CASE MANAGEMENT SYSTEM

PROJECT REPORT

For the partial fulfillment for the award of the degree
Of
POST-GRADUATION DIPLOMA

IN

INFORMATION AND COMMUNICATION TECHNOLOGY

(PGDICT)

Batch No-44

BKIICT, Bangladesh Computer Council
Sher-e-Bangla Nagar, Dhaka-1207.

09 September 2024

PROJECT REPORT

On

CASE MANAGEMENT SYSTEM

Submitted To

Madho Sudun Chanda
Regional Director
and
Course Coordinator PGDICT, Batch-44
BKIICT, Bangladesh Computer Council (BCC)

Under the Supervision of

Muhammad Shajjad Hossain Khan
Managing Director, Zend IT Solution Ltd.
and
Instructor PGDICT, Batch-44
BKIICT, Bangladesh Computer Council (BCC)
E-mail: shajjadhossain81@gmail.com

Submitted by

Name	**Roll**
Anika Akter	**05**
Nilufar Easmin	**08**
Nadia Islam	**09**
Mst. Peya Khatun	**12**

Course: Post Graduation Diploma in ICT (PGDICT), Batch-44

BKIICT, Bangladesh Computer Council
E-14/X, ICT Tower, Agargaon,
Sher-e-Bangla Nagar, Dhaka-1207.

CERTIFICATE

This is to certify that the project work “**Case Management System**” is a collaboration of work done by **Anika Akter, Nilufar Easmin, Nadia Islam and Mst. Peya Khatun** under my guidance in partial fulfillment of the requirements for the training program in Information and Communication Technology.

Muhammad Shajjad Hossain Khan
Managing Director, Zend IT Solution Ltd.
&
Course Instructor
Bangladesh Computer Council.

DECLARATION

We are happy to declaring the project “**Case Management System**” has been done by us under the supervision of **Muhammad Shajjad Hossain Khan**, Instructor, BCC. We also declare that neither this project nor any part of this project has been submitted elsewhere forward of any degree or Post Graduation Diploma in ICT.

Supervised by:

Muhammad Shajjad Hossain Khan
Managing Director, Zend IT Solution Ltd.
&
Course Instructor
Bangladesh Computer Council.

Abstract

Management of the data related to a police or judicial case has been considered as the most crucial and troublesome task for a lawyer. Therefore, development of an organized and user-friendly software to manage these data could open the prospect for the lawyer or related professionals to manage the cases with considerable confidence. In that retrospect, this project paper explores the scopes for the development, implementation, and benefits of a Case Management System (CMS) software. CMSs are designed to streamline and enhance the management of cases. This paper presents a comprehensive analysis and the key features of a CMS software, detailing its development, core functionalities, and its transformative impact on organizational efficiency and effectiveness. Our project focuses on constructing a Model for Case Management System which serves as a centralized platform where all case-related information—including documents, communications, deadlines, and case histories— have been stored and managed. This centralization reduces data redundancy and minimizes the risk of errors, facilitating a more organized and accurate case management process. The software typically includes features such as workflow automation, real-time case tracking, integrated communication tools, and robust reporting.

ACKNOWLEDGEMENTS

We are immensely grateful to Mr. Muhammad Shajjad Hossain Khan for his exceptional mentorship and guidance throughout the course of this project. His efforts, guidance, and patience have been invaluable in our journey.

We also extend our heartfelt thanks to Madho Sudun Chanda, Regional Director, and Course Coordinator PGDICT, Batch-44, BKIICT, Bangladesh Computer Council (BCC), for his constant support and coordination of the training program. His enthusiasm and inspiration have greatly contributed to our learning experience.

We would like to acknowledge the support and encouragement of all the Course Trainees and Project Assistants who have been instrumental in the successful completion of this project.

Furthermore, we express our gratitude to our beloved parents for their blessings, and to our friends and classmates for their unwavering support and well-wishes.

Lastly, we extend our thanks to Bangladesh Computer Council for providing us with such an invaluable learning opportunity and for the support and cooperation of the talented teachers and trainers who made this project possible.

Table of Contents

Declaration …………………………………………………………………………… .iv

Abstract ………………………………………………………………………………v

Acknowledgements………………………………………………………………..……vi

Tables of Contents ..…………………………………………………………………....vii

Chapter-1 : Introduction 2

1.1 Necessity 2

1.2 Software Development Method 3

1.3 Tools & Layout of Document 3

1.4 Aims and Objectives 3

1.5 Research/Investigations/Analysis 4

1.6 Problem Statement 4

1.7 Solution 5

1.8 Proposed System 6

1.9 Modules of the Case Management System 6

1.10 Similar Projects in Marketplace 7

Chapter-2 : Development Platform Introduction 9

2.1 Used Technologies 9

2.2 Hardware Requirements 9

2.3 Software Requirements 9

2.4 Data Access Features 12

2.5 Advantages of Relational Approach 12

Chapter-3 : System Analysis 16

3.1 Introduction 16

3.2 System Engineering and Analysis 16

3.3 System Analysis 16

3.4 Requirement Analysis 16

3.5 Software Development Life Cycle (SDLC) 17

3.6 Use Case Diagram 18

3.7 Database ER Diagram (Schema Diagram) 19

Chapter-4 : System Design 21
4.1 Introduction 21
4.2 Database Design 21
4.3 System Architecture Design 22
4.4 Code Design 22
4.5 Implementation 23

Chapter–5 : Coding Structure & Design 26
5.1 Major Parts and Dashboard 26
5.1.1 Admin User Login Form Design 26
5.1.2 Master Setup 27
5.2 Report Part 42
5.3 Important Links 44

Chapter-6 : Database Table Design 46
6.1 XAMPP Server Database Table Design View 46
6.2 Open a Connection to MySQL database 46
6.3 Case Management System Database Table Design 46
6.4 Case Details Table Design 47
6.5 Case Status Table Design 47
6.6 Case type Table Design 48
6.7 Court Info Table Design 48
6.8 Court type Table Design 49
6.9 Lawyer Info Table Design 49
6.10 Permission info table Design 50
6.11 Role info table Design 50
6.12 User info table Design 51
6.13 PHP MySQL Database 51
6.14 PHP + MySQL Database System 51
6.16 Database Queries 52

Chapter-7 : Application Testing 53
7.1 Introduction 53
7.2 Objectives of Software Testing 54
7.3 Principles of Software Testing 54

7.4 Strategy for Software Testing 55

Chapter-8 : Future Works Development 58
8.1 Introduction 58
8.2 List of Future Works Development 58

Chapter-9 : Conclusion 60

References : 62

IST OF FIGURES

Figure No.	Title	Page No.
Figure 1	Software Development Life Cycle (SDLC)	17
Figure 2	Use Case Diagram of Case Management System	18
Figure 3	Entity Relationship (ER) Schema Diagram of Case Management System	19
Figure 4	Software Architecture Design	22
Figure 5	Dashboard	26
Figure 6	Login Form	26

Chapter-1

Introduction

Chapter-1: Introduction

“CASE MANAGEMENT SYSTEM (CMS)”

Case Management System software is a transformative tool that addresses the complexities of case handling, leading to improved operational efficiency, enhanced accuracy, and better overall outcomes for organizations. By integrating CMS software, organizations can achieve higher levels of productivity and effectiveness in managing their case-related activities.

The development of CMS software involves several critical phases, starting with thorough requirements gathering to ensure the system addresses the specific needs of its users. This is followed by designing the system architecture, developing the software, and implementing it within the organization. The development process often employs agile methodologies to allow for iterative improvements and responsiveness to user feedback. Additionally, effective training and ongoing support are crucial to ensure users can leverage the full potential of the CMS.

The adoption of CMS software Will leads to significant improvements in organizational efficiency. It enhances productivity through automation of routine tasks and streamlined workflows, ensures greater accuracy in case management, and supports better decision-making with detailed reporting and analytics. Furthermore, it fosters increased transparency and accountability by providing real-time insights into case progress and status.

1.1 Necessity

In legal services case management involves handling a complex array of tasks and information. Cases often involve multiple stakeholders, extensive documentation, numerous deadlines, and intricate workflows. Traditional methods of managing these cases, such as manual processes or basic digital tools, can become overwhelming and inefficient along with several risks related to data mismanagement. CMS software is necessary to address these complexities by providing an integrated platform that streamlines and organizes case management tasks.

The accuracy of case-related information is crucial for effective case management. Manual entry and management of case data can lead to errors, miscommunication, and loss of critical information. CMS software centralizes all case information in a single repository, ensuring that data is accurate, up-to-date, and easily accessible. This centralized approach minimizes the risk of errors and ensures that all stakeholders have access to the correct information, thereby improving the overall reliability of case management.

As organizations grow, the volume and complexity of cases often increase. Manual case management systems and basic digital tools may not scale effectively to handle larger volumes of data and more complex workflows. CMS software is designed to scale with organizational growth, providing the necessary tools and infrastructure to manage increasing caseloads efficiently.

Transparency and accountability are crucial for managing cases effectively. CMS software provides real-time visibility into case status, progress, and outcomes. This transparency helps stakeholders track the progress of cases, monitor performance, and hold team members accountable for their tasks and responsibilities.

1.2 Software Development Method

For developing this Case Management System we have followed the agile methods that is an iterative methodology that incorporates a cyclic and collaborative process (Cohn). We have followed the agile methodology, our project moved through a series of cycles throughout the lifetime of the project. We kept touch us to our clients and beneficiaries for bringing any kind of changes, adding newly acquired knowledge and vetting with them.

As there are many software applications and models like Waterfall model, Iterative model, Spiral model, V-model and Big Bang model are used. We prefer to use the agile model in this system development. I tried to use test case and case software approaches (Ferreira, Pedro F.; Machado).

1.3 Tools & Layout of Document

In this application different types of tools are used. For documentation Ready set Tigris are used and for front end and backend part Eclipse and MySQL plays a great role.

This documentation starts with formal introduction. After introduction analysis and design of the project are described. In analysis and design of the project have many parts such as project proposal, mission, goal, target audience, environment. After that design and table diagram will be found. Use cases and test cases are in chapter 2 and chapter 3 respectively. Finally, this documentation finished with result and conclusion part.

1.4 Aims and Objectives

Its main aim is to enhance the efficiency and effectiveness of case Monitoring processing through automation and integration. CMS software aims to automate routine tasks, such as case filing, scheduling, and document management, thereby streamlining the entire case processing workflow. The software aims to offer features that allow court personnel to track case status, monitor deadlines, and manage case-related activities efficiently, ensuring timely and organized case handling.

The software aims to centralize case information in a secure database, reducing the risk of data entry errors and ensuring that all case-related data is accurate and up-to-date. The software also aims to provide real-time visibility into case progress and decision-making, allowing stakeholders to track case activities and hold responsible parties accountable.

Here are some prospective objectives for the Case Management System (CMS):

- To automate tasks such as case filing, scheduling of hearings, and document generation.
- To create a unified database for storing all case-related information, including documents, case notes, and court orders.
- To provide real-time updates on case status, deadlines, and progress.
- To develop reporting tools that generate insights into case metrics, workload distribution, and performance indicators.
- To Accommodate growth and evolving needs of the court system, ensuring long-term usability and relevance.

1.5 Research/Investigations/Analysis

It's very considerable and important action to perform a research, investigation and analysis before developing of a new system/software.

For a hassle free and disciplined official management, it is obligatory to find the information in a single point and a single source. Conduct market research to identify leading vendors, technologies, and best practices in Case Management System.

Analyze industry trends, market dynamics, and emerging technologies to inform decision-making and solution selection. Evaluate competitor strategies and benchmark against industry standards to identify opportunities for differentiation and innovation.

1.6 Problem Statement

Every organization face significant challenges in managing case workflows, data, and communications. These inefficiencies result from outdated manual processes, fragmented data management, and inadequate communication channels. The key problems include:

Manual and Time-Consuming Processes:

- Traditional court case management relies heavily on paper-based documentation and manual entry, which leads to slow processing times, high administrative overhead, and a greater likelihood of errors.

Fragmented Data Management:

- Case information is often stored in disparate systems or physical files, leading to difficulties in accessing and sharing information. This fragmentation can result in inconsistencies, missed deadlines, and a lack of visibility into case status.

Poor Communication and Coordination:

- Effective case management requires seamless communication among judges, lawyers, court staff, and other stakeholders. Inadequate communication tools hinder coordination, resulting in delays and misunderstandings.

Limited Data Analysis and Reporting:

- Traditional systems may lack advanced reporting and analytics capabilities, making it challenging to generate insights into case performance, workload distribution, and operational efficiency.

Compliance and Security Issues:

- Ensuring adherence to legal regulations and maintaining data security are critical concerns. Manual processes and outdated systems often struggle to meet regulatory requirements and protect sensitive information.

1.7 Solution

Implementation of a Comprehensive Case Management System

Automation of Administrative Tasks:

- The CMS reduces manual workload, accelerates case processing, and minimizes errors associated with manual data entry.

Centralized Case Repository:

- Centralized storage ensures that all stakeholders have access to up-to-date and consistent case information, improving data accessibility and reducing the risk of inconsistencies.

Real-Time Case Tracking:

- CMS provides real-time updates on case status, deadlines, and progress through an integrated tracking system. Stakeholders can monitor the current state of cases, manage deadlines effectively, and make timely decisions based on accurate information.

Enhanced Communication and Collaboration Tools:

- This common platform Facilitates seamless interaction between judges, lawyers, court staff, and other parties, improving coordination and reducing the likelihood of misunderstandings or delays.

Advanced Reporting and Analytics:

- The software enables data-driven decision-making, supports strategic planning, and helps identify areas for operational improvement.

Compliance and Security Features:

- CMS includes features to ensure compliance with legal regulations and protect sensitive information, such as audit trails, access controls, and data encryption.

Scalability and Flexibility:

- The CMS is designed to be scalable and adaptable, allowing for future expansion and adjustments based on evolving court or organization needs.

By addressing these core issues with a comprehensive CMS solution, organizations can achieve greater efficiency, accuracy, and effectiveness in case management. The system streamlines processes, improves data handling, enhances communication, and provides valuable insights, ultimately contributing to a more efficient and transparent system.

1.8 Proposed System

The CMS software will address the problems identified by automating and streamlining court case management processes. This software digitally creates and track cases from initiation to resolution. Automate case number generation and real-time status updates. This system Utilize a centralized, secure database to store and manage case information, reducing the risk of data loss and improving retrieval efficiency. This proposed CMS software aims to transform court case management by leveraging modern technology to enhance efficiency, transparency, and overall effectiveness in handling court cases.

System Requirements: The system requirement for this project is Software Requirement Microsoft Visual Studio 2010 It is an integrated development environment from Microsoft. It is used to develop console and graphical user interface application along with Windows Forms or WPF application, web sites, web application and web services in both native codes together with managed code for all platforms. It includes a code editor supporting intelligence as well as code refactoring. In this project Microsoft Visual Studio 2010 has been used. Microsoft Visual PHP It's an implementation of the php language, targets the .NET framework, along with the language services that lets the visual studio IDE support php projects. While the languages are a part of visual studio, the compiler is available separately as a part of the .NET framework. In my project I have used php 4.0.

Microsoft SQL Server It is a RDBMS developed by Microsoft. As a database it is a software product whose primary function is to store and retrieve data as requested by other applications, be it those on another computer across a network. I have used Microsoft SQL Server 2010 during the completion of my project. Hardware Requirements Computer Network Main Server machine SUMMARY *Create a Case Management System using PHP and visual studio. *Has minimal system requirements. *Implemented using Microsoft PHP, Microsoft Visual Studio, and Microsoft SQL Server.

1.9 Modules of the Case Management System

The proposed system will contain Six major sections:

- ❖ Dashboard
    - ➢ Pending Cases
    - ➢ Disposed Cases
    - ➢ Total Cases
    - ➢ Total Lawyers

- ❖ Master Setup
    - ➢ Court type
    - ➢ Case Status
    - ➢ Lawyer Information
    - ➢ Case Type
    - ➢ Court Information

- Admin Part (Backend)
    - Case Details List

- Report Part
    - Court Type-Wise
    - Case No-Wise
    - Date Range-Wise

- User Management Section
    - User Information
    - User-Wise Permission
    - Role Information
- Important Links
    - Bangladesh Supreme Court
    - Subordinate Courts
    - Laws of Bangladesh
    - Attorney General’s Office

1.10 Similar Projects in Marketplace:

https://scms.gov.bd/

Chapter-2

Development Platform Introduction

Chapter-2: Development Platform Introduction

2.1 Used Technologies

Case Management System is based on any windows environment. To run the system only need any popular browser. To do this, following tools and technologies are used.

