VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 018

A Mini -Project Work on

"Court Case Management System"

A Dissertation work submitted in partial fulfillment of the requirement
for the award of the degree

Bachelor of Engineering
In
Information Science & Engineering

Submitted by

Akshay Kumar S **1AY18IS010**
Dhyan Rajesh **1AY18IS039**

Under the guidance of
Prof. Surekha K.B
Associate Professor

ACHARYA

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ACHARYA INSTITUTE OF TECHNOLOGY

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI.APPROVED BY AICTE, NEW DELHI, ACCREDITED BY NAAC, NEW DELHI)

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

2020-21

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
ACHARYA INSTITUTE OF TECHNOLOGY
(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI.APPROVED BY AICTE, NEW DELHI, ACCREDITED BY NAAC, NEW DELHI)
Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

Certificate

This is to certify that the Mini-Project work entitled **"Court Case Management System"** is a bonafide work carried out by **Akshay Kumar S (1AY18IS010)** and **Dhyan Rajesh (1AY18IS039)** in partial fulfillment for the award of the degree of **Bachelor of Engineering** in **Information Science and Engineering** of the **Vishvesvaraya Technological University**, Belagavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Engineering Degree.

Prof. Surekha K.B Guide	**Prof. Marigowda C K** HOD

Name of the Examiners	**Signature with date**
1. _____	_____
2. _____	_____

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this mini-project would be incomplete without the mention of the people who made it possible through constant guidance and encouragement.

We would take this opportunity to express our heart-felt gratitude to **Sri. B. Premnath Reddy**, Chairman, Acharya Institutes and **Dr. Prakash M R**, Principal, Acharya Institute of Technology for providing the necessary infrastructure to complete this mini-project.

We wish to express our deepest gratitude and thanks to **Prof. Marigowda C K**, Head of the Department, Information Science and Engineering.

We wish to express sincere thanks to my guide **Prof. Surekha K B**, Associate Professor, Department of Information Science and Engineering for helping us throughout and guiding us from time to time.

A warm thanks to all the faculty of Department of Information Science and Engineering, who have helped us with their views and encouraging ideas.

ABSTRACT

The purpose of the Court Case Management System is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirement .The required hardware and software are easily available and easy to work with. Court Case Management System, as described above, can lead to error free, secure, reliable and fast management system.

It can assist the advocate to concentrate on their activities rather than to concentrate on record keeping. Thus it will help the advocate and client in better utilization of resources. The advocate can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system. This project assists the advocate on the usage of databases required by the client. It can be used to make and hold databases of dates of hearing of their cases in the Court that are being updated by the advocate and make them available to the client.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

In Today's world, it is difficult to manage Cases for an Advocate using manual system and also Client doesn't have any idea about his case hearing dates in the court. To know the case proceedings and hearing dates, every time he has to go to the Court or Contact his Lawyer to Know about his Case. To eliminate the above difficulties, features like 'search for your case using case id' must be developed and Advocate need to maintain about Client and his respective Case details in the Database, which will help the Client to know about his case by using this system.

The project, "COURT CASE MANAGEMENT SYSTEM" is also a step towards offering more or less the similar features. This system enables to manage and record the activities of all Case proceedings handled by the advocate.

Court Case Management System enables the Client to search for case proceedings and hearing dates in a more systematic and efficient manner, hence improving the managing of the Case best efficient way.

1.1 Introduction to DBMS

DBMS stands for **D**ata**b**ase **M**anagement **S**ystem. We can break it like this DBMS = Database + Management System. Database is a collection of data and Management System is a set of programs to store and retrieve those data. Basically DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database. A datum is a unit of data. Meaningful data combined to form information. Hence, information is interpreted data – data provided with semantics. MS. ACCESS is one of the most common examples of database management software.

Database systems are meant to handle large collection of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

1.1.1 Why DBMS?

- To develop software applications in less time.
- Data Independence and efficient use of data.
- For uniform data administration.
- For data integrity and security.
- For concurrent access of data, and data recovery from crashes.
- To use user-friendly declarative query language.

1.1.2 Database applications

- **Telecom:** There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.

- **Industry:** Where it is a manufacturing unit, warehouse or distribution center, each one needs a database to keep the records of ins and outs. For example distribution center should keep a track of the product units that supplied into the center as well as the products that got delivered out from the distribution center on each day; this is where DBMS comes into picture.

- **Education sector:** Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.

- **Online shopping:** You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and

preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

- **Banking system:** For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.

1.1.3 Advantages of DBMS

A DBMS manage data and has many advantages.

- **Data Independence:** Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.

- **Efficient data access:** DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.

- **Data integrity and security:** If data is accessed through the DBMS, the DBMS can enforce integrity constraints on the data.

- **Data administration:** When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

- **Providing backup and recovery:** A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery.

- **Permitting inferencing and actions using rules:** Some database systems provide capabilities for defining deduction rules for inferencing new information from the stored database facts.

