PROJECT SYNOPSIS:

Introduction:

This software is very useful for Managing the daily cases diary for Lawyers/Advocates. This is of software, which store information about an Advocates Clients, Opponents, Case No, Previous Date, Current Date, Next Date, Court Name. You can filter the data according to case number, client parties, next date of hearing.

Software manages day to day activity and builds comprehensive client/case/matter database and history that can help improve every aspect of your practice. You will have immediate access to information. The software is very easy to operate and light on system tool to maintain information about clients, cases, hearings, books etc..

Objectives of the project **:**

This software is very easy to use. The main goal is reducing paperwork. It helps us to recollect the information about the previous and future case Information

Existing System :

In the existing system the details of the client entered manually. A case diary is used for maintained the details of client. All day to day work is maintained in different registers

There is no maintenance of the records of payment.

There is all paper work. It is difficult to explain the history of cases.

Advantages Of Proposed System :

1. **Time Saving.**

Since all the client details and related information are store in the system it enables its significant reduction in the number of registration taken up for the storage and retrieval of information.

1. **Accuracy in Reliability**.

Problems related with data loss and inconsistency can be eliminated , thus making the system more accurate and reliable.

1. **Information maintenance.**

It can now be an easy task as it removes the large registration from the scene.

1. **Availability of reports.**

Reports can be generated manually.

Input Of The Project :

* Details of clients.
* Details of case
* Case schedule-time and date of case.
* Adding new case details.
* Advocate details.
* Court details.

Output Of The Project :

* Admin view.
* Court profile.
* Client details view.
* Schedule view.
* Advocate view.
* Case report.
* Judgement report.

**Limitations:**

1. Multiple language is not supported.
2. Presently it cannot be implemented as android application.
3. Not supports for multiple parties.
4. Currently report generation in the form of pdf is not available

Future:

1. In future it can be implemented as android application.
2. Report generation can be found in the form of pdf.

**Scope of the project :**

* Keeping all records of client and court cases along with the personal details
* Maintaining day to day information about

Client and cases

* Maintaining all sorts of records in databases.
* Storing information about the client payment transacting.
* Storing information about old cases and court hearings
* Provide search facility for searching desired data.
* Provide report of all transactions.
* Reminders can be provided as sms or email for related advocate and client.

**Why You Need XAMPP, MySQL, and PHP?**

PHP is a powerful scripting language that can be run by itself in the command line of any computer with PHP installed. However, PHP alone isn't enough in order to build dynamic web sites. To use PHP on a web site, you need a server that can process PHP scripts. XAMPP server allows developers to test PHP scripts locally; this makes it an invaluable piece of your local development environment.

Additionally, dynamic websites are dependent on stored information that can be and easily; this is the main difference between a dynamic site and a static HTML site. However, PHP doesn’t provide a simple, efficient way to store data. This is where a relational database management system like MySQL comes into play.

**PHP :**

PHP originally stood for “Personal Home Page” and was released as a

Free, open source project. Over time, the language was reworked to meet the needs of its users. In 1997, PHP was renamed to the current “PHP: Hypertext Pre-processor.” PHP is generally used as a server-side scripting language; it is especially well-suited for creating dynamic web pages and client-side GUI applications. . PHP generally runs on a web server, taking PHP code as its input and creating web pages as output. The scripting language features integrated support for interfacing with databases such as MySQL, which makes it a prime candidate for building all manner of web applications, from simple personal web sites to complex enterprise-level applications.

Unlike HTML, which is parsed by a browser when a page loads, PHP is preprocessed by the machine that serves the document (this machine is referred to as a server). All PHP code contained with the document is processed by the server before the document is sent to the visitor’s browser. PHP is a scripted language, which is another great advantage for PHP programmers. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use. Many programming languages require that you compile files into machine code before they can be run, which is a time-consuming process. Bypassing the need to compile means you’re able to edit and test code much more quickly Because PHP is a server-side language, running PHP scripts on your local machine requires installing a server on your local machine.

PHP is free software released under the PHP License. It is incompatible with the GNU General Public License (GPL), due to restrictions on the usage of the term PHP. It is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as its input and creating web pages as output. It can be deployed on most web servers and on almost every operating system and platform free of charge. PHP is installed on more than 20 million websites and 1 million web servers.