2.2 Hardware Requirements:

- **CPU:** Intel® Core™ i5 Base Frequency 2.0 GHz & Turbo Frequency 3.8 GHz with 4 Core 6 MB Intel® Smart Cache;
- **Memory:** 4 GB;
- **Disk space:** 20 GB (3 GB for database files + enough for attachments)
- **Other:** Another Network card is required

2.3 Software Requirements:

- Scripting Language – PHP, JAVA Script
- Database Server - MYSQL
- Web Server – APACHE
- Framework – JQUERY, AJAX
- Other Language - HTML, CSS, BOOTSTRAP
- IDE – NETBEANS 8.1
- Operating System – Microsoft WINDOWS 11 pro

2.3.1 HTML

HTML Stands for “Hyper-Text Markup Language". This is the language that Web pages are written in. Also known as hypertext documents, Web pages must conform to the rules of HTML in order to be displayed correctly in a Web browser. The HTML syntax is based on a list of tags that describe the page's format and what is displayed on the Web page (Duckett).

2.3.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design (Duckett).

2.3.3 Java Script

A scripting language developed by Netscape to enable Web authors to design interactive sites. Although it shares many of the features and structures of the full Java language, it was developed independently. Java script can interact with HTML source code, enabling Web authors to spice up their sites with dynamic content. JavaScript is endorsed by a number of software companies and is an open language that anyone can use without purchasing a license. It is supported by recent browsers from Netscape and Microsoft, though Internet Explorer supports only a subset, which Microsoft calls JSCRIPT (Haverbeke).

2.3.4 PHP

PHP stands for hypertext preprocessor, which is usually called Personal Home Page (PHP).PHP is a server-side scripting language. There are two types of webpages

- Static Web Page: Static web page is a page which is static, which can only be viewed by user.
- Dynamic Web Page: Dynamic web page is a page which is user interactive which means that type of page can interact with user. PHP is a server-side scripting/language used for creating dynamic web pages.

Main features of PHP:

- PHP interact with one main external tool, the MYSQL database management system to access data stored in database.
- PHP is a server-side scripting language that can be embedded with HTML.
- PHP scripts are parsed and interpreted on the server side of a web application.

PHP has promising future because it is popular with Web developers and Web designers. It is powerful and easy to use. It can handle larger projects as well as smaller personal home pages. PHP is an open-source software which means that it can be redistributed free of charge and its source code is available for development (Welling, Luke; Thomson) (*W3Schools*).

2.3.5 MYSQL

SQL stands for Structured Query Language. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. MYSQL is a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon relational algebra. Some common relational database management systems that use SQL are: MYSQL, Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. MYSQL is an open-source language to retrieve data from database. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. MYSQL has proved to be a standard language as it allows users to learn one set of commands and use it to create retrieve alter and transfer information regardless Vof whether they are working PC a mini mainframe It is a high-level language that provides a greater degree of abstraction increased acceptance and availability It can be easily ported (Welling, Luke; Thomson) (Ahaghoghi, Seyed M.M. (Saied); Williams)(*MySQL Documentation*).

2.3.6 XAMPP Server

XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl. This simple and lightweight solution helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself.
XAMPP is said the “cross-platform” as it works on OS platforms like Windows, Linux, and Mac.

2.3.7 Database Platform

A database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information, such as modeling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.
Database management systems (DBMS) are computer software applications that interact with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase and IBM DB2 (Coronel, Carlos; Morris) (Islam, Anwarul; Islam, Md. Manjurul; Renu).

2.3.8 SQL Server Database Introduction

In computing, Microsoft SQL Server is a relational database management system, currently developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server - aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users (Ahaghoghi, Seyed M.M. (Saied); Williams).

2.3.9 SQL Server Database System Features

a) Supports most administrative tasks for SQL Server.

b) A single, integrated environment for SQL Server Database Engine management and authoring.

c) Dialogs for managing objects in the SQL Server Database Engine, Analysis Services, and Reporting Services, that allows you to execute your actions immediately, send them to a Code Editor, or script them for later execution.

d) Non-modal and resizable dialogs allow access to multiple tools while a dialog is open.

e) A common scheduling dialog that allows you to perform action of the management dialogs at a later time.

f) Exporting and importing SQL Server Management Studio server registration from one Management Studio environment to another.

g) Save or print XML Show plan or Deadlock files generated by SQL Server Profiler, review them later, or send them to administrators for analysis.

h) A new error and informational message box that presents much more information, allows you to send Microsoft a comment about the messages, allows you to copy messages to the clipboard, and allows you to easily e-mail the messages to your support team.

i) An integrated Web browser for quick browsing of MSDN or online help.

2.4 Data Access Features

Data Access Features allow you to create databases front-end application, and scalable server-side components for most database formats, including, Microsoft SQL Server and other enterprise-level databases.

2.5Advantages of Relational Approach

The Relational system offer Benefits such as easy access to all data, flexibility in data modeling. Reduces data storage and redundancy independence of physical storage and logical data design and high-level data manipulation language SQL. As the technologies associated with RDBMS have grown rapidly in recent years, the relational database has become apparent to a much wider audience. The phenomenal growth of the relational technology has laid to more demand for RDBMS in environment rising from PCs to large; highly secure CPUs with users ranging from very casual to very sophisticated. Some of advantages of relational approach over other approaches to data management are as follows (Islam, Anwarul; Islam, Md. Manjurul; Renu).

2.5.1 Power

The relation approach is very powerful and flexible in access to information and interrelating information without and programming concepts.

2.5.2 Adaptability

The features that make the relational approach more capable of accommodating changes are the immunity of the application programs activities.

2.5.3 Data Independence

The relational approach is the only one offers the four important Investment-protection features such as physical data independence, logical data independence, integrity independence and distribution independence.

2.5.4 Productibility

The ability to end user to make direct use of information relational databases without assistance is undoubtedly the primary reason why RDBMS market has been expanded so quickly.

2.5.5 Person to Person Communicability

With the relational approach an executive can; readily communicate with colleagues about the information stored in the database actions.

2.5.6 Database Controllability

The relational model was designed to provide much stronger machinery for maintaining the entity and referential integrity but also domain integrity, column integrity and defined integrity.

2.5.7 Optimizable

The translation from source code to efficient target code is usually called optimization problem. Almost all the present DBMS have superior capabilities in this area.

2.5.8 Ease of Conversion

All information in a database is perceived in the form of values. The language used in creating and manipulation relational database is a much higher level and will be much easier to convert to whatever approach replaces the relation model.

Chapter-3

System Analysis

Chapter-3: System Analysis

3.1 Introduction

Systems analysis is a problem-solving technique that decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose. As the software system requirements were predictable, it is decided to follow the classical system development life cycle method. This process demands a systematic, sequential approach to software development that begins at the system level and progress through analysis, design, coding, testing and maintenance. The steps that is applicable to all software engineering paradigms. The program is followed by SDLC (Software Development Life Cycle) (Ferreira, Pedro F.; Machado).

3.2 System Engineering and Analysis

Software is always a part of a large system; work begins by establishing requirement for all system elements and then allocating some subsets of this requirement to software. This system view is essential when software must interface with other elements such as hardware, people and database (Hoffer).

3.3 System Analysis

Analysis involves the requirement determination and specification. Systems analysis is a problem-solving technique that decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose. According to the Merriam-Webster dictionary, systems analysis is the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Analysis and synthesis, as scientific methods, always go hand in hand, they complement one another (Kendall, Kenneth E.; Kendall).

3.4 Requirement Analysis

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

Requirement's analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

3.5 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is a well-defined and structured sequence of stages in software engineering to develop the intended software product. It provides a series of steps to be followed to design and develop a software product efficiently. SDLC framework includes the following steps:

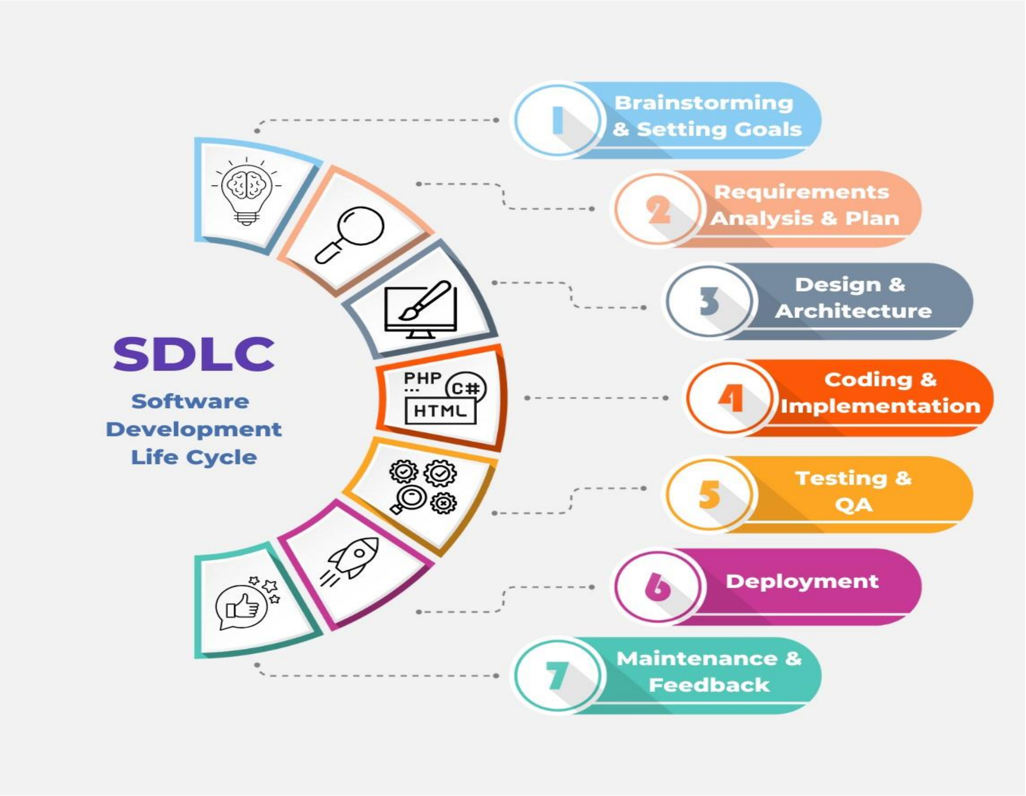

SDLC
Software Development Life Cycle
PHP
C#
HTML
1 Brainstorming & Setting Goals
2 Requirements Analysis & Plan
3 Design & Architecture
4 Coding & Implementation
5 Testing & QA
6 Deployment
7 Maintenance & Feedback

Figure 1: Software Development Life Cycle (SDLC).

Database design is the process of producing a detailed data model of database. And for that, first we give the use case diagram which required for the project:

3.6 Use Case Diagram:

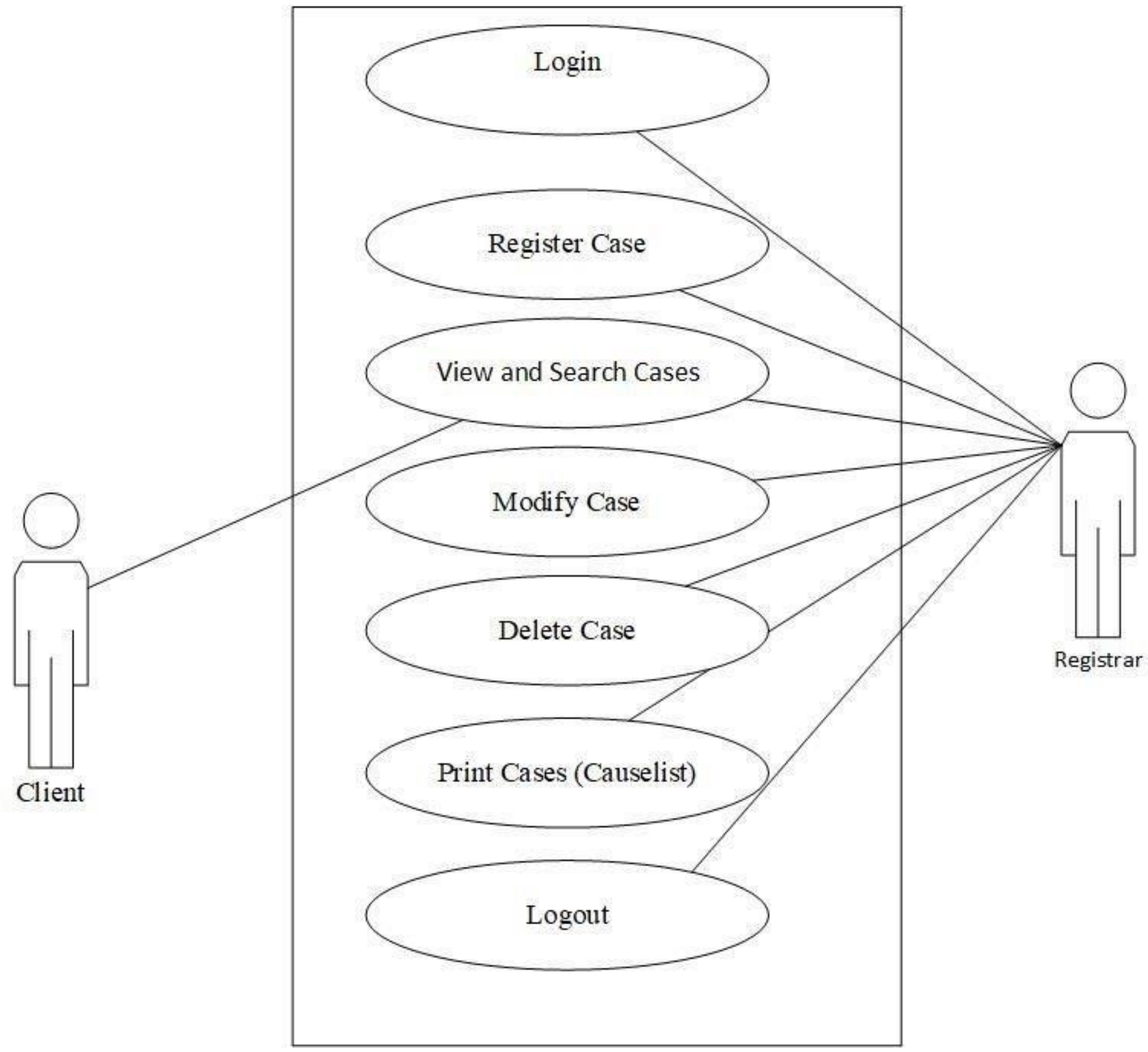


Figure 2: Use Case Diagram of Case Management System.

The first steps for the functional requirement collection are the use cases. Use cases are "a description of set of sequences of actions, including variants, that a system performs that yield an observable result of value to an actor".

3.7 Database ER Diagram (Schema Diagram)

The next step of the design phase is to draw a Database ER Diagram of the system. Since the programming language of the system is an object oriented one, a Database ER Diagram is particularly adapted to show the classes of the system, their inter relationships, and the operations and attributes of the classes.

Here is the class diagram of the project.