1.1.4 Components of DBMS

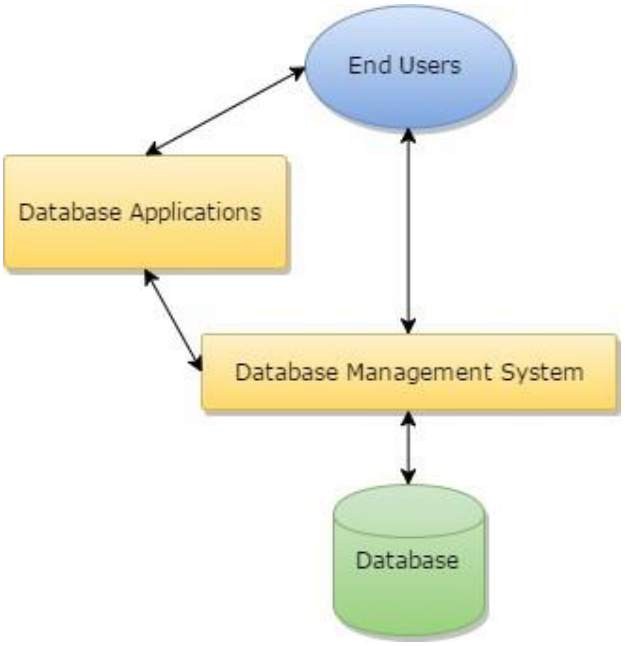

Fig-1.1: Components of a Database Management System

- **Users:** Users may be of any kind such as DB administrator, System developer or database users.

- **Database application:** Database application may be Departmental, Personal, organization's and / or Internal.

- **DBMS:** Software that allow users to create and manipulate database access.

- **Database:** Collection of logical data as a single unit.

- **Database access language:** This is used to access the data to and from the database, to enter new data, update existing data, or retrieve required data from databases. The user writes a set of appropriate commands in a database access language, submits these to the DBMS, which then processes the data and generates and displays a set of results into a user readable form.

1.1.5 Three-Schema architecture

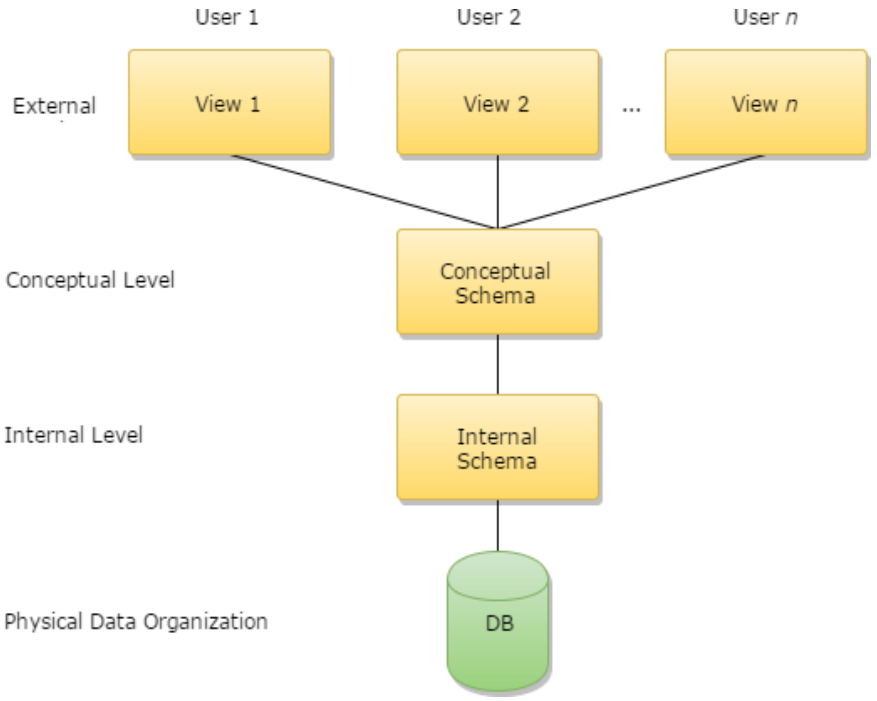

Fig-1.2: Architecture of database system

The levels form a three-level architecture that includes an external, a conceptual, and an internal level. The way users recognize the data is called the external level. The way the DBMS and the operating system distinguish the data is the internal level, where the data is actually stored using the data structures and file. The conceptual level offers both the mapping and the desired independence between the external and internal levels.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Hardware Requirements

- **Processor:** Intel Core2 Quad @ 2.4 GHz on Windows® Vista 64-Bit / Windows® 7 64-Bit / Windows® 8 64-Bit / Windows® 8.1 64-Bit.
- **RAM:** 2GB of RAM
- **Memory:** 256GB Hard drive

2.2 Software Requirements

- **Operating system:** Windows® Vista 64-Bit / Windows® 7 64-Bit / Windows® 8 64-Bit / Windows® 8.1 64-Bit.
- **Front end:** HTML CSS JavaScript
- **Back end:** MySQL
- **Software:** PHP
- **IDE:** XAMPP