**Usage:**

PHP is a general-purpose scripting language that is especially suited for web development. PHP generally runs on a web server, taking PHP code as its input and creating web pages as output. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. It can automatically detect the language of the user. From PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor. Originally designed to create dynamic web pages, PHP’s principal focus is sever side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft’s Active Server Pages, Sun Microsystems’ Java Server Pages . PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include Cake PHP , CodeIgniter, and Zend Framework, offering features similar to other web application frameworks.

**Speed Optimization:**

As with many scripting languages, PHP scripts are normally kept as

human-readable source code, even on production web servers. In this case, PHP scripts will be compiled at runtime by the PHP engine, which increases their execution time. PHP scripts are able to be compiled before runtime using PHP compilers as with other programming languages such as C (the language PHP and its extensions are written in). Code optimizers aim to reduce the computational complexity of the compiled code by reducing its size and making other changes that can reduce the execution time with the overall goal of improving performance. The nature of the PHP compiler is such that there are often opportunities for code optimization, and an example of a code optimizer is the Zend Optimizer PHP extension.

Another approach for reducing overhead for high load PHP servers is using PHP accelerators. These can offer significant performance gains by caching the compiled form of a PHP script in shared memory to avoid the overhead of parsing and compiling the code every time the script runs.

**HTML:**

HTML means Hypertext Markup Language. HTML is a method of describing the format of document, which allows them to be viewed on computer screen. Web browsers display HTML documents, program which can navigate across networks and display a wide variety of types of information. HTML tag can be developed to be simple text or to be complex multimedia extra advantages containing, moving images, virtual reality, and java applets.

The global publishing format of the Internet is HTML. It allows authors to use not only text but also format that text with headings, list and tables, and also includes still images videos, and sound within text. Readers can access pages information from anywhere in the world at the click of mouse button information can be downloaded to readers own PC or workstations HTML pages can also be used for entering a data and as a front end for commercial transaction.

**Create a connection to a database:**

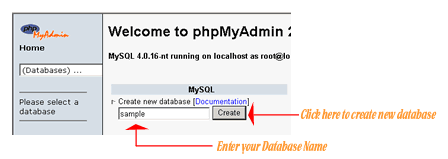
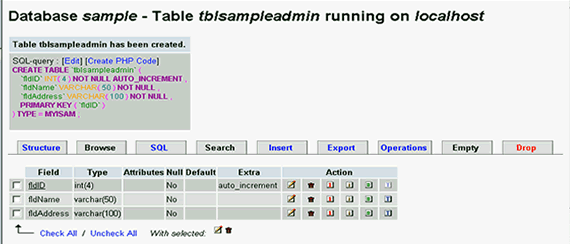
Before you can access data in a database, you must create a connection to the database. In PHP, this is done with the My SQL connect() function.

Syntax: My SQL connect (server name, username, password);

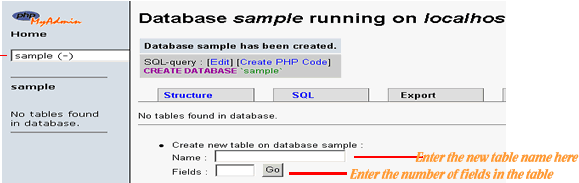
Server name: Optional Specifies the Server to connect .Default values is localhost: 3306

**Steps to create a database in PHP My Admin:**

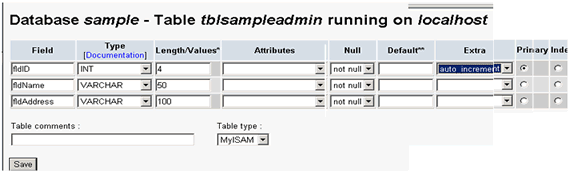
1. The following figure shows your PHP My Admin interface, just enter your database name and click the 'Create' button to create your database.



2) Now to create a new table enter your table name and the number of fields in the table, then click the 'Go' Button.



3) The next step is to create the fields, just enter values for each field name, type, length of the field, null option and mention whether it is a primary key or not. Then click the 'Save' button to complete your table creation.



4) The following figure is displayed upon successful creation of your table

**Database Evolution**:

SQL was invented back in the 1960's by E.F. Cod of IBM. in order to increase data integrity and reduce repetitive data. RDBMS systems didn't appear until the late 70's when Sybase and Oracle introduced systems. These systems existed on mainframes at the time.