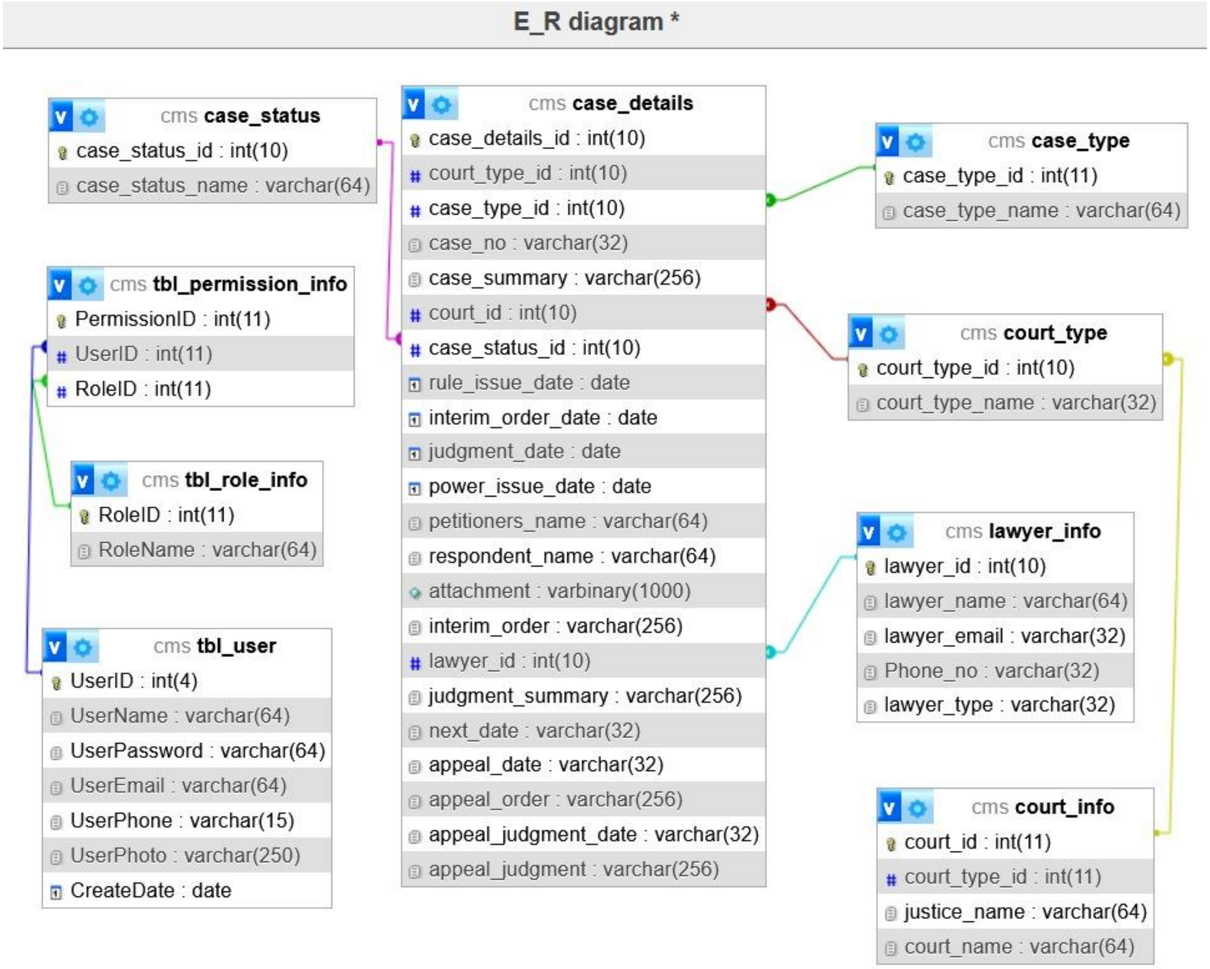


Figure 3: Entity Relationship (ER) Schema Diagram of Case Management System.

Chapter-4

System Design

Chapter-4: System Design

4.1 Introduction

The design phase is concerned with the physical construction of the system. Included are the design or configuration of the network (hardware, operating system, programming, etc.), design of user interfaces (forms, reports, etc.), design of system interfaces (for communication with other systems), and security issues. It is important that the proposed design be tested for performance, and to ensure that it meets the requirements outlined during the analysis phase. In other words, the main objective of this phase is to transform the previously defined requirements into a complete and detailed set of specifications which will be used during the next phase. Some of the activities that need to take place during the design phase are:

a) Design the application
b) Design and integrate the network
c) Design and integrate the database
d) Create a contingency plan
e) Start a Maintenance, Training and Operations plan
f) Review the design
g) Articulate the business processes and procedures
h) Establish a transition strategy
i) Deliver the System Design Document
j) Review final design

A database system is essentially nothing more than a computerized record keeping system the database itself can be regarded as kind of electronic filing cabinet. A database consists of same collection of some collection of persistent data that is used by the applications system of given some instituted. The term “instituted” here is simply a convenient generic term for any reasonable self- contained science, technical or other institution.

4.2 Database Design

A database management system (DBMS) is a collection of programs that enables you to store, modify, and extract information from a database. There are many different types of database management systems, ranging from small systems that run on personal computers to huge systems that run on mainframes.

DBMS is a software that handles the storage, retrieval, and updating of data in a computer system. Example: Ex-SQL Server (Microsoft), MySQL (Freeware), Oracle (Oracle), NoSQL (Oracle), NonStop SQL (Hewlett Packard (Coronel, Carlos; Morris)).

4.3 System Architecture Design

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.

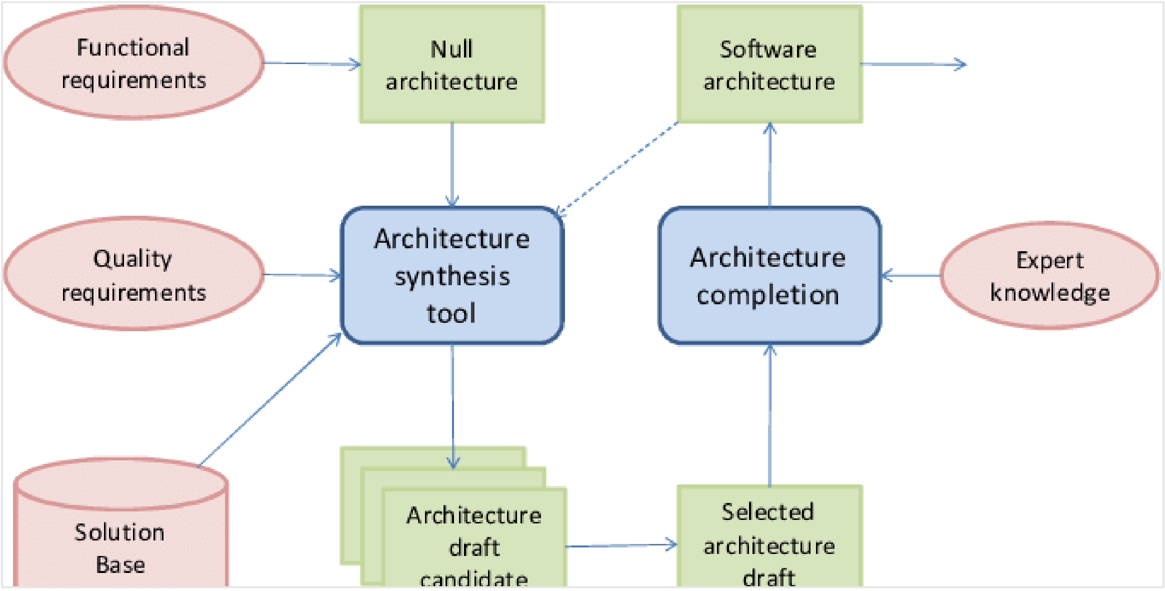


Figure 4: Software Architecture Design.

Software architecture refers to the high-level structures of a software system, the discipline of creating such structures, and the documentation of these structures. It is the set of structures needed to reason about the software system. Each structure comprises software elements, relations among them, and properties of both elements and relations. The architecture of a software system is a metaphor, analogous to the architecture of a building (Ferreira, Pedro F.; Machado).

4.4 Code Design

Design patterns are solutions to software design problems you find again and again in real-world application development. Patterns are about reusable designs and interactions of objects.

The 23 Gang of Four (GoF) patterns are generally considered the foundation for all other patterns. They are categorized in three groups: Creational, Structural, and Behavioral (for a complete list see below).

To give you a head start, the PHP source code for each pattern is provided in 2 forms: structural and real-world. Structural code uses type names as defined in the pattern definition and UML diagrams. Real-world code provides real-world programming situations where you may use these patterns.

A third form, .NET optimized, demonstrates design patterns that fully exploit built-in .NET 4.5 features, such as, generics, attributes, delegates, reflection, and more. These and much more are available in our .NET Design Pattern Framework 4.5.

This structural code demonstrates the Singleton pattern which assures only a single instance (the singleton) of the class can be created.

This real-world code demonstrates the Singleton pattern as a Load Balancing objects. Only a single instance (the singleton) of the class can be created because servers may dynamically come on-or off-line and every request must go through the one object that has knowledge about the state of the (web) farm (Pressman, Roger S.; Maxim).

4.5 Implementation

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation (Pressman, Roger S.; Maxim).

An implementation co-ordination committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions are made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

Implementation is the final and important phase. This is the most critical stage it achieving a successful new system and in giving the users confidence that the new system will work is effective. The system can be implemented only after thorough testing. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

Chapter-5

Coding Structure & Design

Chapter–5: Coding Structure & Design

Introduction of Coding Structure

The system has mainly Six parts:

5.1 Major Parts and Dashboard

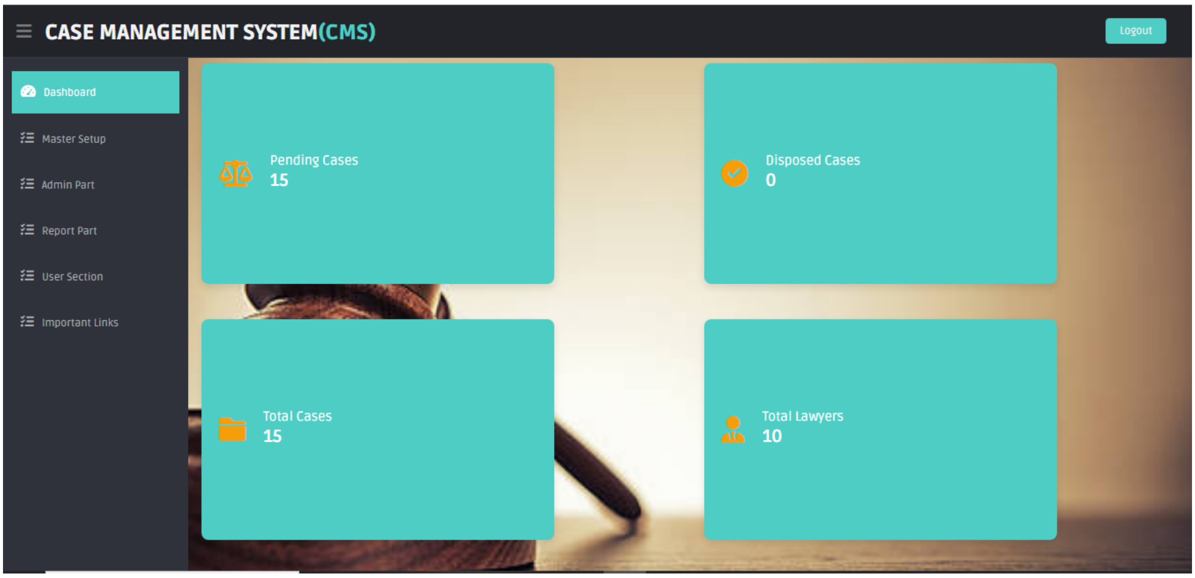


Figure 5: Dashboard

5.1.1 Admin User Login Form Design:

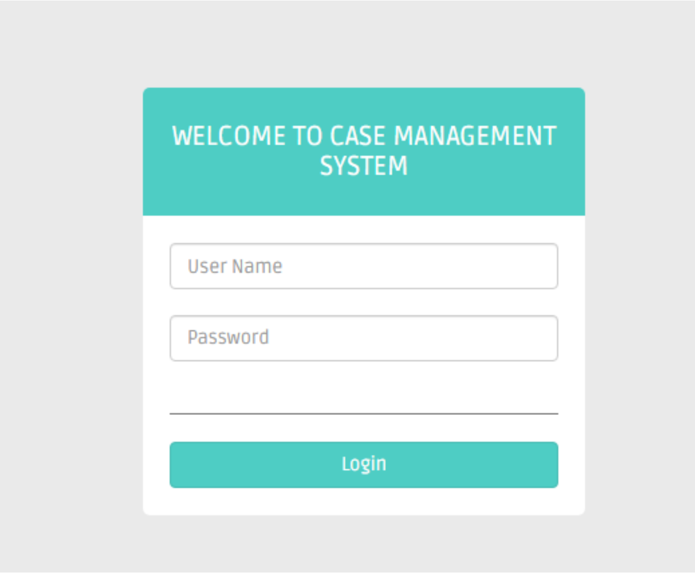
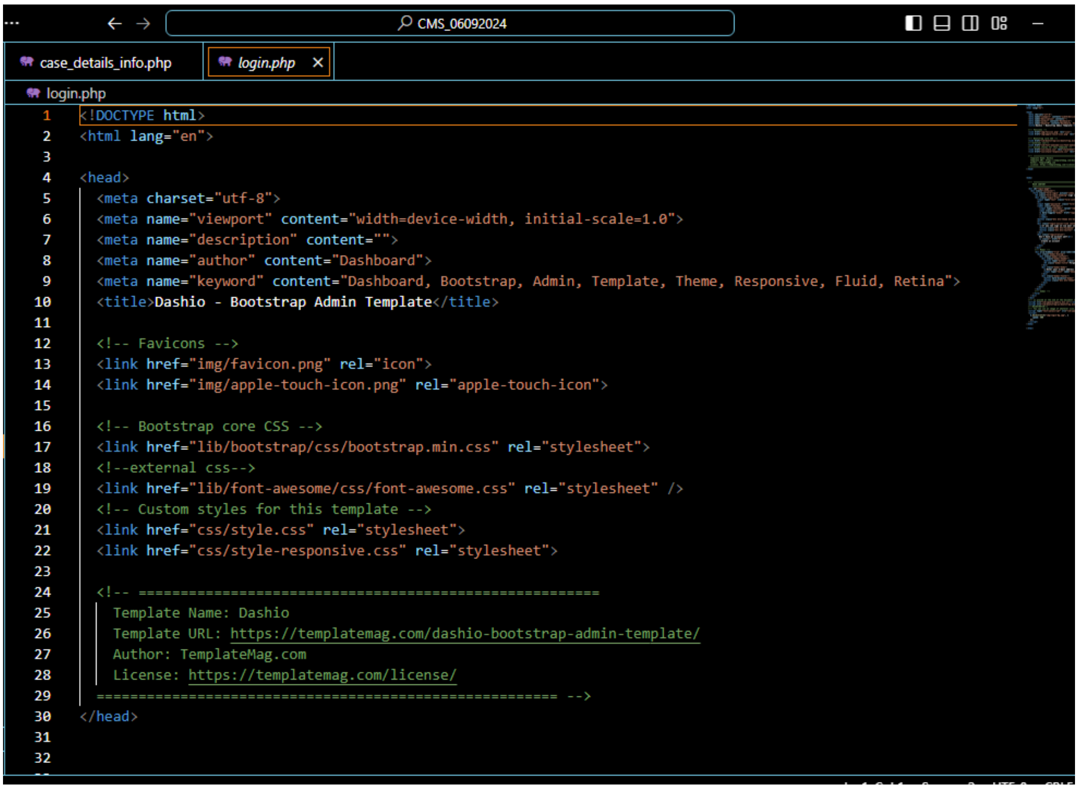


Figure 6: Login Form

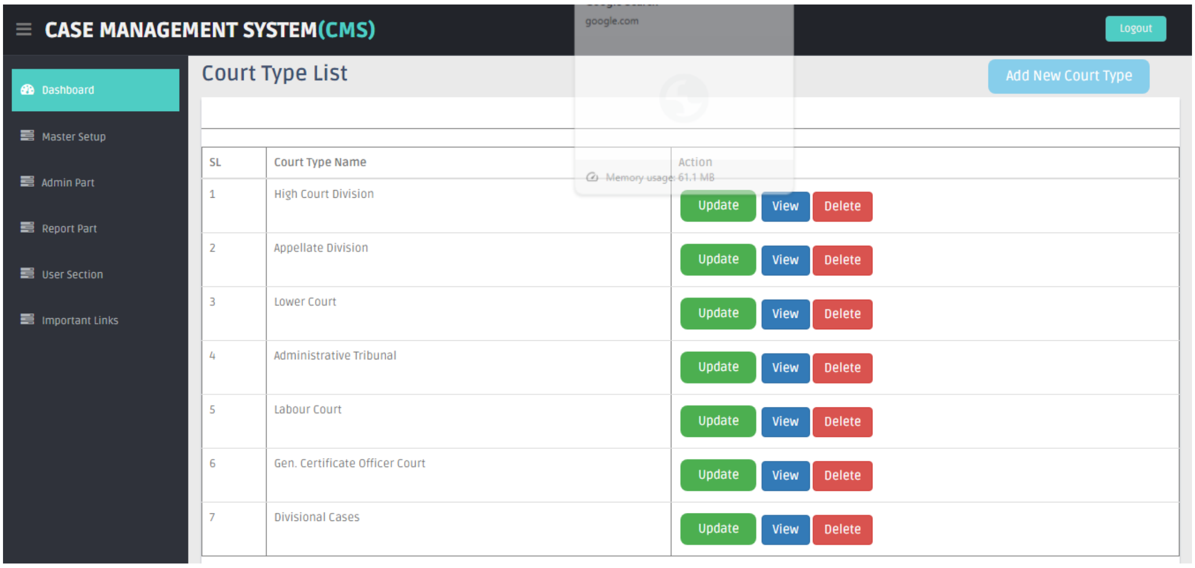
Code:

File: login.php (admin)



```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="">
  <meta name="author" content="Dashboard">
  <meta name="keyword" content="Dashboard, Bootstrap, Admin, Template, Theme, Responsive, Fluid, Retina">
  <title>Dashio - Bootstrap Admin Template</title>

  <!-- Favicons -->
  <link href="img/favicon.png" rel="icon">
  <link href="img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Bootstrap core CSS -->
  <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <!--external css-->
  <link href="lib/font-awesome/css/font-awesome.css" rel="stylesheet" />
  <!-- Custom styles for this template -->
  <link href="css/style.css" rel="stylesheet">
  <link href="css/style-responsive.css" rel="stylesheet">

  <!-- =======================================================
    Template Name: Dashio
    Template URL: https://templatemag.com/dashio-bootstrap-admin-template/
    Author: TemplateMag.com
    License: https://templatemag.com/license/
  ======================================================= -->
</head>
```

5.1.2 Master Setup: >> Court Type List View

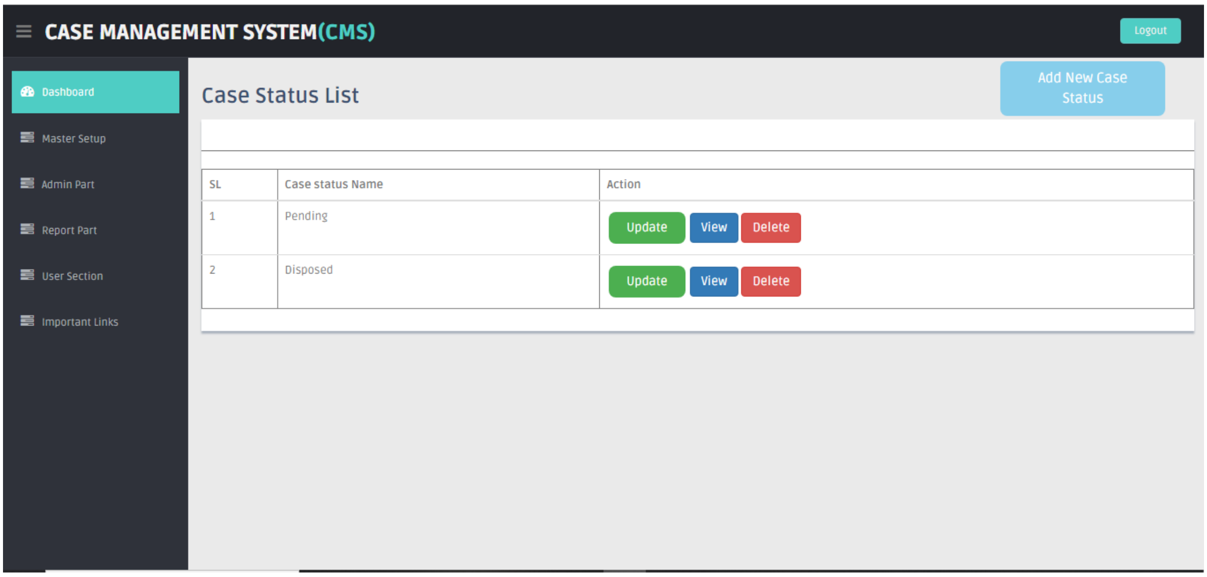


Court Type List

Add New Court Type

SL	Court Type Name	Action
1	High Court Division	Update View Delete
2	Appellate Division	Update View Delete
3	Lower Court	Update View Delete
4	Administrative Tribunal	Update View Delete
5	Labour Court	Update View Delete
6	Gen. Certificate Officer Court	Update View Delete
7	Divisional Cases	Update View Delete

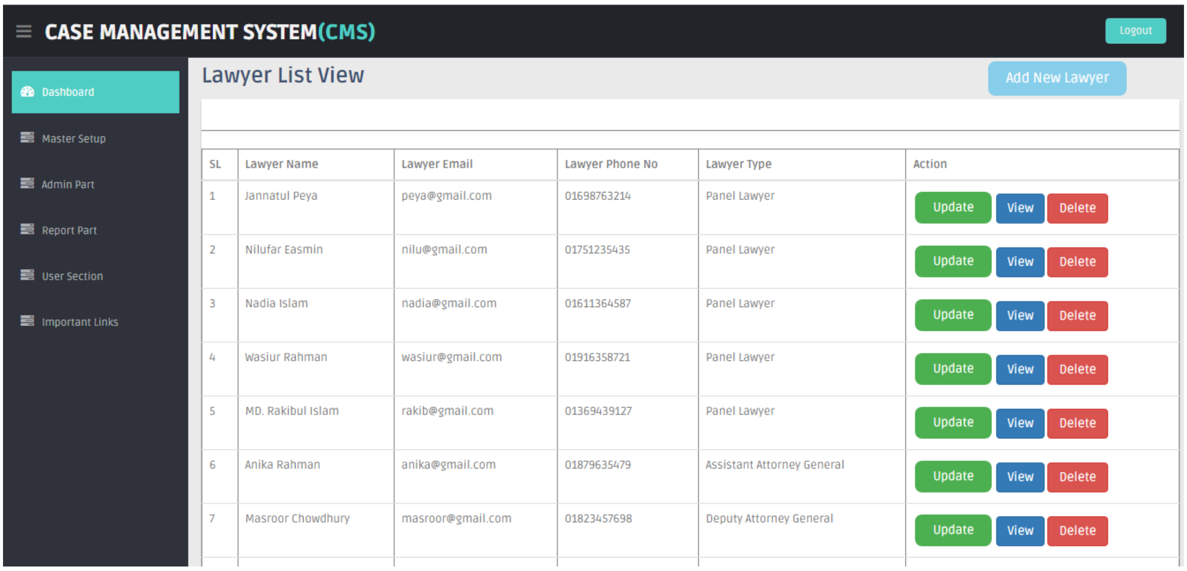
5.1.3 Master Setup: >> Case Status List View



Case Status List

SL	Case status Name	Action
1	Pending	Update View Delete
2	Disposed	Update View Delete

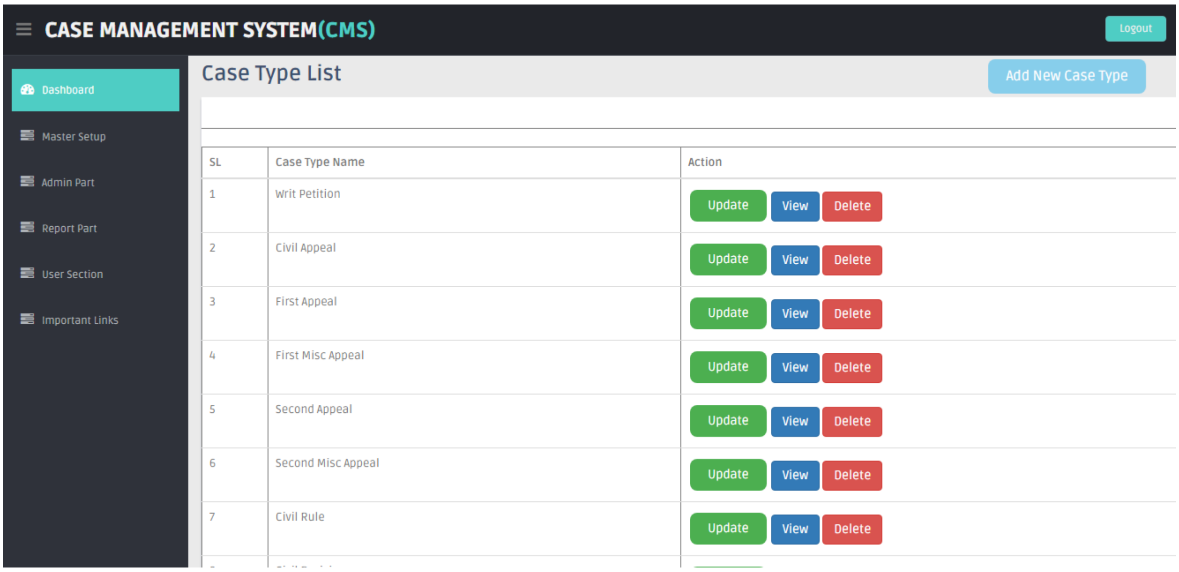
5.1.4 Master Setup: >> Lawyer List View



Lawyer List View

SL	Lawyer Name	Lawyer Email	Lawyer Phone No	Lawyer Type	Action
1	Jannatul Peya	peya@gmail.com	01698763214	Panel Lawyer	Update View Delete
2	Nilufar Easmin	nilu@gmail.com	01751235435	Panel Lawyer	Update View Delete
3	Nadia Islam	nadia@gmail.com	01611364587	Panel Lawyer	Update View Delete
4	Wasiur Rahman	wasiur@gmail.com	01916358721	Panel Lawyer	Update View Delete
5	MD. Rakibul Islam	rakib@gmail.com	01369439127	Panel Lawyer	Update View Delete
6	Anika Rahman	anika@gmail.com	01879635479	Assistant Attorney General	Update View Delete
7	Masroor Chowdhury	masroor@gmail.com	01823457698	Deputy Attorney General	Update View Delete

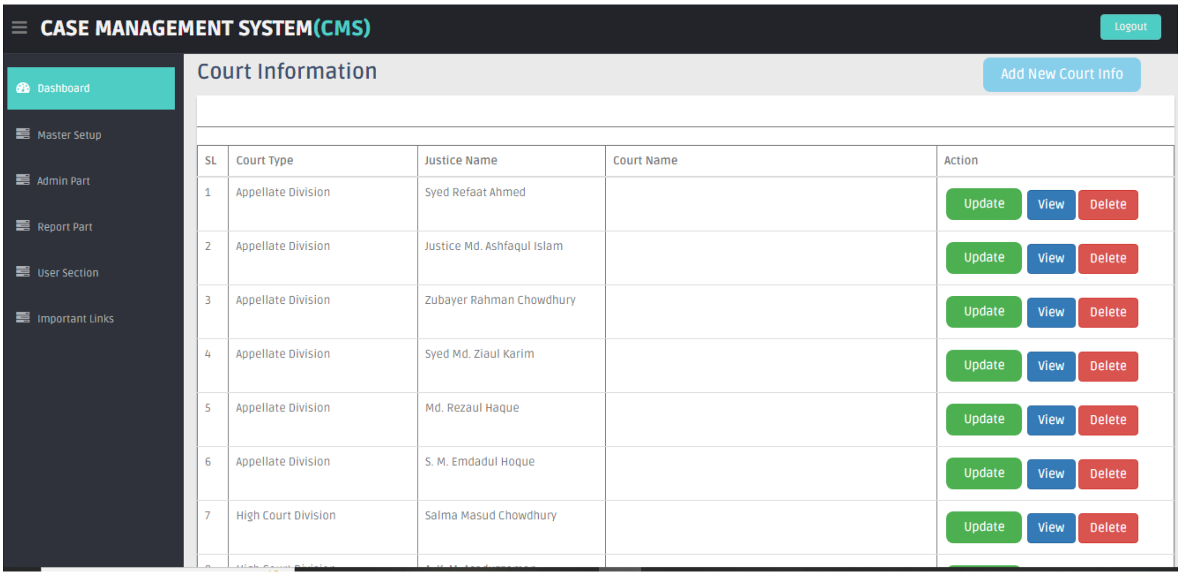
5.1.5 Master Setup: >> Case type List view



Case Type List

SL	Case Type Name	Action
1	Writ Petition	Update View Delete
2	Civil Appeal	Update View Delete
3	First Appeal	Update View Delete
4	First Misc Appeal	Update View Delete
5	Second Appeal	Update View Delete
6	Second Misc Appeal	Update View Delete
7	Civil Rule	Update View Delete

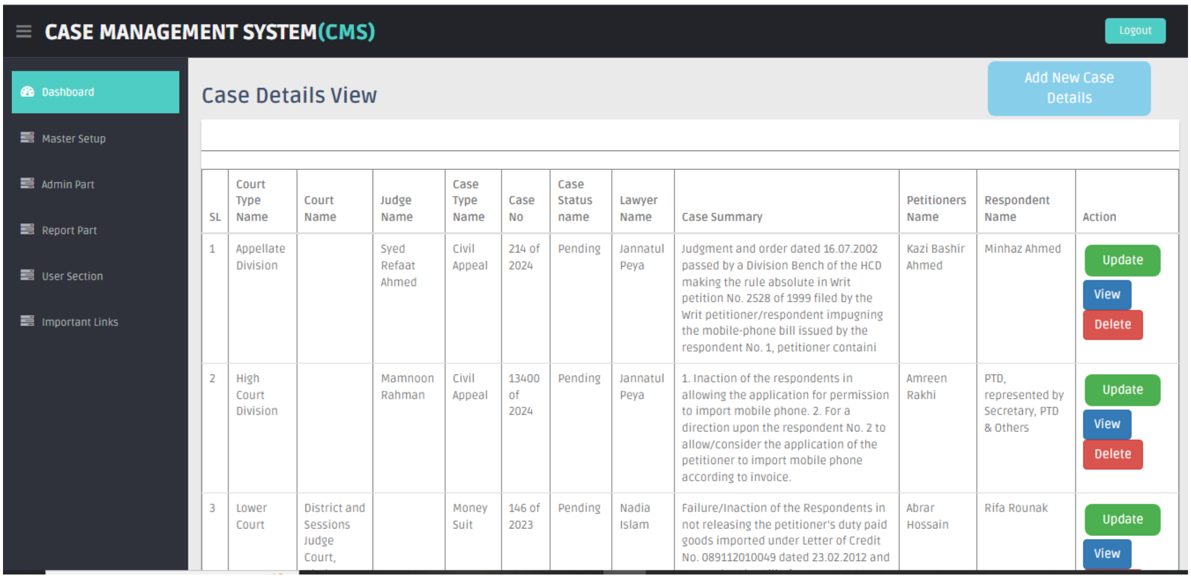
5.1.6 Master Setup >> Court Information List view



Court Information

SL	Court Type	Justice Name	Court Name	Action
1	Appellate Division	Syed Refaat Ahmed		Update View Delete
2	Appellate Division	Justice Md. Ashfaqul Islam		Update View Delete
3	Appellate Division	Zubayer Rahman Chowdhury		Update View Delete
4	Appellate Division	Syed Md. Ziaul Karim		Update View Delete
5	Appellate Division	Md. Rezaul Haque		Update View Delete
6	Appellate Division	S. M. Emdadul Hoque		Update View Delete
7	High Court Division	Salma Masud Chowdhury		Update View Delete

5.1.7 Admin Part >> Case Details List View



Case Details View

SL	Court Type Name	Court Name	Judge Name	Case Type Name	Case No	Case Status name	Lawyer Name	Case Summary	Petitioners Name	Respondent Name	Action
1	Appellate Division		Syed Refaat Ahmed	Civil Appeal	214 of 2024	Pending	Jannatul Peya	Judgment and order dated 16.07.2002 passed by a Division Bench of the HCD making the rule absolute in Writ petition No. 2528 of 1999 filed by the Writ petitioner/respondent impugning the mobile-phone bill issued by the respondent No. 1, petitioner containi	Kazi Bashir Ahmed	Minhaz Ahmed	Update View Delete
2	High Court Division		Mamnoon Rahman	Civil Appeal	13400 of 2024	Pending	Jannatul Peya	1. Inaction of the respondents in allowing the application for permission to import mobile phone. 2. For a direction upon the respondent No. 2 to allow/consider the application of the petitioner to import mobile phone according to invoice.	Amreen Rakhi	PTD, represented by Secretary, PTD & Others	Update View Delete
3	Lower Court	District and Sessions Judge Court,		Money Suit	146 of 2023	Pending	Nadia Islam	Failure/Inaction of the Respondents in not releasing the petitioner's duty paid goods imported under Letter of Credit No. 089112010049 dated 23.02.2012 and	Abrar Hossain	Rifa Rounak	Update View

