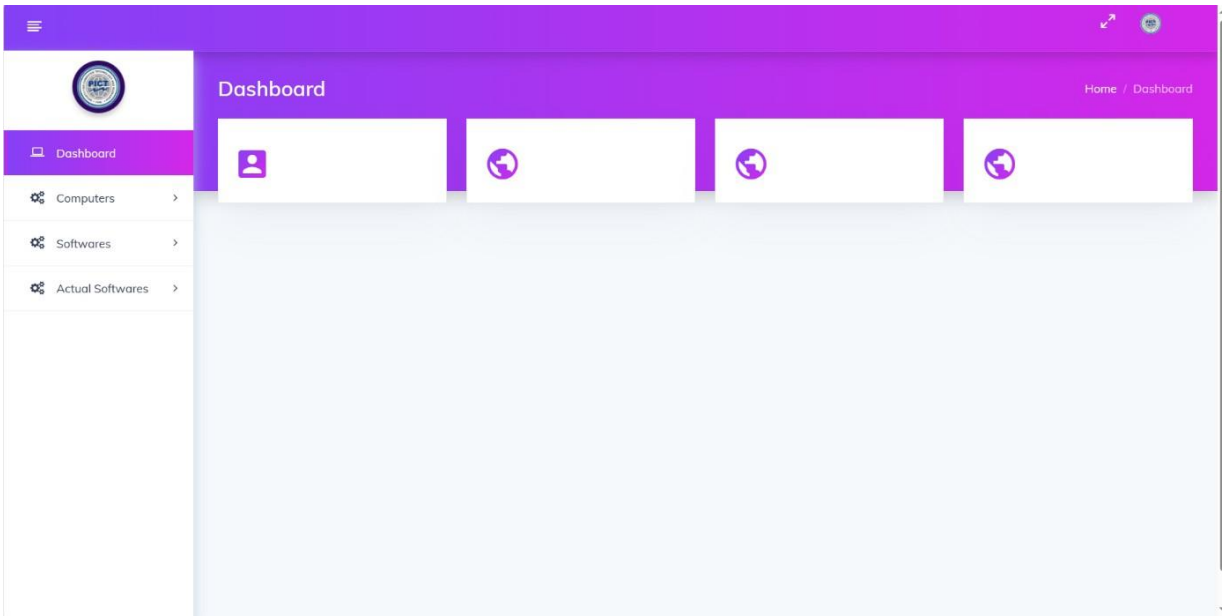
6. Modern engineering tools used -

– HTML
– CSS
– JavaScript
– Bootstrap
– Databases used: MySQL
– Backend Language: PHP
– Browser used: Google Chrome, Microsoft Edge
– Software used: XAMPP
– Editor used: Visual Studio
– Framework : CodeIgnitor

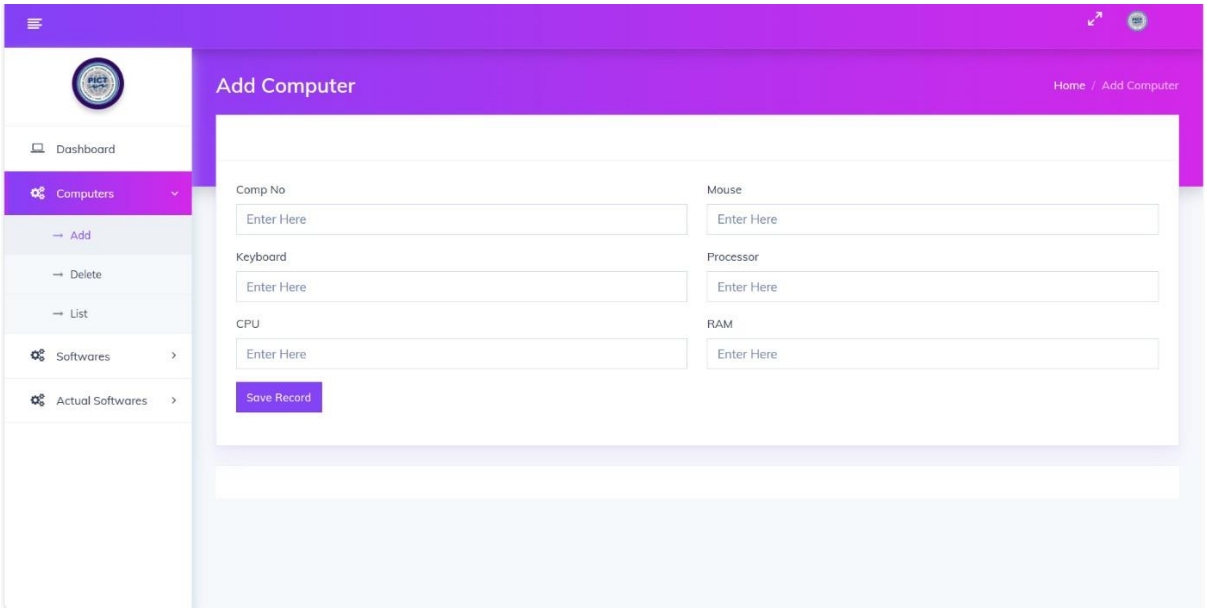
7. Outcome/ results of internship work (screenshots of workdone)