ANSI-SQL came to be in the 1980's. This was important because it meant that disparate systems could communicate through an agreed set of standards. There are different levels of ANSI-SQL compliance. Almost every major RDBMS today is entry level compliant, including SQL Server 2000. Every RDBMS has its own flavour of SQL that complements ANSI-SQL with proprietary elements. SQL Server's flavour of SQL is known as Transact SQL (T-SQL).

SQL Server was originally a Sybase product. Microsoft bought the product outright from Sybase and by version 7.0, the version prior to 2000, all the code had been rewritten by Microsoft's programming gurus

FEATURES OF SQL :

* It is simple English like language and uses simple commands such as SELECT, CREATE, DROP etc

.

* It is not having condition loops, variables and most of the commands are single line commands.
* To implement application logics, SQL has got extension language
* popularly called as PL/SQL (Procedural language of sql).
* One of the key features of sql server is the XML support. XML has

Grown to be standard technology for organizations that share data on the web.

* The entire SQL has been divided into 4 major categories.

1. Data Manipulation Language.

2. Data Definition Language.

3. Transaction Control language.

4. Data Control Language.

**PROJECT SUBJECT:**

In this Advocate Case Diary client makes enquiry to the employee of the case details and appointments then the employee checks for the availability of the dates. If the dates are available then the meetings and court related information is gathered and fixed. This process is going on and lastly the bill is generated according to the payment that is needed to be done for the advocate. Also they can change the information of the client (if necessary)

Again there is an Administrator. He or she is also having some responsibility regarding to the Advocate Management System Software. His or her work is to calculate the salary of the employee working under the Head Advocate and also be done with the job of attendance and for security purpose they can change their password. The employees cannot access the information regarding to the administrator. This is done for the security purpose.

**SYSTEM FEATURES:**

**1.Case Entry :**

Information need to be entered once . you have to just fill up requirement information of your client case . When case is decided you can select the option of decided cases so that further will not be shown in your diary.

**2. Personal information:**

You have to make entries relating to your personal information diary.You can see this specific date or between any date and many more information as detailed below.

**3.Next Hearing Date:**

This entry can be made directly into diary or by using special feature which shows all cases where next hearing date not entered .

**4.Case Stage Entry :**

Facility to know progress stage of case. This entry made while entering hearing the date in the diary

**5**.**Court Wise Case**:

Diary will show your list of cases for specific court.

**6**.**Active Case**:

You have no option in the case details to make case active or decided.

**7.Individual Case Details**:

Facility to see all hearing dates and other details of an individual case provided.

MODEL USED IN THE PROJECT:

WATERFALL MODEL:

The **waterfall model** is a relatively linear sequential design approach for certain areas of engineering design. In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, deployment and maintenance.

The Waterfall Model originated in the manufacturing and construction industries; where the highly structured physical environments meant that design changes became prohibitively expensive much sooner in the development process. When first adopted for software development, there were no recognized alternatives for knowledge-based creative work.[[1]](https://en.wikipedia.org/wiki/Waterfall_model#cite_note-benington-1)

Requirement Analysis

System Design

Implementation

Testing

Deployment

Maintenance

Model

In Royce's original waterfall model, the following phases are followed in order :

Software requirement: captures product requirement document

* Analysis: resulting in modules, schema and business rules

resulting in the software architecture.

* Coding: the developing ,providing and integration of software
* Testing: the systematic discovery and debugging of defects
* Operations the, installation migration and support and maintenance of complete systems thus the waterfall model maintains that one should move to a phase only when its preceding phase is reviewed and verified.

Various modified waterfall model(including Royce's final model), however, can include slight or major variations on this process.[[3]](https://en.wikipedia.org/wiki/Waterfall_model#cite_note-royce-3) These variations included returning to the previous cycle after flaws were found downstream, or returning all the way to the design phase if downstream phases deemed insufficient.

Project Modules :

**1**. Admin:

a) Admin module maintains a clean and enjoyable working environment.

b) Handling internal and external communication or management systems.

c) Managing administrative staff.

2.client

a) clients have a major role to play in the production and systametic approach.

b) client responsibility is to provide all the facts that the lawyer requests.

c) it confirms the certain details of events which gives success of case

d) according to the advice given by lawyer client take decisions but client cannot leave decisions upto the lawyer

e) lawyer discuss the fees with the client if the client cannot pay the fees as requested by the lawyers then the lawyer has right to stop work on the file.