Code:

```
<!DOCTYPE html>
<html lang="en">

<?php
   include('connection.php');
   include('function.php');

  $court_type_list           = court_type_list();
  $case_type_list            = case_type_list();
  $case_status_type_list     = case_status_type_list();
  $lawyer_list               =lawyer_list();
  $court_name_list           =court_name_list();
  $justice_name_list         =justice_name_list();

  if(isset($_POST['save']))
  {
     $court_type_id         = $_POST['court_type_id'];
     $case_type_id          = $_POST['case_type_id'];
     $case_no               = $_POST['case_no'];
```

```
    $court_id             = $_POST['court_id'];
    $case_status_id       = $_POST['case_status_id'];
    $rule_issue_date      = $_POST['rule_issue_date'];
    $interim_order_date   = $_POST['interim_order_date'];
    $judgment_date        = $_POST['judgment_date'];
    $power_issue_date     = $_POST['power_issue_date'];
    $petitioners_name     = $_POST['petitioners_name'];
    $respondent_name      = $_POST['respondent_name'];
    $case_summary         = $_POST['case_summary'];
    $attachment           = $_POST['attachment'];
    $interim_order        = $_POST['interim_order'];
    $lawyer_id            = $_POST['lawyer_id'];
    $judgment_summary     = $_POST['judgment_summary'];
    $next_date            = $_POST['next_date'];
    $appeal_date          = $_POST['appeal_date'];
    $appeal_order         = $_POST['appeal_order'];
    $appeal_judgment_date = $_POST['appeal_judgment_date'];
    $appeal_judgment      = $_POST['appeal_judgment'];

    // Handling file upload
    $attachment = '';
    if (isset($_FILES['attachment']) && $_FILES['attachment']['error'] == 0) {
        $upload_dir = 'uploads/'; // Make sure this directory exists and is
writable
        $upload_file = $upload_dir . basename($_FILES['attachment']['name']);
        $file_type = strtolower(pathinfo($upload_file, PATHINFO_EXTENSION));

        // Validate file type (e.g., allow only certain types of images)
        if (in_array($file_type, ['jpg', 'jpeg', 'png', 'gif'])) {
            if (move_uploaded_file($_FILES['attachment']['tmp_name'],
$upload_file)) {
                $attachment = basename($_FILES['attachment']['name']);
            } else {
```

```
        echo "Error uploading file.";
        exit;
      }
    } else {
      echo "Invalid file type.";
      exit;
    }
  }
$sql = "insert into  case_details (court_type_id, case_type_id, case_no,
court_id, case_status_id, rule_issue_date, interim_order_date,judgment_date,
power_issue_date,petitioners_name,respondent_name,case_summary,attachment,inte
rim_order,lawyer_id,judgment_summary,next_date,Appeal_date,Appeal_order,appeal
_judgment_date,appeal_judgment) values
 ('$court_type_id','$case_type_id','$case_no','$court_id','$case_status_id','$
rule_issue_date','$interim_order_date','$judgment_date','$power_issue_date','$
petitioners_name','$respondent_name','$case_summary','$attachment','$interim_o
rder','$lawyer_id','$judgment_summary','$next_date','$appeal_date','$appeal_or
der','$appeal_judgment_date','$appeal_judgment')";
 //echo  $sql;
 //die;
  $result = $conn->query($sql);

  if($result == 1)
  {
   echo "Successfully Inserted!";
   header("Location:case_details_list_view.php");
  }
 }
?>

<head>
 <?php  include('header_resources.php');   ?>
</head>

<body>
 <section id="container">
```

```
    <?php
     include('header.php');
     include('nav.php');
    ?>
    <!--main content start-->
    <section id="main-content">
      <section class="wrapper">
        <h3> Case Details Information</h3>
        <!-- BASIC FORM ELELEMNTS -->
        <div class="row mt">
          <div class="col-lg-12">
            <div class="form-panel">

              <form class="form-horizontal style-form"
action="case_details_info.php" method="post" enctype="multipart/form-data">
  <div class="form-group">
        <label class="col-sm-2 col-sm-2 control-label">Court Type Name</label>
    <div class="col-sm-6">
    select name="court_type_id" id="court_type_id" onClick="change_fuc();"
style="height:30px; width:600px;">
   <option value='0'>------------Select Any Option----------------</option>
                      <?php
                        foreach($court_type_list as $key => $value)
                        {
                          echo "<option value='$key'>$value</option>";
                        }
                      ?>
                  </select>
                  </div>
                </div>

                <div class="form-group">
```

```
         <label class="col-sm-2 col-sm-2 control-label">Court
Name</label>
         <div class="col-sm-6">
         <select name="court_id" id="court_name" style="height:30px;
width:600px;">
          <option value='0'>------------Select Any Option-----------
-----</option>
           <?php
            foreach($court_name_list as $key => $value)
            {
             echo "<option value='$key'>$value</option>";
            }
           ?>
         </select>
         </div>
        </div>

        <div class="form-group">
         <label class="col-sm-2 col-sm-2 control-label">Justice
Name</label>
         <div class="col-sm-6">
         <select name="court_id" id="justice_name"
style="height:30px; width:600px;">
          <option value='0'>------------Select Any Option-----------
-----</option>
           <?php
            foreach($justice_name_list as $key => $value)
            {
             echo "<option value='$key'>$value</option>";
            }
           ?>
         </select>
         </div>
        </div>
```

```
   <div class="form-group">
      <label class="col-sm-2 col-sm-2 control-label">Case Type Name</label>
       <div class="col-sm-6">
 <select name="case_type_id" id="case_type_id" onClick="change_fuc();"
style="height:30px; width:600px;">
                    <option value='0'>------------Select Any Option-----------
-----</option>
                      <?php
                        foreach($case_type_list as $key => $value)
                        {
                          echo "<option value='$key'>$value</option>";
                        }
                      ?>
                  </select>
                  </div>
                </div>

     <div class="form-group">
     <label class="col-sm-2 col-sm-2 control-label">Case No</label>
                  <div class="col-sm-6">
                    <input type="text" name="case_no" class="form-control">
                  </div>
                  </div>



                  <div class="form-group">
          <label class="col-sm-2 col-sm-2 control-label">Case Summary</label>
                  <div class="col-sm-6">
                 <input type="text" name="case_summary" class="form-control">
                  </div>
                  </div>
                  <div class="form-group">
      <label class="col-sm-2 col-sm-2 control-label">Petitioners Name</label>
```

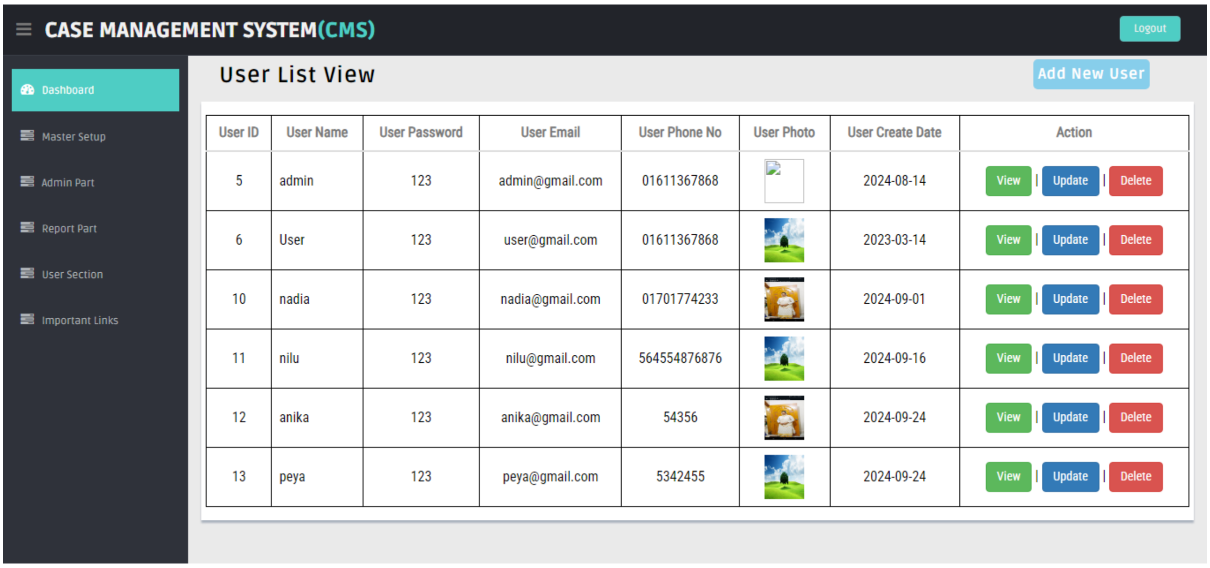
```
       <div class="col-sm-6">
      <input type="text" name="petitioners_name" class="form-control">
                 </div>
                 </div>
<div class="form-group">
<label class="col-sm-2 col-sm-2 control-label">Respondent name</label>
  <div class="col-sm-6">
 <input type="text" name="respondent_name" class="form-control">
                 </div>
 </<div class="form-group">
 <label class="col-sm-2 col-sm-2 control-label">Rule Issue Date</label><div
class="col-sm-6">
<input type="date" name="rule_issue_date" class="form-control">
                 </div>
                </div>
  <div class="form-group">
  <label class="col-sm-2 col-sm-2 control-label">Attachment</label>
  <div class="col-sm-6">
   <input type="file" name="attachment" class="form-control">
                 </div>
                 </div>
<div class="form-group">
<label class="col-sm-2 col-sm-2 control-label">Interim Order Date</label>
<div class="col-sm-6"><input type="date" name="interim_order_date"
class="form-control">
                 </div>
                 </div>
 <div class="form-group">
  <label class="col-sm-2 col-sm-2 control-label">Interim Order</label>
   <div class="col-sm-6"
<input type="text" name="interim_order" class="form-control">
                 </div>
                 </div>
 <div class="form-group">
  <label class="col-sm-2 col-sm-2 control-label">Lawyer Name</label>
   <div class="col-sm-6">
   <select name="lawyer_id" id="lawyer_id" onClick="change_fuc();"
style="height:30px; width:600px;">
 <option value='0'>------------Select Any Option----------------</option>
           <?php
            foreach($lawyer_list as $key => $value)
            {
             echo "<option value='$key'>$value</option>";
            }
           ?>
         </select>
         </div>
        </div>
<div class="form-group">
<label class="col-sm-2 col-sm-2 control-label">Power Issue Date</label>
<div class="col-sm-6">
<input type="date" name="power_issue_date" class="form-control">
         </div>
         </div>
         <div class="form-group">
         <label class="col-sm-2 col-sm-2 control-label">Case Status
Name</label>
         <div class="col-sm-6">
         <select name="case_status_id" id="case_status_id"
onClick="change_fuc();" style="height:30px; width:600px;">
          <option value='0'>------------Select Any Option-----------
-----</option>
           <?php
            foreach($case_status_type_list as $key => $value)
            {
             echo "<option value='$key'>$value</option>";
```

```
                        }
                      ?>
                  </select>
                  </div>
                </div>
               <div class="form-group">
                  <label class="col-sm-2 col-sm-2 control-label">Next
Date</label>
                  <div class="col-sm-6">
                    <input type="date" name="next_date" class="form-control">
                  </div>
                  </div>
<div class="form-group">
  <label class="col-sm-2 col-sm-2 control-label">Judgement Date</label>
     <div class="col-sm-6">
       <input type="date" name="judgment_date" class="form-control">
                  </div>
                  </div>

                  <div class="form-group">
   <label class="col-sm-2 col-sm-2 control-label">Judgement Summary</label>
   <div class="col-sm-6">
     <input type="text" name="judgment_summary" class="form-control">
                  </div>
                  </div>
 <div class="form-group">
  <label class="col-sm-2 col-sm-2 control-label">Appeal Date</label>
   <div class="col-sm-6">
   <input type="date" name="appeal_date" class="form-control">
                  </div>
                  </div>

 <div class="form-group">
```

```
  <label class="col-sm-2 col-sm-2 control-label">Appeal Order</label>
  <div class="col-sm-6">
   <input type="text" name="appeal_order" class="form-control">
                  </div>
                  </div>
<div class="form-group">
 <label class="col-sm-2 col-sm-2 control-label">Appeal Judgment Date</label>
  <div class="col-sm-6">
   <input type="date" name="appeal_judgment_date" class="form-control">
                  </div>
                  </div>
<div class="form-group">
   <label class="col-sm-2 col-sm-2 control-label">Appeal Judgment</label>
     <div class="col-sm-6">
               <input type="text" name="appeal_judgment" class="form-control">
                  </div>
                  </div>

        <div class="form-group">
                  <div class="col-sm-8" align="center">
                    <input type="submit" name="save" class="btn btn-info">
                  </div>
                </div>

              </form>
            </div>
          </div>
          <!-- col-lg-12-->
        </div>

        <!-- /row -->
      </section>
      <!-- /wrapper -->
```

```
    </section>
    <!-- /MAIN CONTENT -->
    <!--main content end-->
    <!--footer start-->

    <!--footer end-->
  </section>

  <?php   include('footer_resources.php');  ?>

</body>

<script>
    function change_fuc()
    {
      const myElement = document.getElementById("court_type_id").value;
      if(myElement == 1 || myElement == 2)
      {
        document.getElementById("court_name").disabled = true;
        document.getElementById("justice_name").disabled = false;
        document.getElementById("court_name").value = "";
      }else{
        document.getElementById("justice_name").disabled = true;
        document.getElementById("court_name").disabled = false;
        document.getElementById("justice_name").value = "";
      }
            //alert(myElement);
    }

  </script>
</html>
```

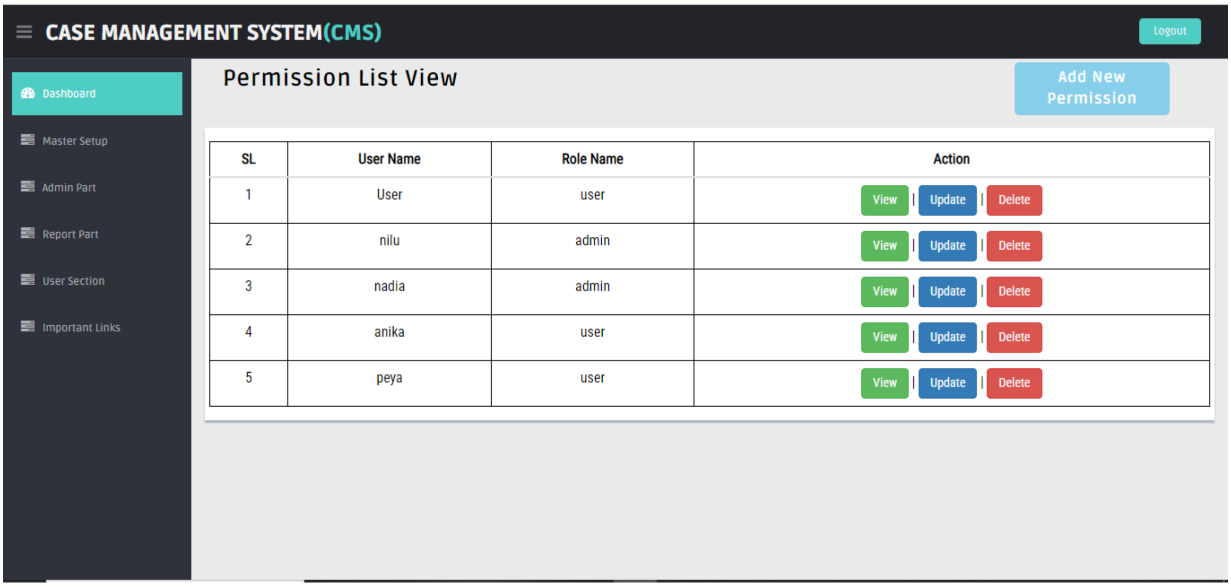
5.1.8 User Role List View



User List View

User ID	User Name	User Password	User Email	User Phone No	User Photo	User Create Date	Action
5	admin	123	admin@gmail.com	01611367868		2024-08-14	View | Update | Delete
6	User	123	user@gmail.com	01611367868		2023-03-14	View | Update | Delete
10	nadia	123	nadia@gmail.com	01701774233		2024-09-01	View | Update | Delete
11	nilu	123	nilu@gmail.com	564554876876		2024-09-16	View | Update | Delete
12	anika	123	anika@gmail.com	54356		2024-09-24	View | Update | Delete
13	peya	123	peya@gmail.com	5342455		2024-09-24	View | Update | Delete