CHAPTER 3

DESIGN

3.1 ER Diagram

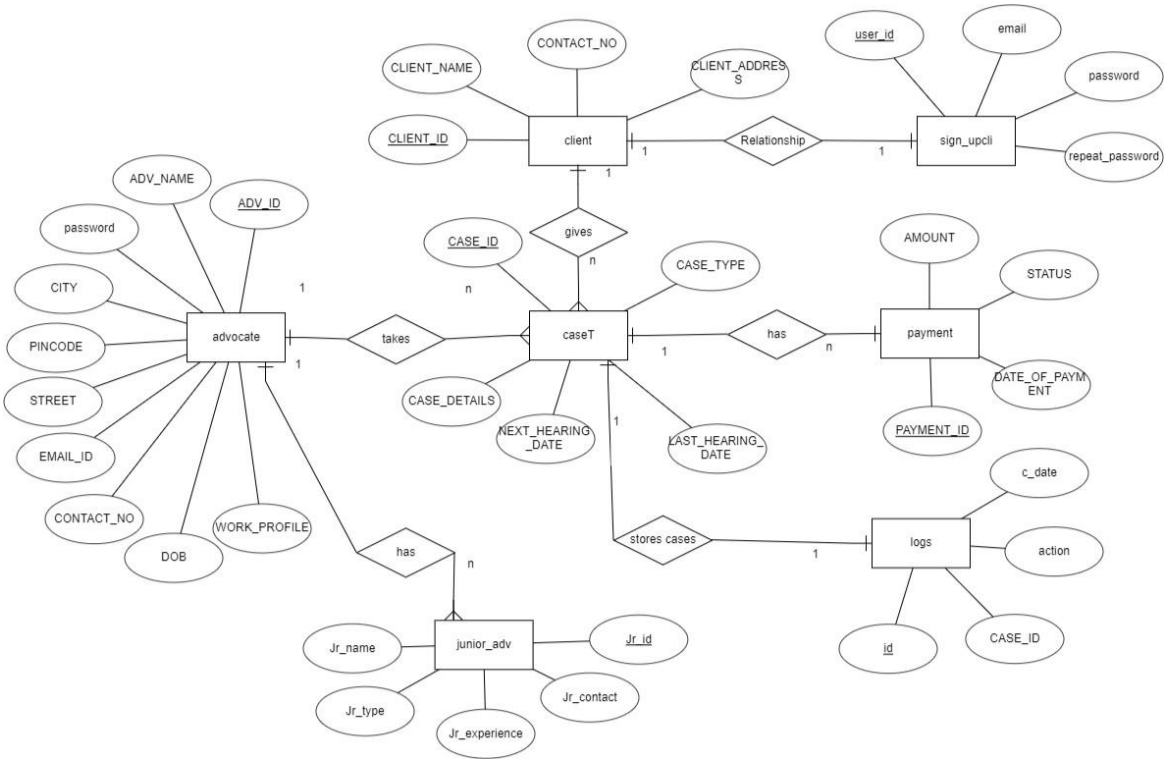

Fig-3.1: Entity Relationship Diagram

1: N

- One advocate can handle many cases.
- One client can have many cases.
- One user can deal with many junior advocates.
- One case can have many payments.

1:1

- One case gets triggers once in logs table.
- One client will have only one login credentials.