3. Court

The **Role** of the **Courts**. The justice system is the mechanism that upholds the rule of law. Our **courts** provide a forum to resolve disputes and to test and enforce laws in a fair and rational manner. ... The

circumstances of any particular case may result in a modification or refinement of the law and its application.

4. Advocate:

a) Collaboration.

b) communication.

c) management.

d) professionalism.

e) passion, preservance

PROCESS LOGIC :

Input process output

Admin view

Client view

Advocate profile

Court profile

Case report

Judgement report

Admin

Client details

Advocate details

Government

Case details

**SYSTEM ANALYSIS**

**Hardware Requirements :**

**Client side:**

|  |  |
| --- | --- |
| Operating System | Windows/Linux |
| Processor | Pentium III or 2.0 GHz or higher |
| RAM | 256 Or More |

**Server side:**

|  |  |
| --- | --- |
| **Operating System** | Windows/Linux |
| Processor | Pentium 4 and Above |
| RAM | 1 GB and Above |
| `Hard Disk | 40 GB and Above |

**Software Requirements :**

|  |  |
| --- | --- |
| Operating System | Windows/Linux |
| Front End | PHP |
| Back end | MYSQL |
| Server | WAMP 3.0 |

**SYSTEM DESIGN :**

**Data Flow Diagram:**

A DFD is a graphical system model that shows all of the main requirements for an information system in one diagram : Inputs and Outputs processes and data storage. A DFD describes what data flows rather than how it is processed. Everyone works on a development project can see all aspects of the system working together at once with the DFD. That is one reason for its popularity. The DFD is also easy to read because it is graphical model.

The DFD is mainly used during problem analysis end users management, and all information system workers typically can read ad interpret the DFD with minimal training**.**

**Data Flow Diagram Notations:**

**Data Flow:**

It may be from file-to-file or file-to-process or process- process. It is generally in terms of attributes. There may be either an input data flow or output data flow.

**Functional Processing:**

The process is nothing but the transformation of data it starts with the subject and it has the verb followed by the subject.

**Data store:**

It includes file, data base and repository. To parallel lines represent it or a one end closed rectangle

Or

**The data flow Diagram:**

Username and password

database

admin

Case reports

Case details

Client details

Judgement report

Court details

government

acknowledgement

**CONTEXT DIAGRAM**

advocate

Login

Admin

Add section,article view

Delete detaills Advocate

Update details

Approve advocate

**Use case Diagram**

A **use case diagram** in the Unified Modeling language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**UML Notations for Use Case**

**Actor:** An actor is a direct external user of a system-an object or a set of objects that communicates directly with the system but that is not part of the system.

**Use case:**

A use case is a coherent piece of functionality that a system can provide by interacting with actors.

**USE CASE DIAGRAM:**

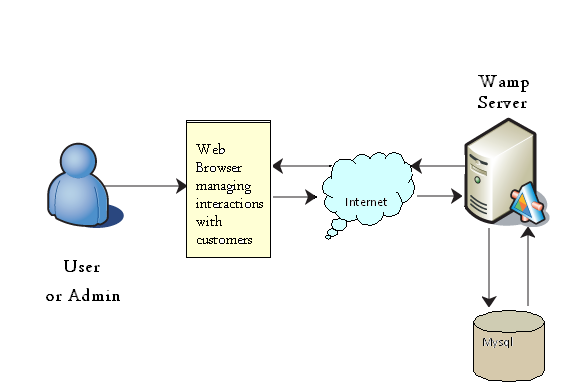
**Advocate**

**Admin**

**User**

**Architecture Diagram :**

The initial design process of identifying the sub-system and establishing a frame work for sub-system control and communication is called architectural design and the output of this design process is a description of the software architecture. The user interface, functional process logic, product data storage and data access are developed and maintained. This diagram is very important to understand the overall concept of system.



ER DIAGRAM:

DIAGRAM:

belong

**n 1 1 n**

Court

Advocate

Add

Admin

**n 1 1**

add

has

**1**

maintain

**1 n Screen Clipping**

User

case

**n n**

Dictionary

manage

**1 n**

view

Add

**n n 1 1**

Law acts

Sections

Articles

**Logical Design:**

The logical design describes the detailed specification for the system, describing its features, effective communication and the user interface requirements. The logical system of a proposed system should include the following.