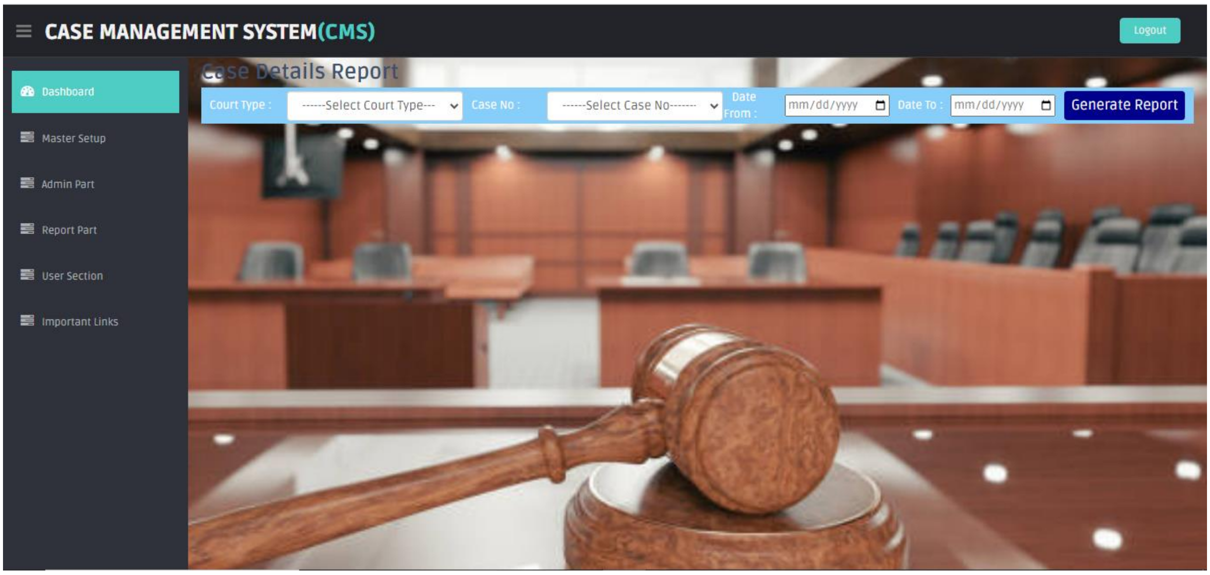
5.1.9 User-wise Permission List View



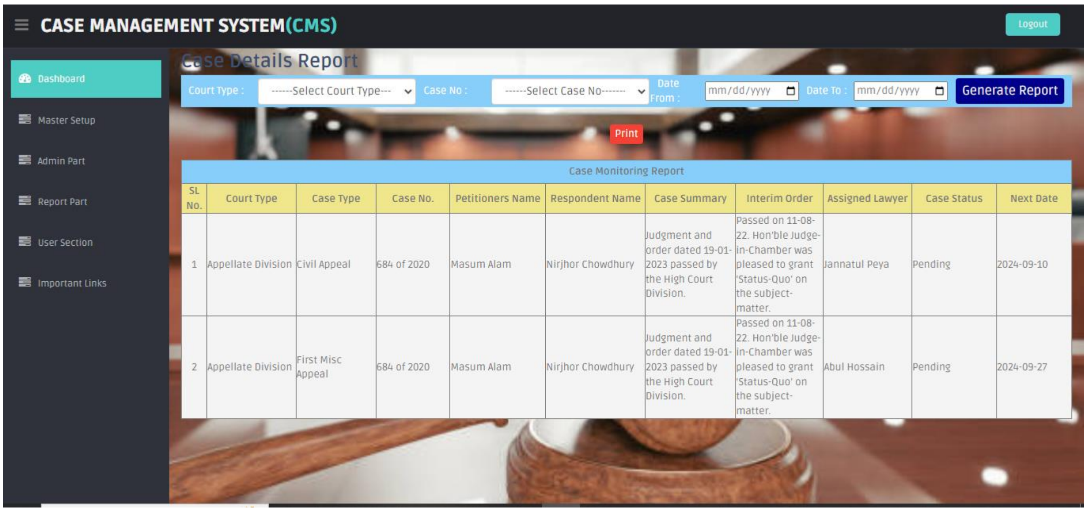
Permission List View

SL	User Name	Role Name	Action
1	User	user	View | Update | Delete
2	nilu	admin	View | Update | Delete
3	nadia	admin	View | Update | Delete
4	anika	user	View | Update | Delete
5	peya	user	View | Update | Delete

5.2 Report Part



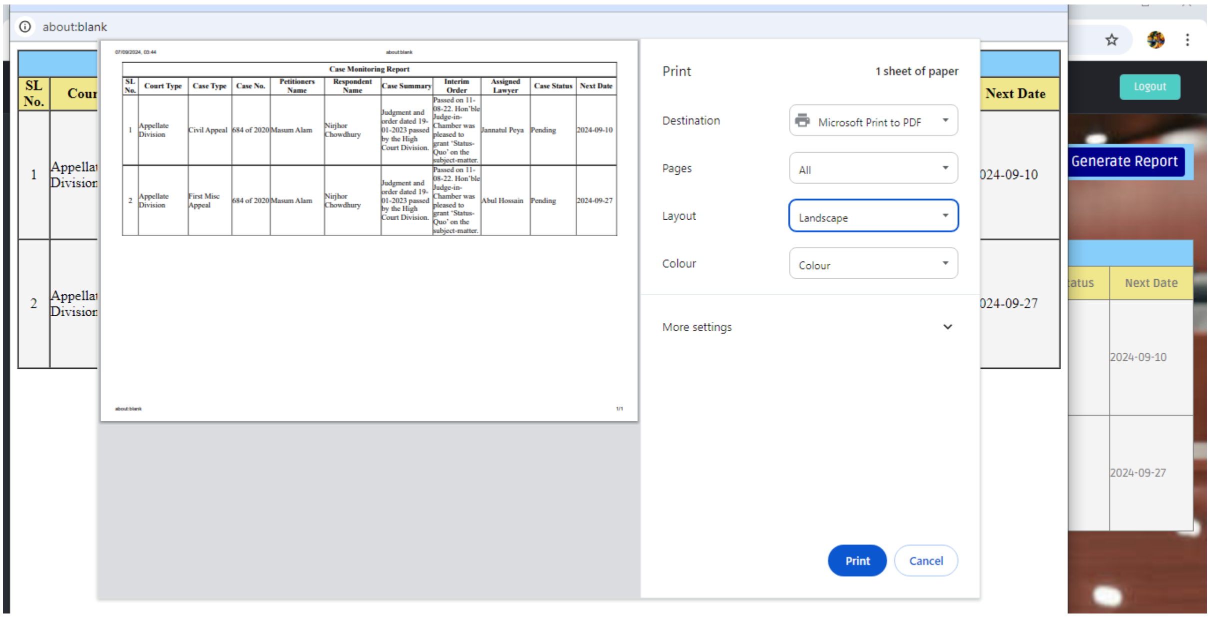
5.2.1 Court Type-Wise Report



Case Monitoring Report

SL No.	Court Type	Case Type	Case No.	Petitioners Name	Respondent Name	Case Summary	Interim Order	Assigned Lawyer	Case Status	Next Date
1	Appellate Division	Civil Appeal	684 of 2020	Masum Alam	Nirjhor Chowdhury	Judgment and order dated 19-01-2023 passed by the High Court Division.	Passed on 11-08-22. Hon'ble Judge-in-Chamber was pleased to grant 'Status-Quo' on the subject-matter.	Jannatul Peya	Pending	2024-09-10
2	Appellate Division	First Misc Appeal	684 of 2020	Masum Alam	Nirjhor Chowdhury	Judgment and order dated 19-01-2023 passed by the High Court Division.	Passed on 11-08-22. Hon'ble Judge-in-Chamber was pleased to grant 'Status-Quo' on the subject-matter.	Abul Hossain	Pending	2024-09-27

5.2.2 Court Type-Wise Report print view



about:blank

07/09/2024, 03:44 about:blank

Case Monitoring Report										
SL No.	Court Type	Case Type	Case No.	Petitioners Name	Respondent Name	Case Summary	Interim Order	Assigned Lawyer	Case Status	Next Date
1	Appellate Division	Civil Appeal	684 of 2020	Masum Alam	Nirjhor Chowdhury	Judgment and order dated 19-01-2023 passed by the High Court Division.	Passed on 11-08-22. Hon'ble Judge-in-Chamber was pleased to grant 'Status-Quo' on the subject-matter.	Jannatul Peya	Pending	2024-09-10
2	Appellate Division	First Misc Appeal	684 of 2020	Masum Alam	Nirjhor Chowdhury	Judgment and order dated 19-01-2023 passed by the High Court Division.	Passed on 11-08-22. Hon'ble Judge-in-Chamber was pleased to grant 'Status-Quo' on the subject-matter.	Abul Hossain	Pending	2024-09-27

about:blank 1/1

Print 1 sheet of paper

Destination: Microsoft Print to PDF

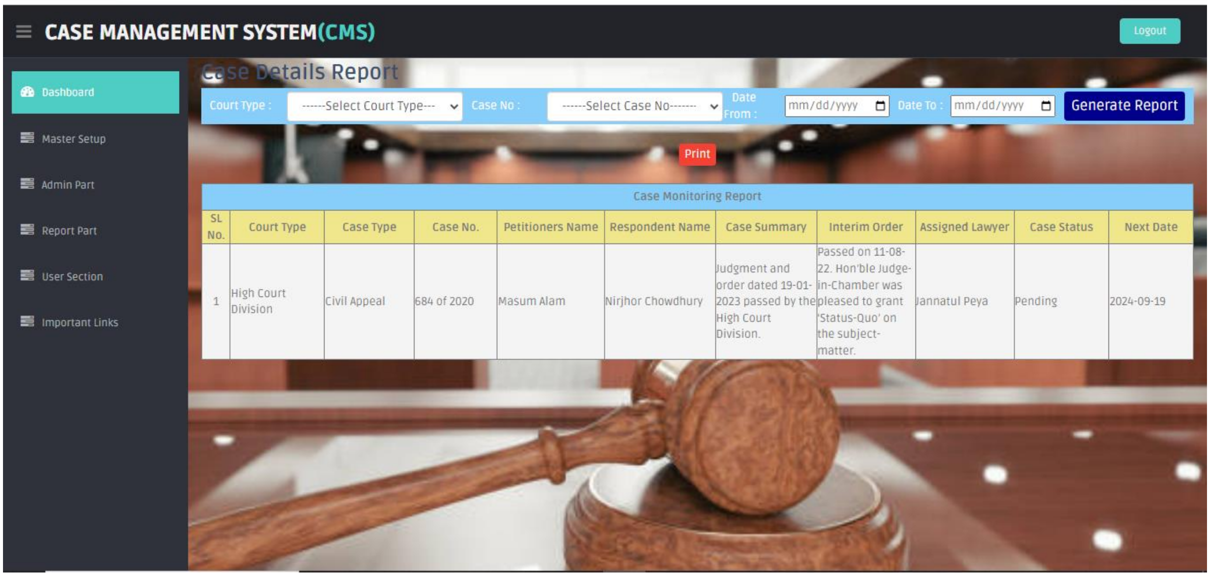
Pages: All

Layout: Landscape

Colour: Colour

More settings

5.2.3 Case No-wise Report



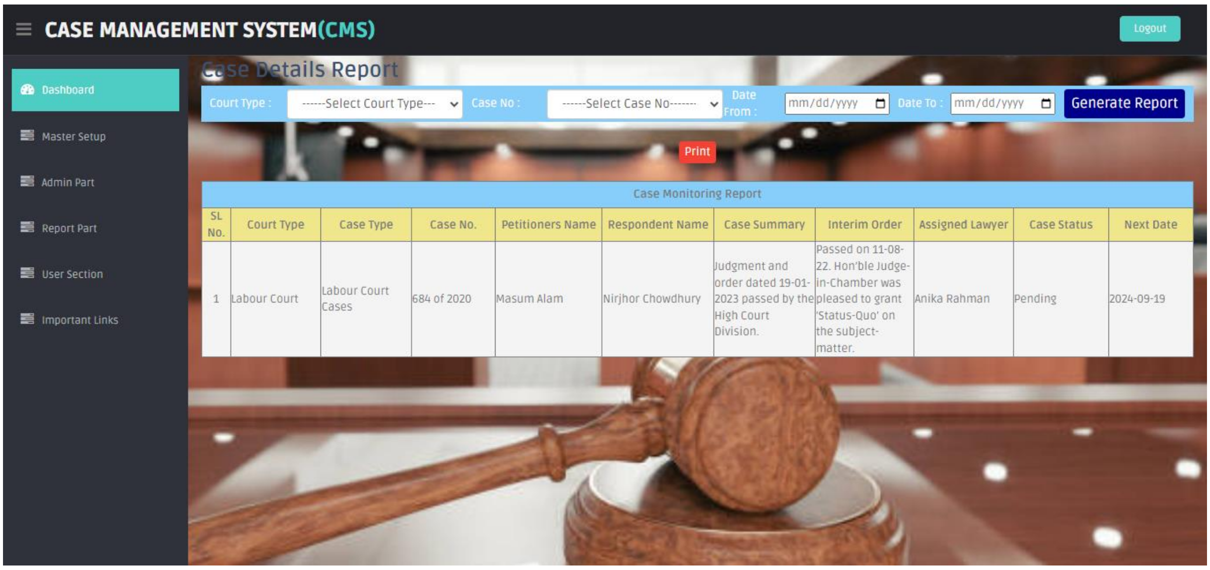
CASE MANAGEMENT SYSTEM(CMS)

Case Details Report

Court Type : ------Select Court Type---- Case No : ------Select Case No------- Date From : mm/dd/yyyy Date To : mm/dd/yyyy

Case Monitoring Report										
SL No.	Court Type	Case Type	Case No.	Petitioners Name	Respondent Name	Case Summary	Interim Order	Assigned Lawyer	Case Status	Next Date
1	High Court Division	Civil Appeal	684 of 2020	Masum Alam	Nirjhor Chowdhury	Judgment and order dated 19-01-2023 passed by the High Court Division.	Passed on 11-08-22. Hon'ble Judge-in-Chamber was pleased to grant 'Status-Quo' on the subject-matter.	Jannatul Peya	Pending	2024-09-19

5.2.4 Date Range-wise Report



Case Monitoring Report

SL No.	Court Type	Case Type	Case No.	Petitioners Name	Respondent Name	Case Summary	Interim Order	Assigned Lawyer	Case Status	Next Date
1	Labour Court	Labour Court Cases	684 of 2020	Masum Alam	Nirjhor Chowdhury	Judgment and order dated 19-01-2023 passed by the High Court Division.	Passed on 11-08-22. Hon'ble Judge-in-Chamber was pleased to grant 'Status-Quo' on the subject-matter.	Anika Rahman	Pending	2024-09-19

5.3 Important Links

Some important links are also included in this software for quick access i.e. https://www.supremecourt.gov.bd/, https://judiciary.gov.bd/, http://bdlaws.minlaw.gov.bd/ and https://attorneygeneral.gov.bd/.

Chapter-6

Database Table Design

Chapter-6: Database Table Design

6.1 XAMPP Server Database Table Design View

In XAMPP Server we created a database for our Project named ‘cms’. This database will contain all the data of Case Management System (backend) and Website (frontend).