3.2 Schema Diagram

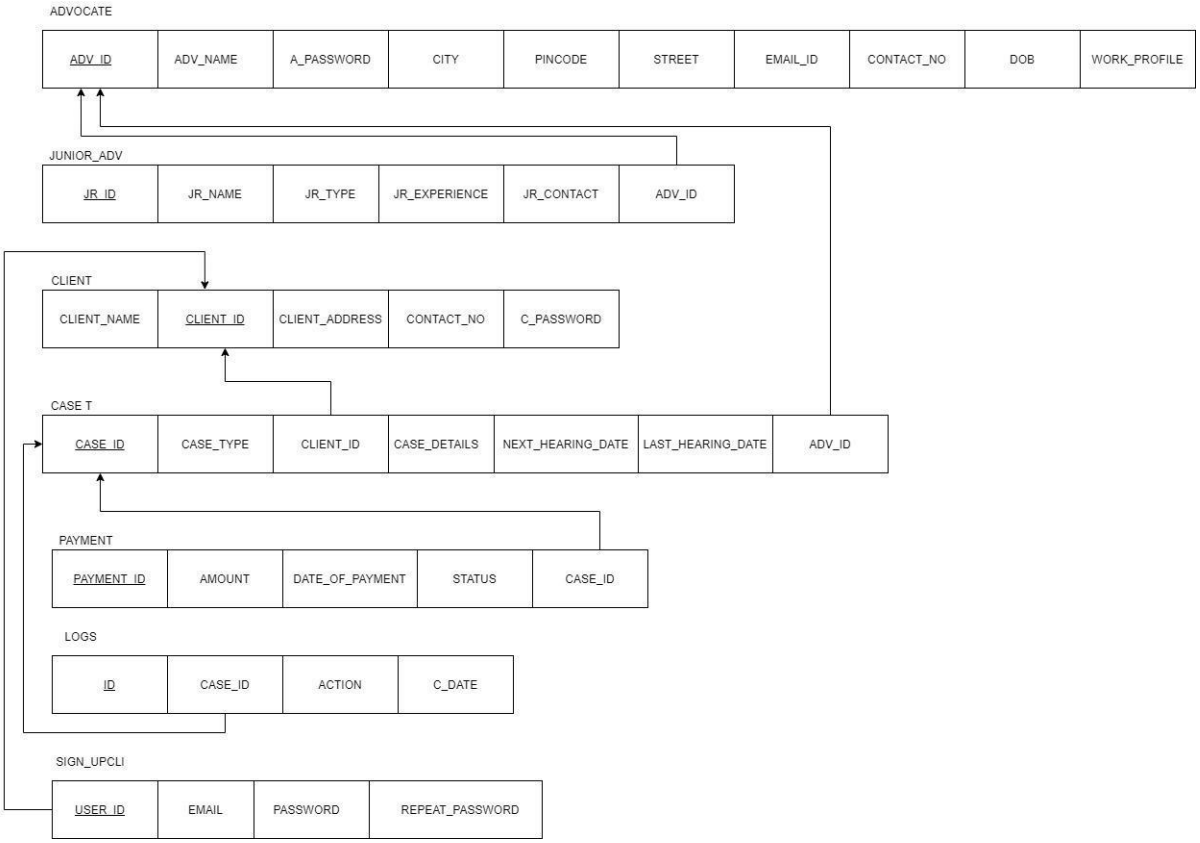

Fig-3.2: Schema Diagram

Schema Diagram: An illustrative display of (most aspects of) a database schema.

Schema Construct: A component of the schema or an object within the schema, e.g., STUDENT, COURSE.

CHAPTER 4

IMPLEMENTATION

4.1 Tables

4.1.1 Advocate

SNO	COLUMN_NAME	DATA_TYPE	DESCRIPTION
1	ADV_ID	Varchar	Primary key
2	ADV_NAME	Varchar	
3	EMAIL_ID	Varchar	
4	CONTACT_NO	Varchar	
5	DOB	Date	
6	WORK_PROFILE	Integer	
7	STREET	Varchar	
8	PINCODE	Integer	
9	CITY	Varchar	
10	PASSWORD	Varchar	

```
CREATE TABLE `advocate` (
  `ADV_ID` varchar (20) NOT NULL,
  `ADV_NAME` varchar (20) DEFAULT NULL,
  `EMAIL_ID` varchar (20) DEFAULT NULL,
  `CONTACT_NO` varchar (20) DEFAULT NULL,
  `DOB` date DEFAULT NULL,
  `WORK_PROFILE` int (3) DEFAULT NULL,
  `STREET` varchar (20) DEFAULT NULL,
  `PINCODE` int (10) DEFAULT NULL,
  `CITY` varchar (30) DEFAULT NULL,
  `Password` varchar (10) DEFAULT NULL
)
```

4.1.2 Case

SNO	COLUMN_NAME	DATA_TYPE	DESCRIPTION
1	CASE_ID	Varchar	Primary key
2	CASE_TYPE	Varchar	
3	CASE_DETAILS	Varchar	
4	LAST_HEARING_DATE	Date	
5	NEXT_HEARING_DATE	Date	
6	CLIENT_ID	Varchar	

```
CREATE TABLE `case_` (
  `CASE_ID` varchar (10) NOT NULL,
  `CASE_TYPE` varchar (20) DEFAULT NULL,
  `CASE_DETAILS` varchar (20) DEFAULT NULL,
  `LAST_HEARING_DATE` date DEFAULT NULL,
  `NEXT_HEARING_DATE` date DEFAULT NULL,
  `CLIENT_ID` varchar (10) DEFAULT NULL
)
```

4.1.3 CLIENT

SNO	COLUMN_NAME	DATA_TYPE	DESCRIPTION
1	CLIENT_ID	Varchar	Primary key
2	CLIENT_NAME	Varchar	
3	CLIENT_ADDRESS	Varchar	
4	CONTACT_NO	Integer	
5	CASE_ID	Varchar	

```
CREATE TABLE `client` (
        CLIENT_ID` varchar (10) NOT NULL,
        CLIENT_NAME` varchar (20) DEFAULT NULL,
       `CLIENT_ADDRESS` varchar (20) DEFAULT NULL,
       `CONTACT_NO` int (10) DEFAULT NULL,
       `CASE_ID` varchar (10) DEFAULT NULL
        )
```

4.1.4 Junior Advocate

SNO	COLUMN_NAME	DATA_TYPE	DESCRIPTION
1	Jr_name	Varchar	Primary key
2	Jr_type	Varchar	
3	Jr_contact	Varchar	
4	Jr_experience	Varchar	

```
CREATE TABLE `junior_adv` (
  `Jr_name` varchar (20) DEFAULT NULL,
  `Jr_type` varchar (20) DEFAULT NULL,
  `Jr_contact` varchar (20) DEFAULT NULL,
  `Jr_experience` varchar (20) DEFAULT NULL)
```

4.1.5 Logs

SNO	COLUMN_NAME	DATA_TYPE	DESCRIPTION
1	ID	Varchar	Primary key
2	CASE_ID	Varchar	
3	ACTION	Varchar	
4	C_DATE	Datetime	

```
CREATE TABLE `logs` (
  `id` int (11) NOT NULL,
  `case_id` varchar (10) NOT NULL,
  `action` varchar (20) NOT NULL,
  `c_date` datetime NOT NULL
)
```

4.1.6 Payment

SNO	COLUMN_NAME	DATA_TYPE	DESCRIPTION
1	PAYMENT_ID	Varchar	Primary key
2	CASE_ID	Varchar	
3	DATE_OF_PAYMENT	Date	
4	CLIENT_ID	Varchar	
5	STATUS	Varchar	
6	AMOUNT	Integer	

```
CREATE TABLE `payment` (
  `PAYMENT_ID` varchar (10) NOT NULL,
  `CASE_ID` varchar (10) DEFAULT NULL,
  `DATE_OF_PAYMENT` date DEFAULT NULL,
  `CLIENT_ID` varchar (10) DEFAULT NULL,
  `STATUS` varchar (10) DEFAULT NULL,
  `Amount` int (7) DEFAULT NULL
)
```