1. External System Structure.
2. Relationship between all the activities.
3. The physical construction and all the activities.
4. Global data.
5. Control flow.
6. Derived program structure.

**Design Principles:**

Basic design principles that enable the software engineer to navigate the design process are:

* The design process should not suffer from “Tunnel vision”.
* The design should be traceable to analysis model.
* The design should not reinvent the wheel.
* The design should minimize the intellectual distance between the software and the problem, as it exists in the real world.
* The design should exhibit uniformity and integrity.
* The design should be structured to accommodate changes.
* The design not coding, the coding is not a design.
* The design should be assessed for the quality, as it is being created, not after the fast.
* The design should be reviewed to minimize the conceptual errors.

**Database table**

## *Table structure for table login*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constraint | Null |
| Username | Varchar(20) |  | Not Null |
| Password | Varchar(20) |  | Not Null |
| Type | Varchar(20) |  | Not Null |
| Hint question | Varchar(20) |  | Not Null |
| Hint answer | Varchar(20) |  | Not Null |
| Status | Varchar(20) |  | Not Null |

## *Table structure for table Court Details*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constraint | Null |
| Court id | int (20) | Primary key | Not Null |
| Court name | Varchar(20) |  | Not Null |
| Address | Varchar(20) |  | Not Null |
| City | Varchar(20) |  | Not Null |
| Contact number | int (20) |  | Not Null |

## Table structure for Articles

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constraint | Null |
| Article id | int (20) | Primary key | Not Null |
| Article Name | Varchar (20) |  | Not Null |
| Short Description | varchar(20) |  | Not Null |
| Complete description | varchar(20) |  | Not Null |

***Table structure for table Case Reports***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constraint | Null |
| Case report id | int (20) | Primary key | Not Null |
| report Name | Varchar (20) |  | Not Null |
| Report | varchar(20) |  | Not Null |
| Upload date | Date |  | Not Null |
| Client case id | Int(20) | Foreign key | Not Null |

***Table structure for table Client Document***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constraint | Null |
| Client doc id | int (20) | Primary key | Not Null |
| Client id | int (20) | Foreign key | Not Null |
| Doc name | varchar(20) |  | Not Null |
| Upload date | Date |  | Not Null |

***Table structure for table Dictionaries***

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constraint | Null |
| Dictionary id | int (20) | Primary key | Not Null |
| Dictionary word | varchar(20) |  | Not Null |
| description | varchar(20) |  | Not Null |

## *Table structure for table client details*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constraint | Null |
| client id | int (20) | Primary key | Not Null |
| Client name | Varchar(20) |  | Not Null |
| Client address | Varchar(20) |  | Not Null |
| City | Varchar(20) |  | Not Null |
| Contact number | Int(12) |  | Not Null |
| Email id | Varchar(20) |  | Not Null |
| Register date | Date |  | Not Null |

**SCREENSHOT:**

10)SOURCE CODE

LOGCHECK:

CONCLUSION

Software is said to have attained its objective only when it meet all requirements of the user, further the user himself is the person to judge the success of the system. Every attempt has been made to ensure that the system is fully functional & works effectively & efficiently. The system has been tested with simple data to cover all possible options & checked for al outputs. Since the system is flexible & modular, further modifications of this package can be easily incorporated.

Importance of the system:

* Less manual work.
* Increased efficiency.
* Decreases the rate of errors.
* It reduces the time consumption.
* Quick (instant) result.

BIBLIOGRAPHY:

***Books:***

1. “Software Engineering” by Ian Somerville, Sixth Edition, Pearson Education Ltd 2007.
2. “Web Programming”, by ‘Chris Bates’ Wiley Dream Tech India, 2nd Edition.
3. Database Management Systems by Navathe

***Websites:***

1. [***http://en.wikipedia.org/wiki/PHP***](http://en.wikipedia.org/wiki/PHP) *for Php.*
2. [***http://www.hotscripts.com/category/php/***](http://www.hotscripts.com/category/php/) ***for Php***
3. [***http://www.mysql.com/click.php?e=35050***](http://www.mysql.com/click.php?e=35050) ***for MySql***