6.2 Open a Connection to MySQL database

Before we can access data in the MySQL database, we need to be able to connect to the server:

```
<?php
        $servername = "localhost";
        $username      = "root";
        $password      = "";
        $dbname        = "cms";
        $conn = new mysqli($servername, $username, $password, $dbname);?>
```

6.3 Case Management System Database Table Design

There are Nine (09) tables has been created so far in our Case Management System database. They are:

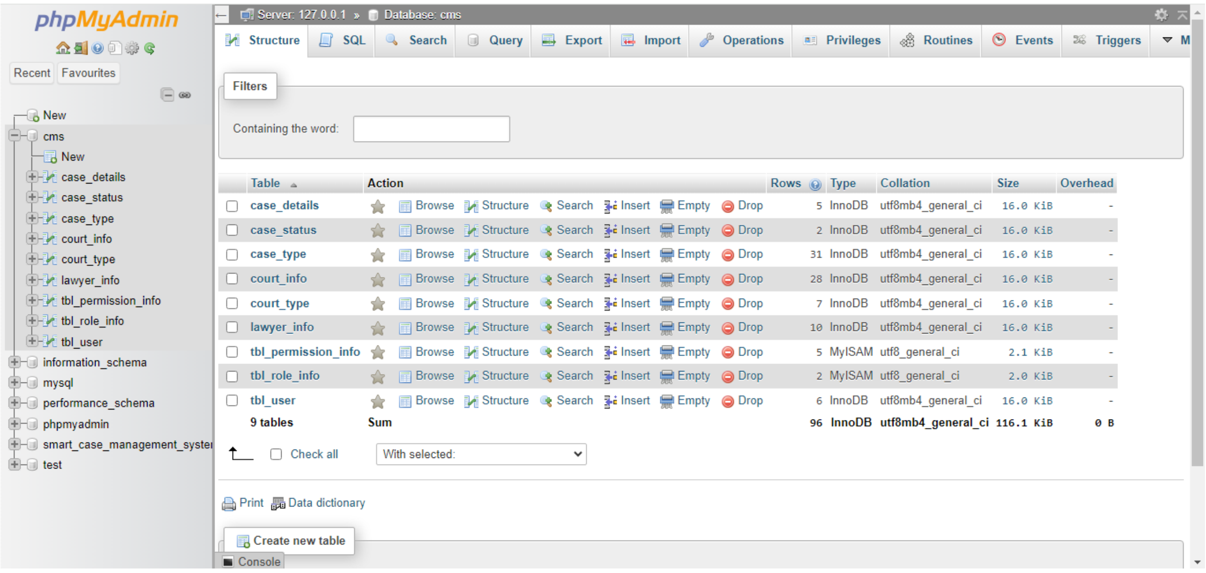
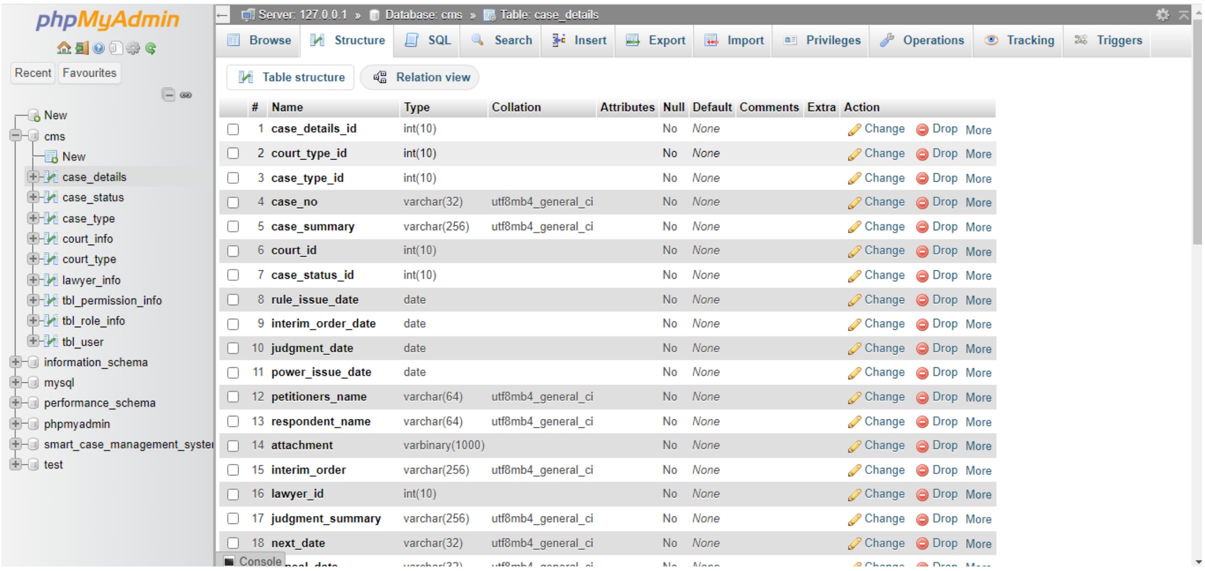


	Table	Action	Rows	Type	Collation	Size	Overhead
□	case_details	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	case_status	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	case_type	Browse Structure Search Insert Empty Drop	31	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	court_info	Browse Structure Search Insert Empty Drop	28	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	court_type	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	lawyer_info	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	tbl_permission_info	Browse Structure Search Insert Empty Drop	5	MyISAM	utf8_general_ci	2.1 KiB	-
□	tbl_role_info	Browse Structure Search Insert Empty Drop	2	MyISAM	utf8_general_ci	2.0 KiB	-
□	tbl_user	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	9 tables	Sum	96	InnoDB	utf8mb4_general_ci	116.1 KiB	0 B

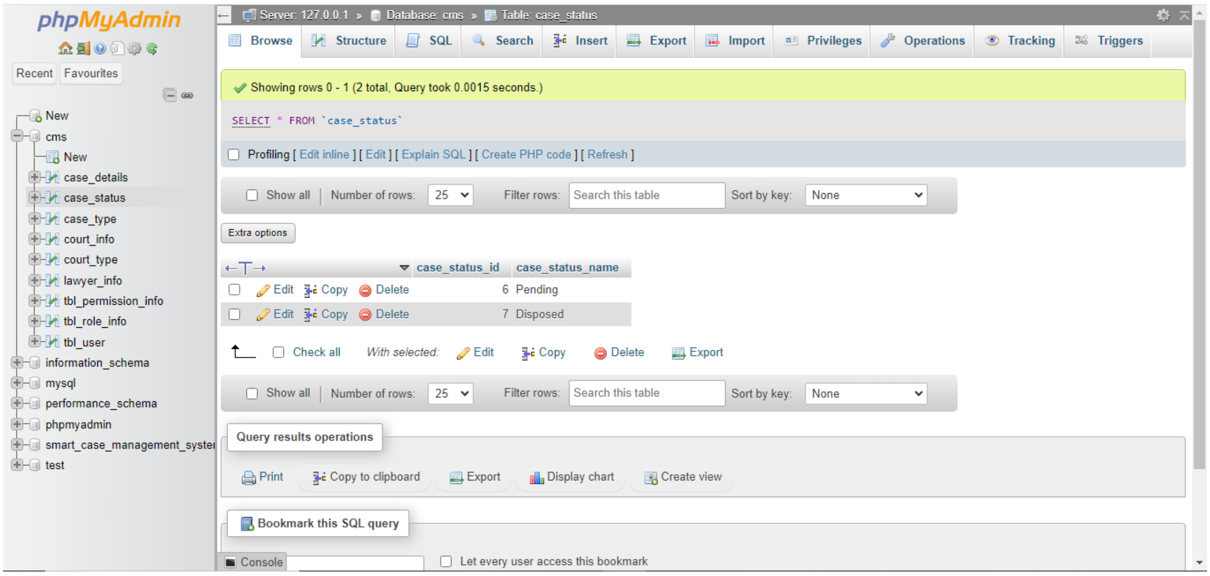
Here is the above-mentioned the phpMyAdmin interface of the database (Case Management System) where all tables are shown:

6.4 Case Details Table Design



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	case_details_id	int(10)			No	None			Change Drop More
2	court_type_id	int(10)			No	None			Change Drop More
3	case_type_id	int(10)			No	None			Change Drop More
4	case_no	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More
5	case_summary	varchar(256)	utf8mb4_general_ci		No	None			Change Drop More
6	court_id	int(10)			No	None			Change Drop More
7	case_status_id	int(10)			No	None			Change Drop More
8	rule_issue_date	date			No	None			Change Drop More
9	interim_order_date	date			No	None			Change Drop More
10	judgment_date	date			No	None			Change Drop More
11	power_issue_date	date			No	None			Change Drop More
12	petitioners_name	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More
13	respondent_name	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More
14	attachment	varbinary(1000)			No	None			Change Drop More
15	interim_order	varchar(256)	utf8mb4_general_ci		No	None			Change Drop More
16	lawyer_id	int(10)			No	None			Change Drop More
17	judgment_summary	varchar(256)	utf8mb4_general_ci		No	None			Change Drop More
18	next_date	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More
	peal_date	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More

6.5 Case Status Table Design



✓ Showing rows 0 - 1 (2 total, Query took 0.0015 seconds.)

```
SELECT * FROM `case_status`
```

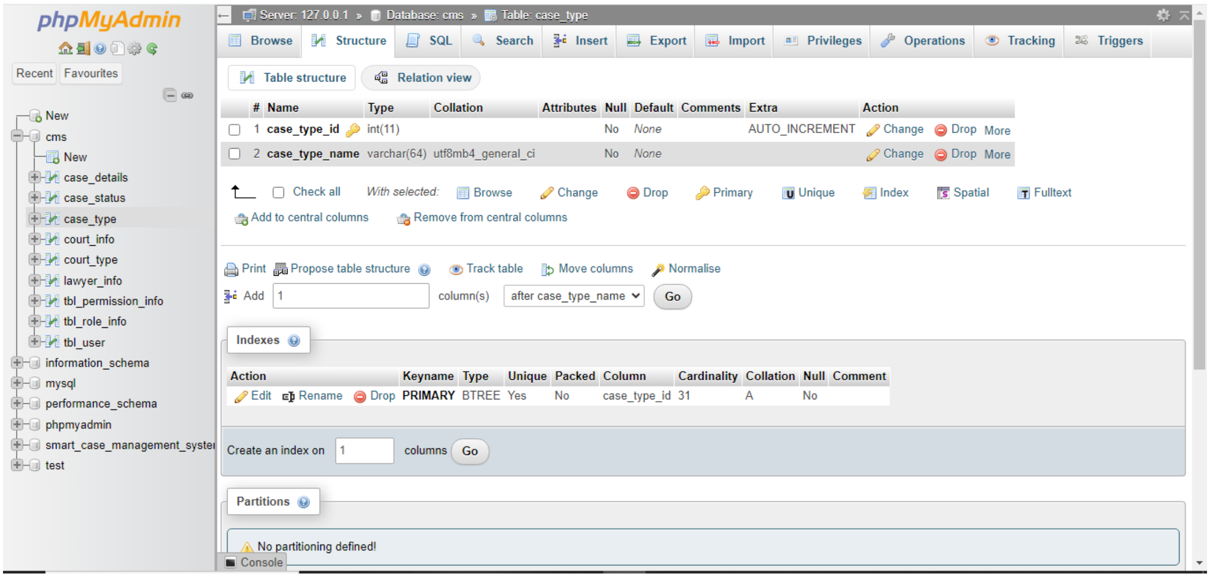
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

		case_status_id	case_status_name
□	Edit Copy Delete	6	Pending
□	Edit Copy Delete	7	Disposed

□ Let every user access this bookmark

6.6 Case type Table Design



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	case_type_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	case_type_name	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More

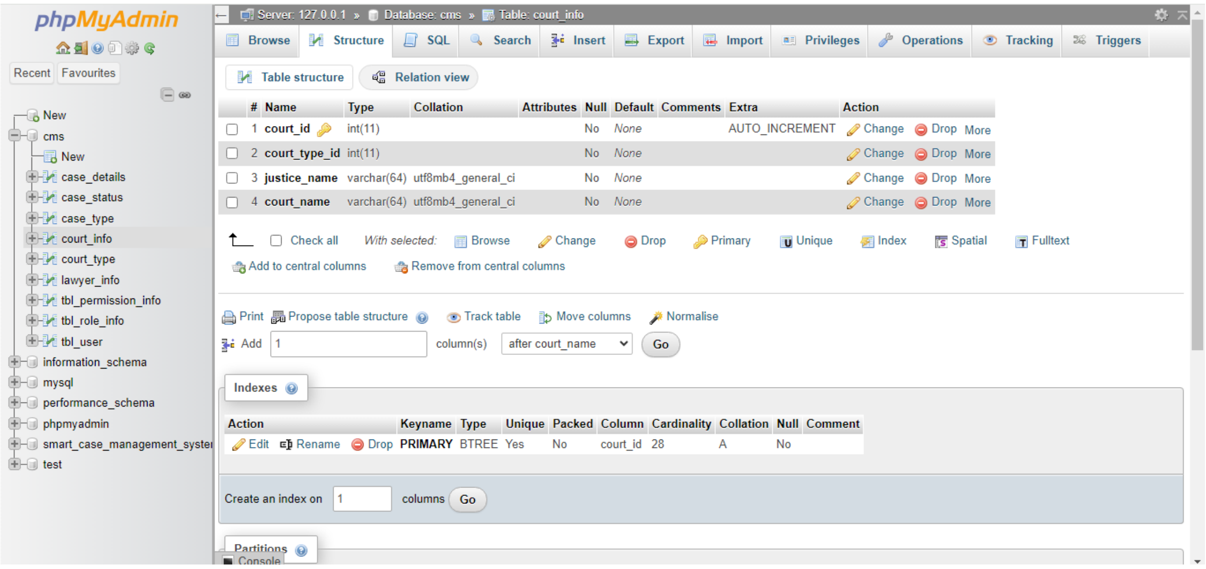
Add 1 column(s) after case_type_name

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	case_type_id	31	A	No	

Create an index on 1 columns

No partitioning defined!

6.7 Court Info Table Design



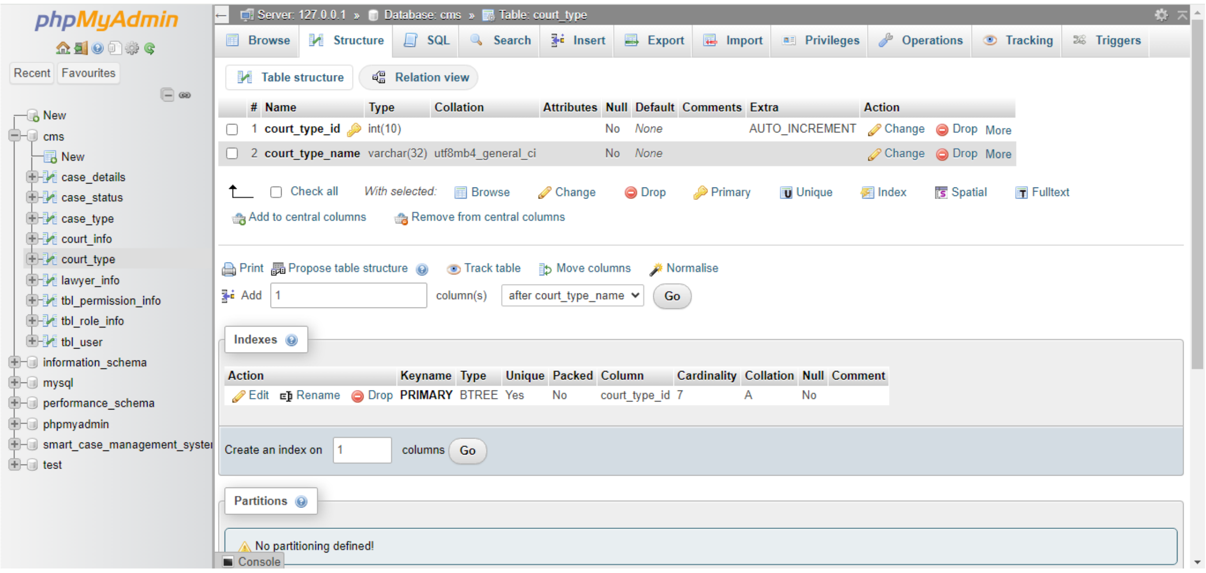
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	court_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	court_type_id	int(11)			No	None			Change Drop More
3	justice_name	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More
4	court_name	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More

Add 1 column(s) after court_name

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	court_id	28	A	No	

Create an index on 1 columns

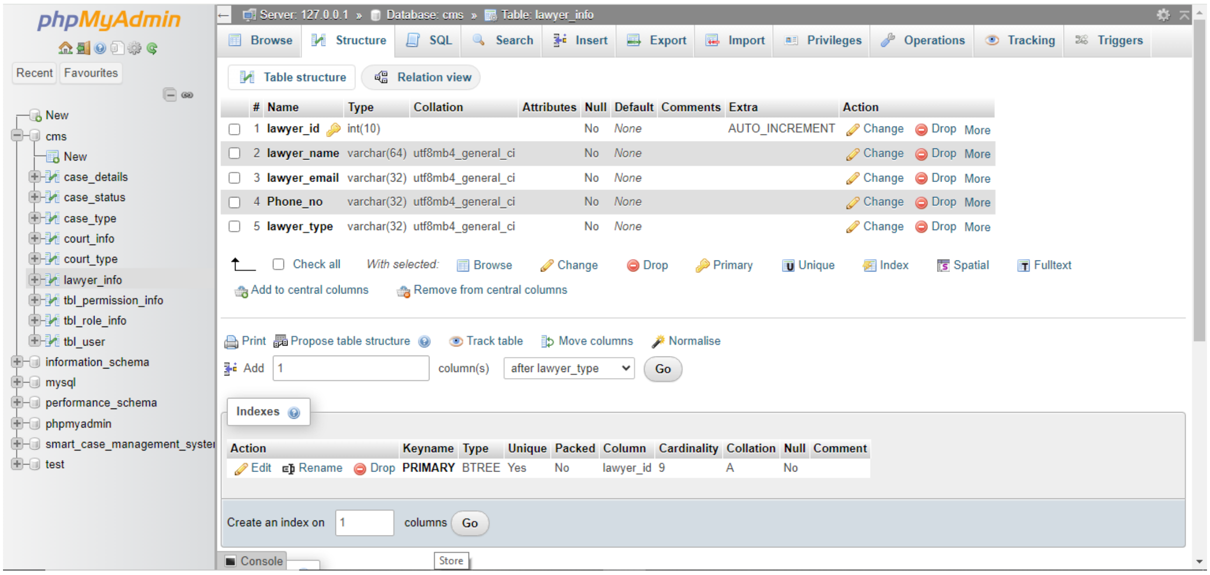
6.8 Court type Table Design



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	court_type_id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	court_type_name	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	court_type_id	7	A	No	