4.1.7 Sign-up Client

SNO	COLUMN_NAME	DATA_TYPE	DESCRIPTION
1	USER_ID	Varchar	Primary key
2	EMAIL	Varchar	
3	PASSWORD	Varchar	
4	REPEAT PASSWORD	Varchar	

```
CREATE TABLE `sign_upcli` (
  `user_id` varchar (20) NOT NULL,
  `email` varchar (20) DEFAULT NULL,
  `password123` varchar (20) DEFAULT NULL,
  `repeat_password` varchar (20) DEFAULT NULL
)
```

4.2 Triggers

```
CREATE TRIGGER `case_added`
AFTER INSERT ON `case_`
FOR EACH ROW INSERT INTO logs VALUES (null, new.case_id, "Added", now ( ))
```

4.3 Stored Procedures

```
DELIMITER $$
CREATE DEFINER=`root `@`localhost` PROCEDURE `mycases` () SELECT * FROM
case_ ORDER BY CASE_ID DESC$$
DELIMITER;
```

CHAPTER 5

SNAPSHOTS

The following snapshot contains the home window of Court Case Management System

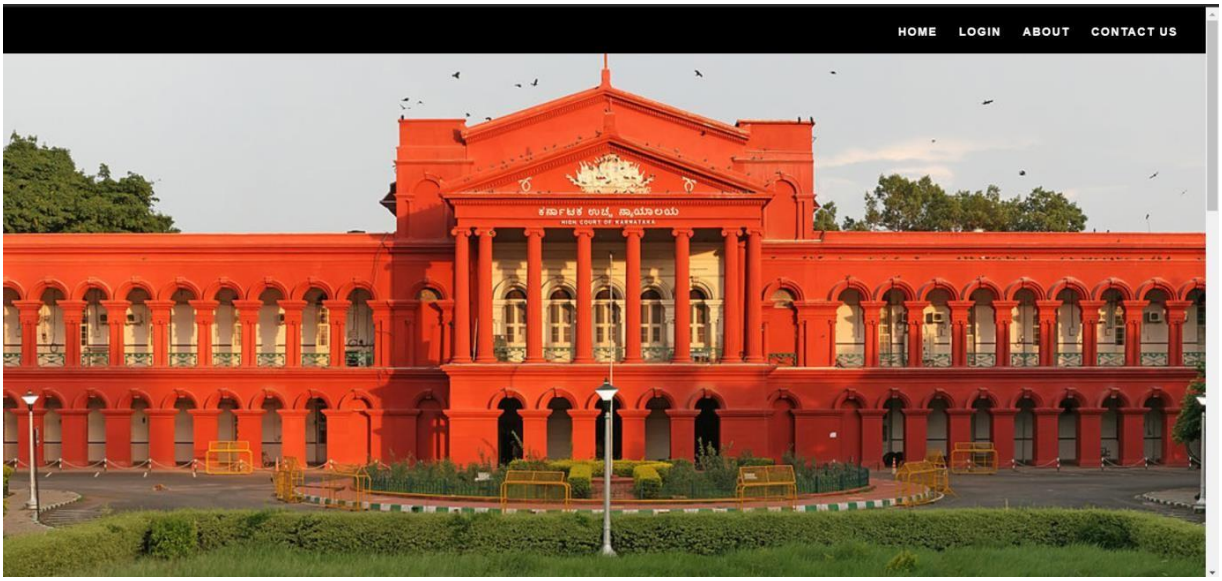

Fig-5.1: Snapshot of login window

The following snapshot contains the Login page of both Client and Advocate.