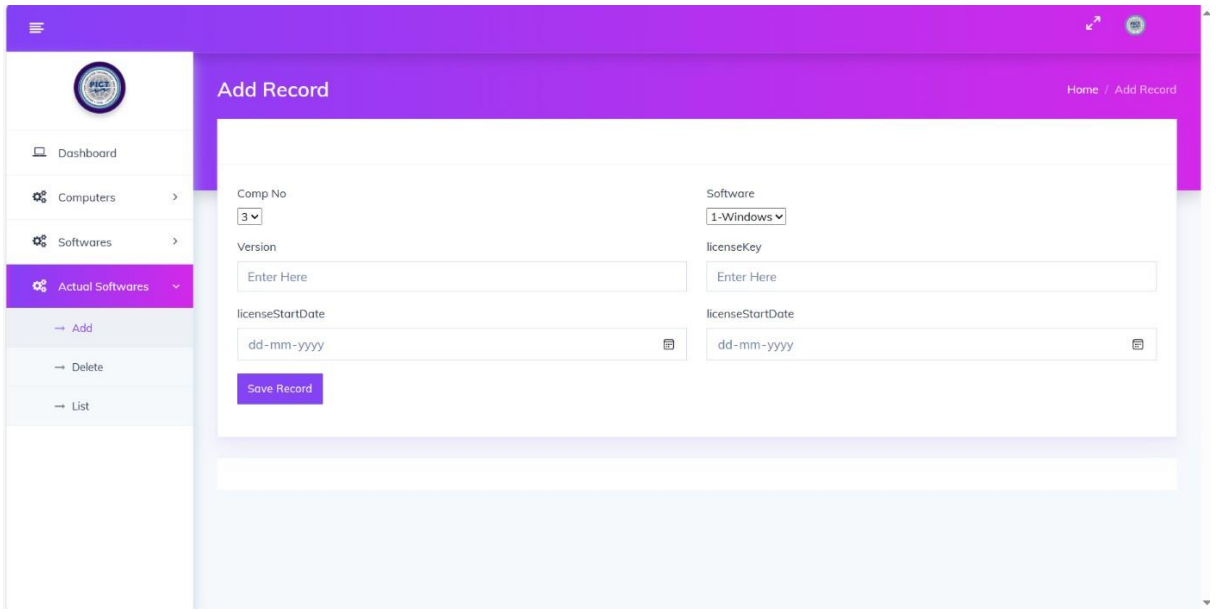
Login Page



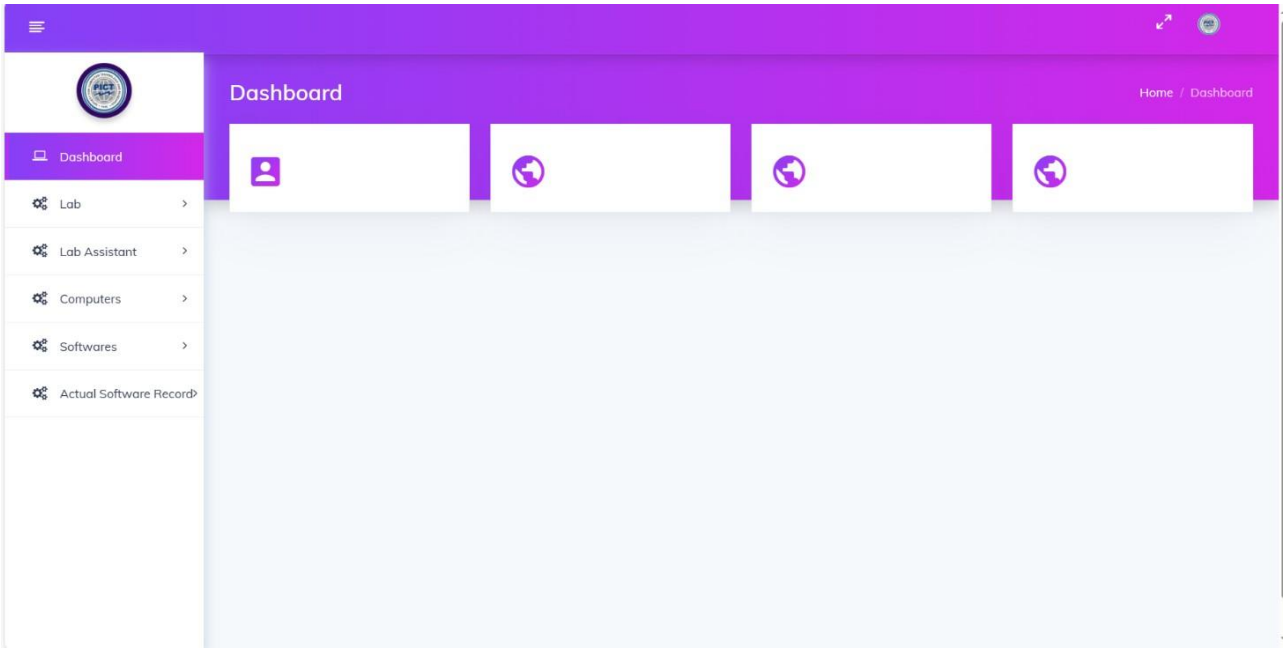
Lab Dashboard



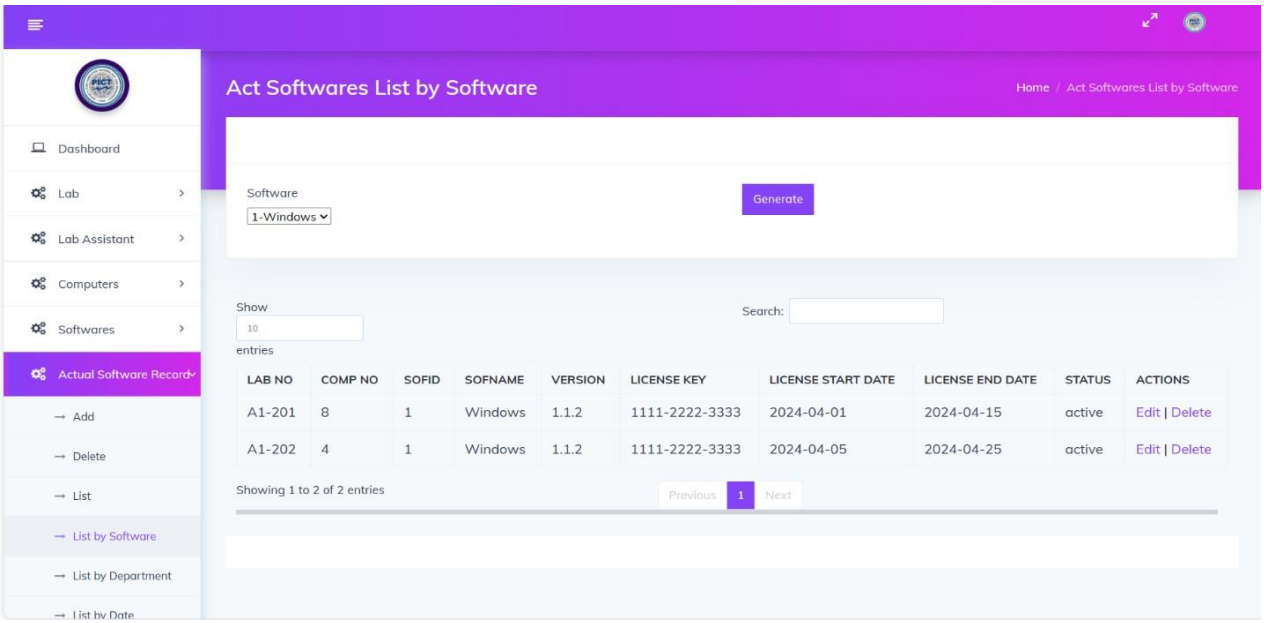
Add a Computer



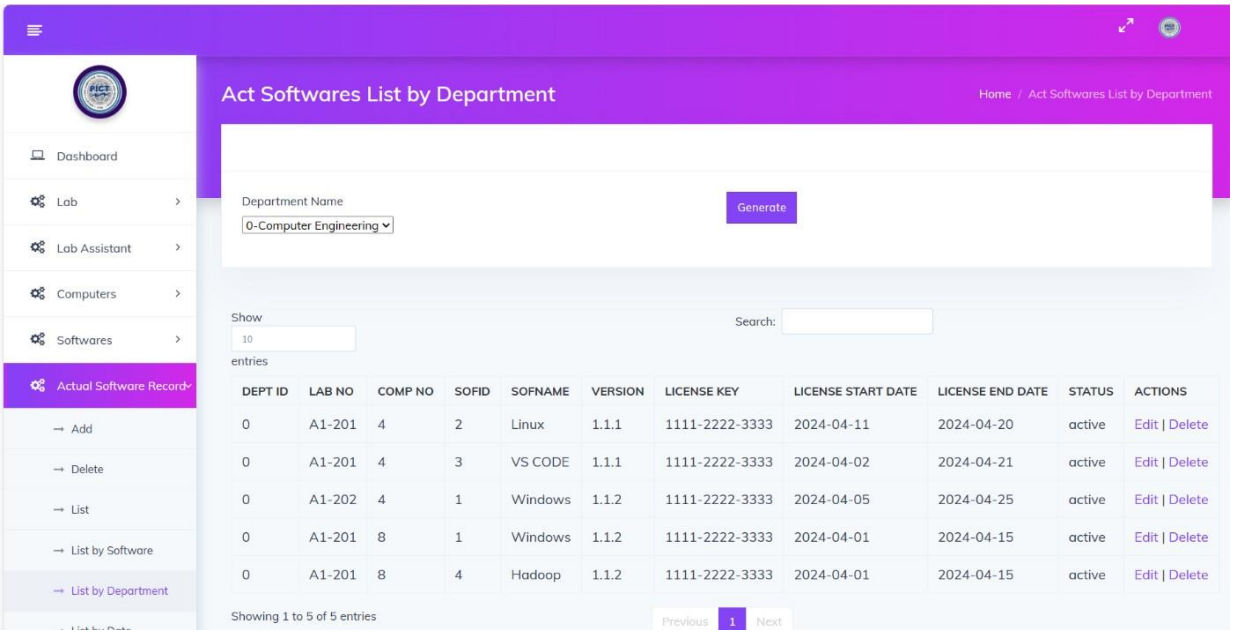
Add a Software License Record



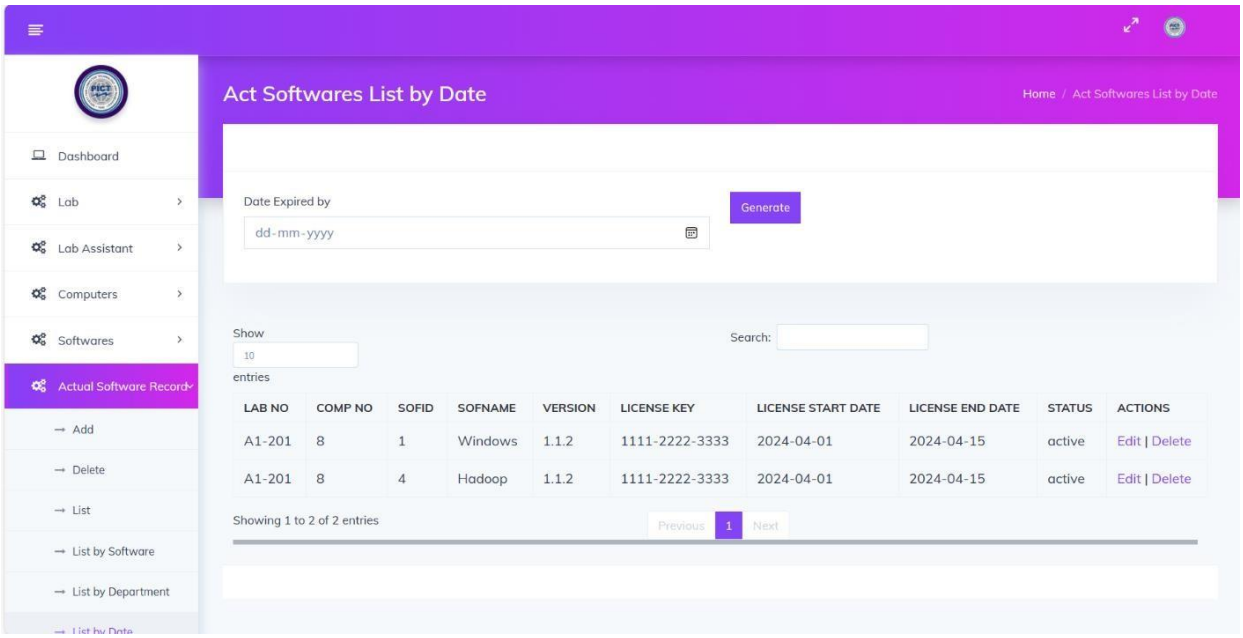
Admin Dashboard



Categorizing Records by Software



Categorizing Records by Department



Categorizing Records by Date

8. Any achievement (Job opportunity, project sponsorship, patent, commercial product, research publications, pre-placement offers, a strong professional network etc.)

During my tenure as intern, a pivotal achievement wasbackend development for the project. This initiative empowered me to leverage my expertise in HTML, CSS, JavaScript and to create a dynamic and user-friendly interface for the financial management system. I also contributed to the seamless integration of the front-end with the back-end, which was developed using Code Igniter. Additionally, my involvement in this project honed my skills in PHP and solidified my understanding of full-stack development methodologies. This experience for this project has been instrumental in enriching my professional journey, equipping me with valuable insights and skills that I look forward to applying in future endeavors.