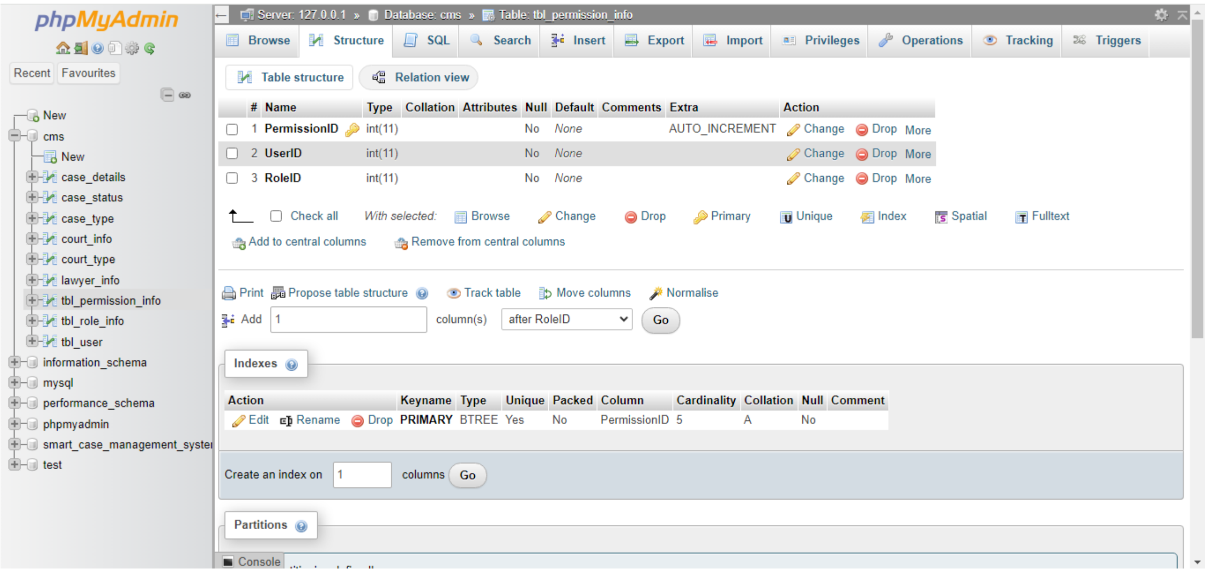
6.9 Lawyer Info Table Design



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	lawyer_id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	lawyer_name	varchar(64)	utf8mb4_general_ci		No	None			Change Drop More
3	lawyer_email	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More
4	Phone_no	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More
5	lawyer_type	varchar(32)	utf8mb4_general_ci		No	None			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	lawyer_id	9	A	No	

6.10 Permission info table Design



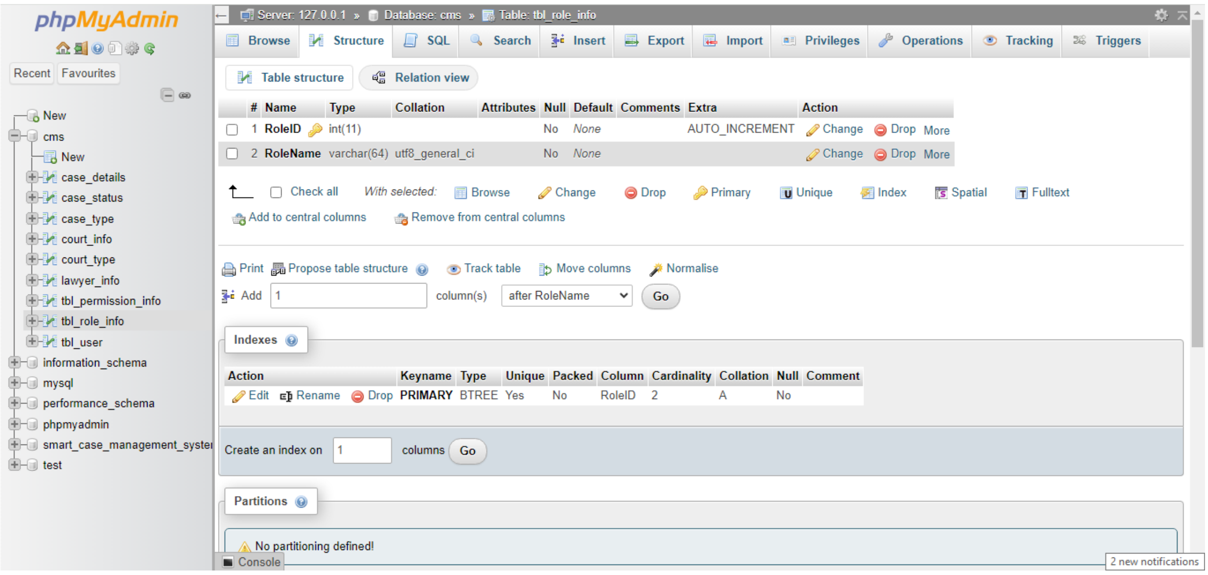
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PermissionID	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	UserID	int(11)			No	*None*			Change Drop More
3	RoleID	int(11)			No	*None*			Change Drop More

Add 1 column(s) after RoleID Go

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	PermissionID	5	A	No	

Create an index on 1 columns Go

6.11 Role info table Design



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	RoleID	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	RoleName	varchar(64)	utf8_general_ci		No	*None*			Change Drop More

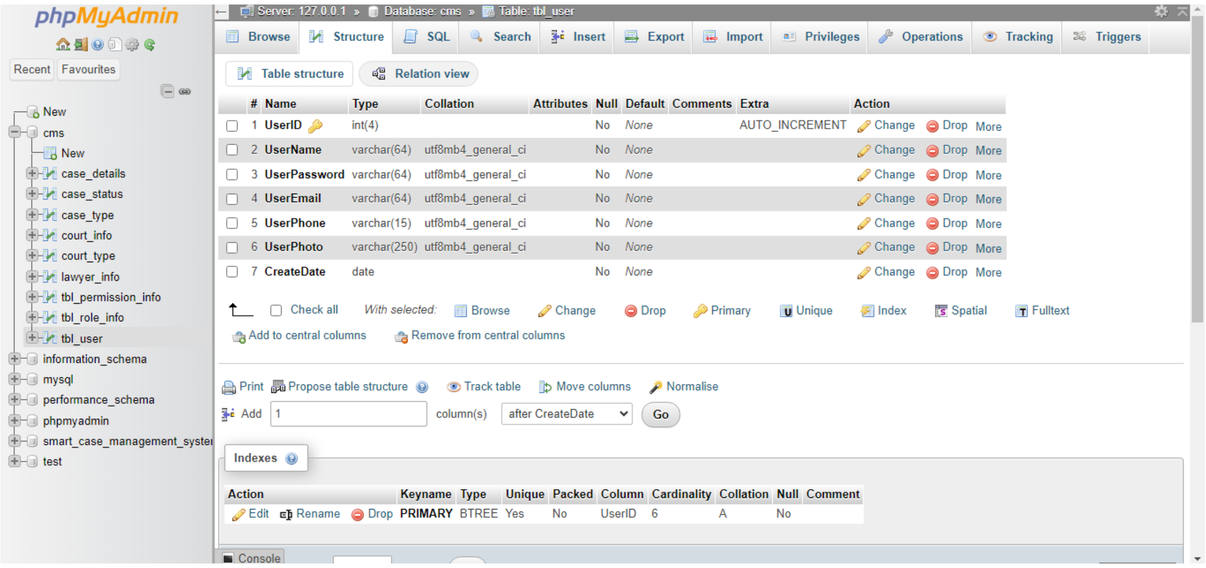
Add 1 column(s) after RoleName Go

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	RoleID	2	A	No	

Create an index on 1 columns Go

No partitioning defined!

6.12 User info table Design



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	UserID	int(4)			No	*None*		AUTO_INCREMENT	Change Drop More
2	UserName	varchar(64)	utf8mb4_general_ci		No	*None*			Change Drop More
3	UserPassword	varchar(64)	utf8mb4_general_ci		No	*None*			Change Drop More
4	UserEmail	varchar(64)	utf8mb4_general_ci		No	*None*			Change Drop More
5	UserPhone	varchar(15)	utf8mb4_general_ci		No	*None*			Change Drop More
6	UserPhoto	varchar(250)	utf8mb4_general_ci		No	*None*			Change Drop More
7	CreateDate	date			No	*None*			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	UserID	6	A	No	

6.13 PHP MySQL Database

With PHP, we can connect to and manipulate databases. MySQL is the most popular database system used with PHP.

- MySQL is a database system used on the web;
- MySQL is a database system that runs on a server;
- MySQL is ideal for both small and large applications;
- MySQL is very fast, reliable, and easy to use;
- MySQL uses standard SQL;
- MySQL compiles on a number of platforms;
- MySQL is free to download and use;
- MySQL is developed, distributed, and supported by Oracle Corporation.

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful for storing information categorically (Welling, Luke; Thomson).

Here are some syntax rules to follow:

- The SQL query must be quoted in PHP;
- String values inside the SQL query must be quoted;
- Numeric values must not be quoted;
- The word NULL must not be quoted.

6.14 PHP + MySQL Database System

PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform) (Welling, Luke; Thomson).

6.15 Database Queries:

A query is a question or a request.

We can query a database for specific information and have a record set returned.

Look at the following query (using standard SQL):

```
SELECT lawyer_name FROM lawyer_info;
```

The query above selects all the data in the " lawyer_name" column from the "lawyer_info" table.

Chapter-7

Application Testing

Chapter-7: Application Testing

7.1 Introduction

Software testing is the process of executing a program with intension of finding errors in the code. It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not. Generally, system is perfect due to communication problems between user and developer, time constraints or conceptual mistakes by developer. The purpose of system testing is to check and find out these errors or faults as early as possible so losses due to it can be saved. Testing is the fundamental process of software success.

Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases i.e. design development and maintenance phase. Testing is used to show incorrectness and considered to success when an error is detected.

7.2 Objectives of Software Testing

The software testing is usually performed for the following objectives:

- **Software Quality Improvement:** The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So, great consideration is required for checking for quality of software.
- **Verification & Validation:** Verification means to test that we are building the product in right way i.e. are we using the correct procedure for the development of software so that it can meet the user requirements. Validation in means to check whether we are building the right product or not.
- **Software Reliability Estimation:** The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software reliability (Paul C. Jorgensen).

7.3 Principles of Software Testing

Software testing is an extremely creative and challenging task. Some important principles of software testing are as given: - All tests should be traceable to Customer requirements. Testing time and resources should be limited i.e. avoid redundant testing. It is impossible to test everything. Use effective resources to test. Test should be planned long before testing begins i.e. after requirement phase. Test for invalid and unexpected input conditions as well as valid conditions. Testing should begin in "in the small" and progress towards testing in the large". For the most effective testing should be conducted by an independent party. Keep software static (without change meanwhile) during test. Document test cases and test results. Examining what the software not doing which it expected to do and also checking what it is doing that was not expected to do(Paul C. Jorgensen).

7.4 Strategy for Software Testing

Different levels of testing are used in the test process each level of testing aims to test different aspects of the system. The first level is unit testing. In this testing 29 individual components are tested to ensure that they operate correctly. It focuses on verification efforts. The second level is integration testing. It is a systematic technique for constructing the program structure. In this testing, many tested modules are combined into the subsystem which is then tested. The good here is to see if the modules can be integrated properly. Third level is integration testing. System testing is actually a series of different tests whose primary purpose is to fully exercise computer-based system. These tests fall outside scope of software process and are not conducted solely by software engineers.

- **Black Box Testing:** Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing (Tim Koomen)

- **White Box Testing:** White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for Verification (P. C. Jorgensen).

- **Unit Testing:** The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by java. The various controls are tested to ensure that each performs its action as required (Paul C. Jorgensen).

- **Integration Testing:** Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may float produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole (M. Ramachandran, S. Muthuvelayutham).

- Here the Server Module and Client module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data (Paul C. Jorgensen) .

- **Functional Testing:** Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing (Paul C. Jorgensen).

- **System Testing:** System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing (Paul C. Jorgensen).

- **Stress Testing:** Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

- **Performance Testing:** Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

- **Usability Testing:** Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

- **Acceptance Testing:** Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing (Paul C. Jorgensen).

- **Regression Testing:** Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing (Paul C. Jorgensen).

Chapter-8

Future Works Development

Chapter-8: Future Works Development

8.1 Introduction

In this project I have represented an efficient way of developing dynamic website by using power of PHP and My SQL. Still few limitations exist in the software; my aim is to develop this web site up to highly useful.

On this topic further study may be needed for more development. To make it global, easier and dynamic following study can be made:

Develop it for dynamic use not only used by internet, also by mobile communication

every user transaction can be possible by online and confirmation also possible by online system every user of the whole world can get help from this site.

8.2 List of Future Works Development:

1) Advanced Analytics and Predictive Insights
2) IoT (Internet of Things) Integration
3) Mobile Applications and Remote Access
4) Supply Chain Integration
5) Blockchain for Traceability and Transparency
6) Continuous Improvement and Innovation
7) Scalability and Global Expansion
8) Website Security Check
9) Invoice Emailer
10) Email & Mobile verification
11) Overall design development and bug fix
12) User Database Re-Design, User Role Implementation

Chapter-9

Conclusion

Chapter-9: Conclusion

The development and implementation of a Case Management System (CMS) represent a significant advancement in the efficiency and effectiveness of Organizations. As outlined in this project paper, the proposed CMS addresses critical challenges faced by the court or administrative Personnel, including inefficiencies in case handling, lack of transparency, data management issues, and integration difficulties.

The successful deployment of the CMS will require careful planning and execution, including detailed requirement analysis, system design, and development. Adequate training for users and administrators is essential to ensure smooth adoption and utilization of the system. Ongoing support and maintenance will be crucial to address any issues and incorporate user feedback for continuous improvement.

In conclusion, the proposed Court Case Management System offers a transformative solution to the challenges faced by the different organizations. By adopting this system, courts can significantly improve their operational efficiency, ensure better transparency and accountability, and ultimately enhance the delivery of justice. The successful implementation of the CMS will set a precedent for modernizing judicial functional systems and pave the way for future advancements in court case management.

References

References:

Books and Academic Journals

"Court Management: A Comprehensive Guide" by John W. R. Clegg and Jonathan P. A. Gill

"Information Technology for Management: Transforming Organizations in the Digital Economy" by Efraim Turban, Linda Volonino

"Managing and Improving Process Performance in Court Systems" by M. S. S. White and G. J. Morris

"E-Justice: The Role of Technology in the Courtroom" by Richard Susskind

"Database Systems: The Complete Book" by Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom

Technical Papers and Articles

"Design and Implementation of an E-Court Case Management System" by L. Yang and X. Zhang, *International Journal of Information Technology & Decision Making*, 2020.

"Case Management in E-Government: Design Principles for a Case Management System" by J. P. B. Hughes and R. A. Johnson, *Journal of E-Government Research*, 2021.

"Challenges and Solutions in Court Case Management Systems" by N. K. Prasad and R. Sharma, *Journal of Legal Technology*, 2019.

Standards and Guidelines

"ISO/IEC 27001:2013 - Information Security Management Systems" by International Organization for Standardization

"NIST Special Publication 800-53: Security and Privacy Controls for Information Systems and Organizations" by National Institute of Standards and Technology

Government and Industry Reports

"The State of Technology in the Judiciary" by the National Center for State Courts (NCSC), 2022.

"E-Court Solutions: Trends and Case Studies" by the Bureau of Justice Assistance Online Resources and Websites

"Court Technology: A Guide for the 21st Century" by the Court Technology Laboratory, National Center for State Courts (NCSC)

"Legal Tech News and Updates" by Law Technology Today

"Legal Case Management Software Reviews" on Capterra or G2

Website Rreferences:

https://www.db-book.com/db5/slide-dir/

https://itsourcecode.com/uml/e-commerce-website-dfd-levels-0-1-2-best-dataflow-diagrams-for-2021/

https://library.concordia.ca/help/citing/mla.php?guid=book-one-author

https://www.php.net/

https://stackoverflow.com/

www.tutorialspoint.com

www.w3schools.com