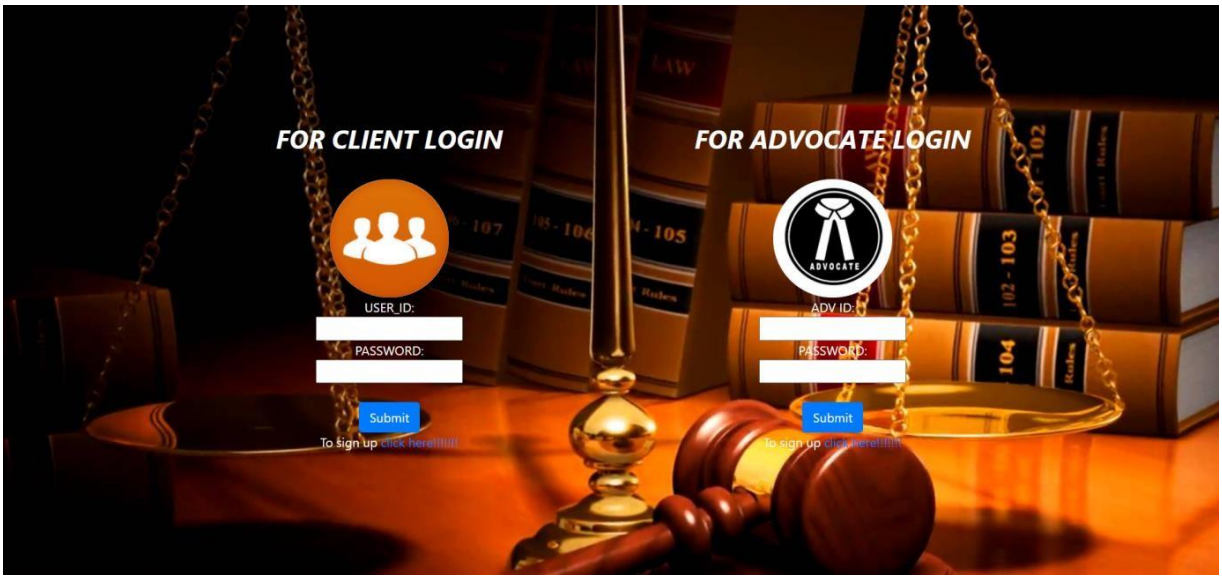

Fig-5.2: Snapshot of welcome screen

The following snapshot contains the registration form of an advocate, where he sign ups by entering his details.

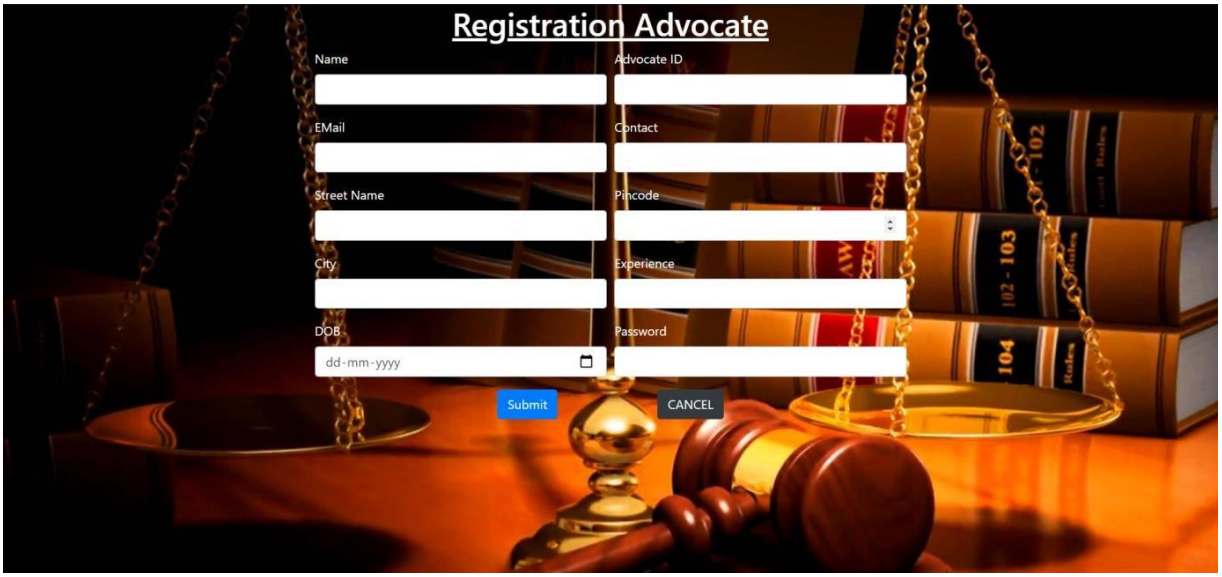

Fig-5.3: Snapshot of Advocate's registration page

The following snapshot contains home page of the Advocate.

Fig-5.4: Snapshot of home page of Advocate

The following snapshot contains the details of Cases. It also has the tabs for adding a case, updating the details of it such as last and first hearing dates. It also has the delete tab which can be used to delete the records of a particular client.

Fig-5.5: Snapshot of the Details of Cases

The following snapshot is appeared when the Advocate clicks on update button to update the case.