Database Schema for Reference

```
-- phpMyAdmin SQL Dump
-- version 4.8.4
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Apr 30, 2024 at 06:40 AM
-- Server version: 10.1.37-MariaDB
-- PHP Version: 5.6.39

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `labms1`
--


-- --------------------------------------------------------
```

```
--
-- Table structure for table `act_table`
--

CREATE TABLE `act_table` (
  `id` int(11) NOT NULL,
  `deptID` int(11) NOT NULL,
  `labID` int(11) NOT NULL,
  `compNo` int(11) NOT NULL,
  `sofID` int(11) NOT NULL,
  `version` varchar(30) NOT NULL,
  `licenseKey` varchar(20) NOT NULL,
  `licenseStartDate` date NOT NULL,
  `licenseEndDate` date NOT NULL,
  `status` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;



-- --------------------------------------------------------
```

```
--
-- Table structure for table `computers`
--

CREATE TABLE `computers` (
  `id` int(11) NOT NULL,
  `labID` int(11) NOT NULL,
  `compNo` int(11) NOT NULL,
  `mouse` varchar(50) NOT NULL,
  `keyboard` varchar(50) NOT NULL,
  `processor` varchar(50) NOT NULL,
  `cpu` varchar(50) NOT NULL,
  `ram` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

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

--
-- Table structure for table `dept`
--

CREATE TABLE `dept` (
  `id` int(11) NOT NULL,
  `dept_name` varchar(40) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;



-- --------------------------------------------------------
```

```
--
-- Table structure for table `lab`
--

CREATE TABLE `lab` (
  `id` int(11) NOT NULL,
  `labNo` varchar(25) NOT NULL,
  `deptID` int(11) NOT NULL,
  `name1` varchar(25) NOT NULL,
  `pass` varchar(20) NOT NULL,
  `labasstID` int(11) NOT NULL,
  `logintoken` varchar(256) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

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

--
-- Table structure for table `labasst`
--

CREATE TABLE `labasst` (
  `id` int(11) NOT NULL,
  `name1` varchar(256) NOT NULL,
  `contact` text NOT NULL,
  `email` text NOT NULL,
  `eid` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
-- ----------------------------------------------------------


--
-- Table structure for table `softwares`
--

CREATE TABLE `softwares` (
  `id` int(11) NOT NULL,
  `sofName` varchar(120) NOT NULL,
  `sofType` varchar(30) NOT NULL,
  `subsType` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
-- --------------------------------------------------------

--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `iddate` datetime NOT NULL,
  `name` text NOT NULL,
  `contact` text NOT NULL,
  `email` text NOT NULL,
  `address` text NOT NULL,
  `pwd` text NOT NULL,
  `parentuser` int(11) NOT NULL,
  `status` int(11) NOT NULL COMMENT '0 OFF 1 ON',
  `lastlogin` datetime NOT NULL,
  `role` int(11) NOT NULL COMMENT '1 Admin  2 Firm  3 Accountant',
  `logintoken` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
-- Indexes for dumped tables
--


--
-- Indexes for table `act_table`
--
ALTER TABLE `act_table`
 ADD PRIMARY KEY (`id`),
 ADD UNIQUE KEY `unique1` (`labID`,`compNo`,`sofID`),
 ADD KEY `ActSofTabForkey1` (`sofID`),
 ADD KEY `ActSofTabForkey3` (`deptID`);


--
-- Indexes for table `computers`
--
ALTER TABLE `computers`
 ADD PRIMARY KEY (`id`),
 ADD UNIQUE KEY `labID` (`labID`,`compNo`);


--
-- Indexes for table `dept`
--
ALTER TABLE `dept`
 ADD PRIMARY KEY (`id`);


--
-- Indexes for table `lab`
--
ALTER TABLE `lab`
```

```
 ADD PRIMARY KEY (`id`),
 ADD UNIQUE KEY `labNo` (`labNo`),
 ADD KEY `ForeignKey2` (`deptID`);

--
-- Indexes for table `softwares`
--
ALTER TABLE `softwares`
 ADD PRIMARY KEY (`id`);

--
-- Indexes for table `users`
--
ALTER TABLE `users`
 ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `act_table`
--
ALTER TABLE `act_table`
 MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=23;

--
-- AUTO_INCREMENT for table `computers`
--
ALTER TABLE `computers`
 MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=33;
```

```
--
-- AUTO_INCREMENT for table `dept`
--
ALTER TABLE `dept`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT for table `lab`
--
ALTER TABLE `lab`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;

--
-- AUTO_INCREMENT for table `softwares`
--
ALTER TABLE `softwares`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `act_table`
```

```
--

ALTER TABLE `act_table`

 ADD CONSTRAINT `ActSofTabForkey1` FOREIGN KEY (`sofID`) REFERENCES `softwares` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,

 ADD CONSTRAINT `ActSofTabForkey3` FOREIGN KEY (`deptID`) REFERENCES `dept` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,

 ADD CONSTRAINT `fk_table2_table1` FOREIGN KEY (`labID`,`compNo`) REFERENCES `computers` (`labID`, `compNo`) ON DELETE CASCADE ON UPDATE CASCADE;


--

-- Constraints for table `computers`

--

ALTER TABLE `computers`

 ADD CONSTRAINT `ForiegnKey1` FOREIGN KEY (`labID`) REFERENCES `lab` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;


--

-- Constraints for table `lab`

--

ALTER TABLE `lab`

 ADD CONSTRAINT `ForeignKey2` FOREIGN KEY (`deptID`) REFERENCES `dept` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;

COMMIT;


/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```