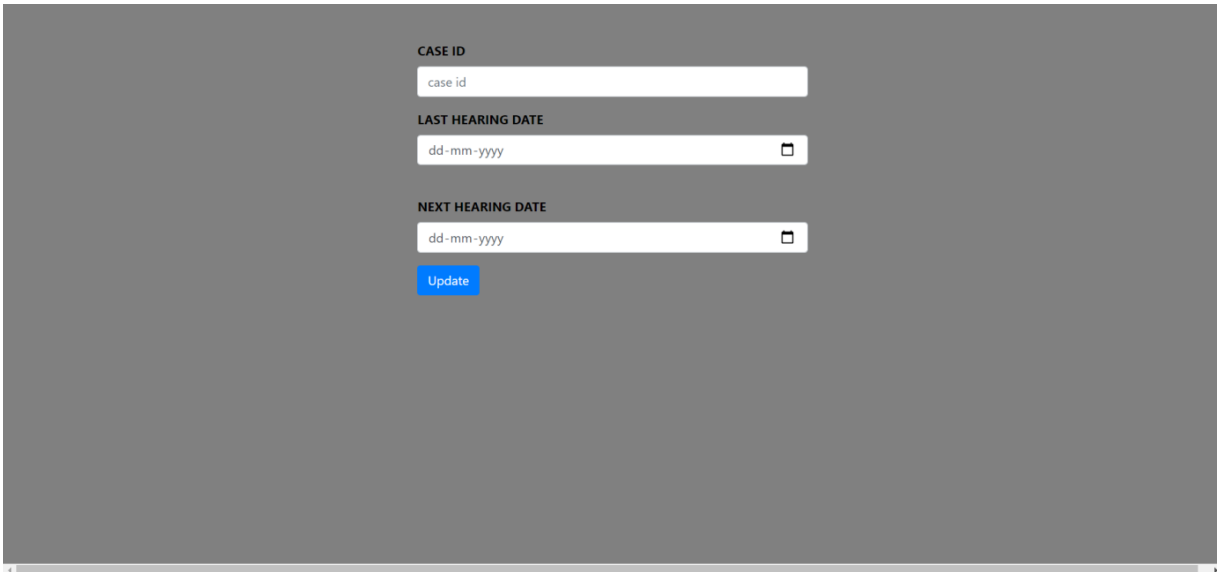

Fig-5.6: Snapshot of Update case window

The following snapshot contains the registration form of the client.

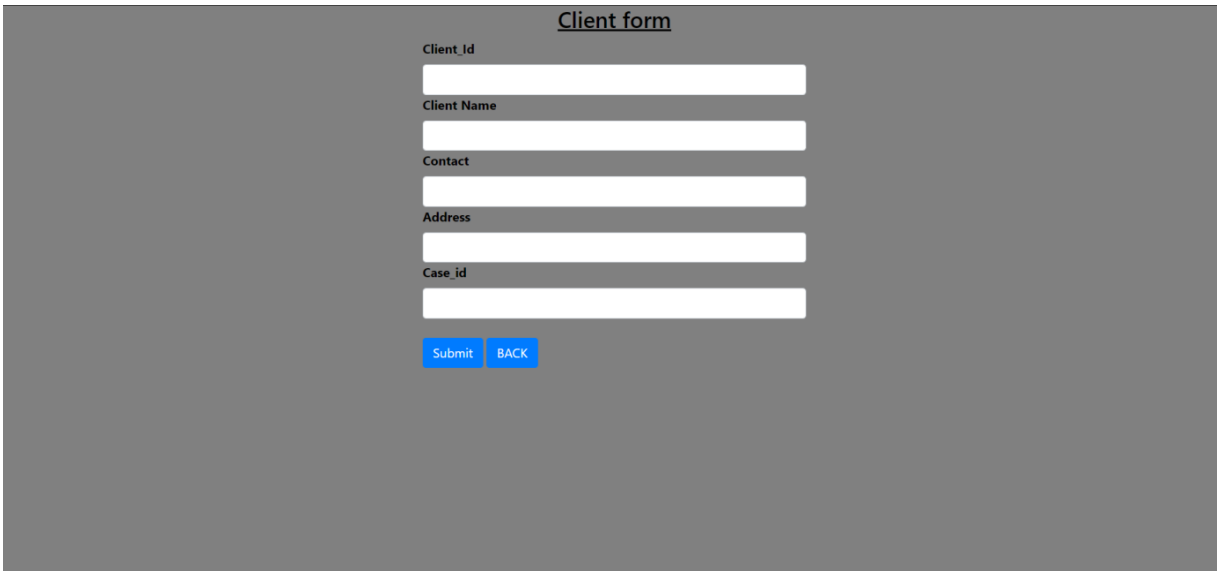

Fig-5.7: Snapshot of all Client Registration form

The following snapshot contains the details of the Clients .It also has the tabs for adding a client and the delete tab which can be used to delete the details of a particular client.

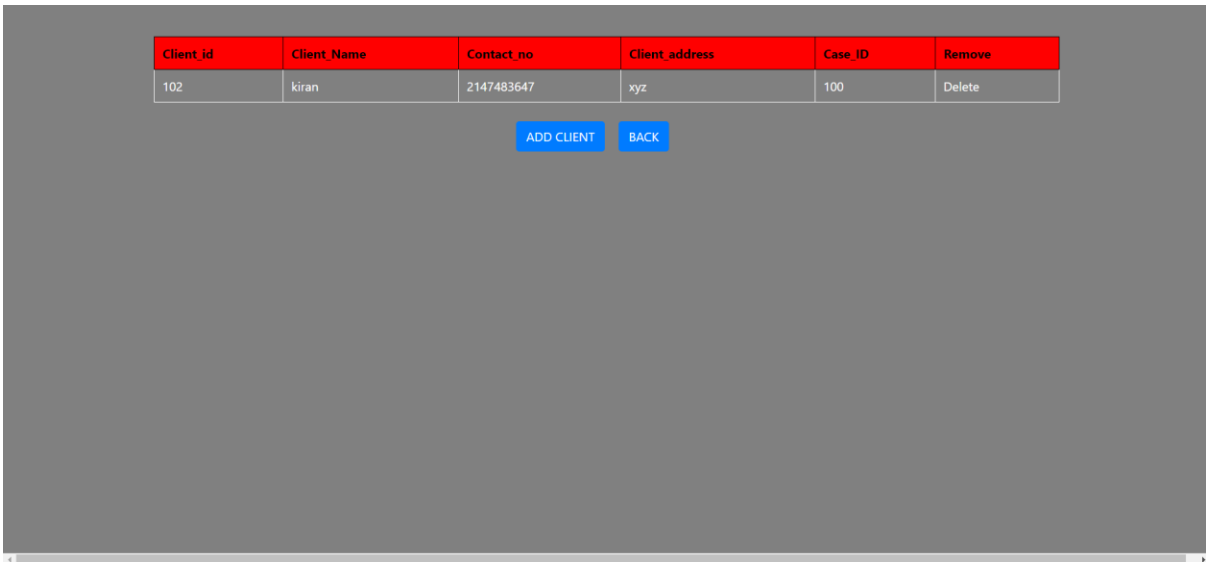

Fig-5.8: Snapshot of Client details.

The following snapshot contains the registration form of junior advocates.

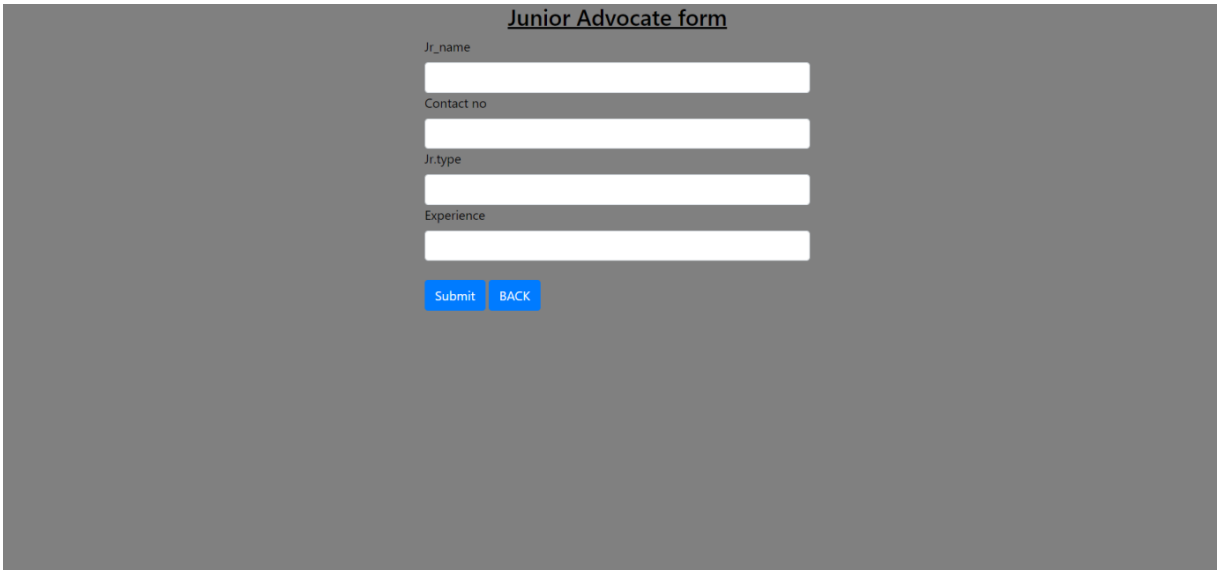

Fig-5.9: Snapshot of registration form of Junior Advocates.

The following snapshot contains the details of the Clients.

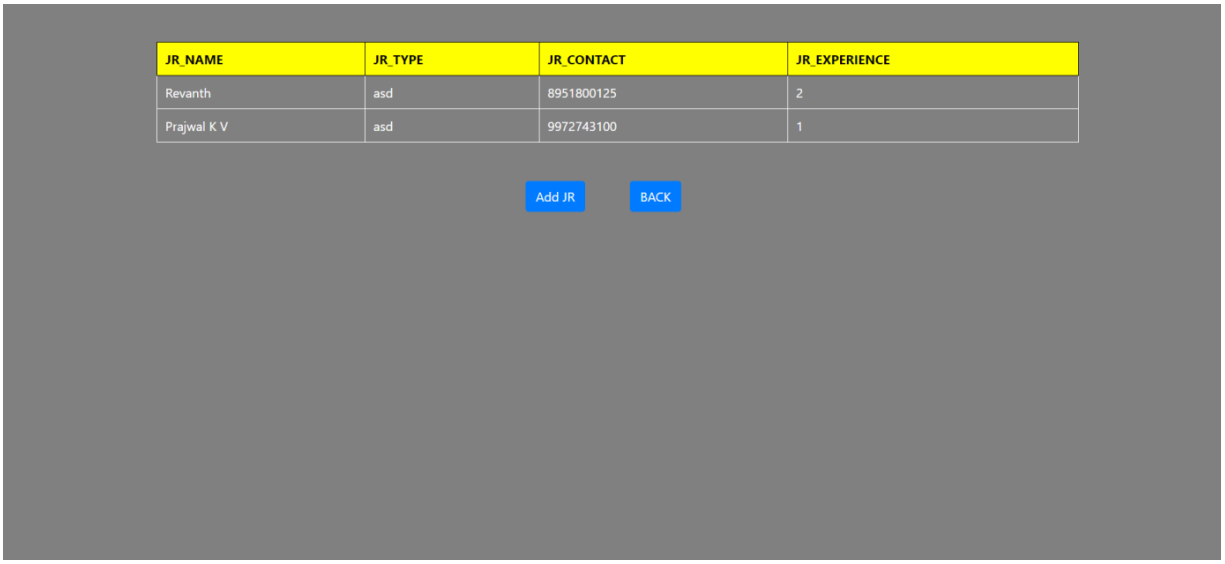

Fig-5.10: Snapshot of the details of junior advocates.

The following snapshot contains the registration form of the payment

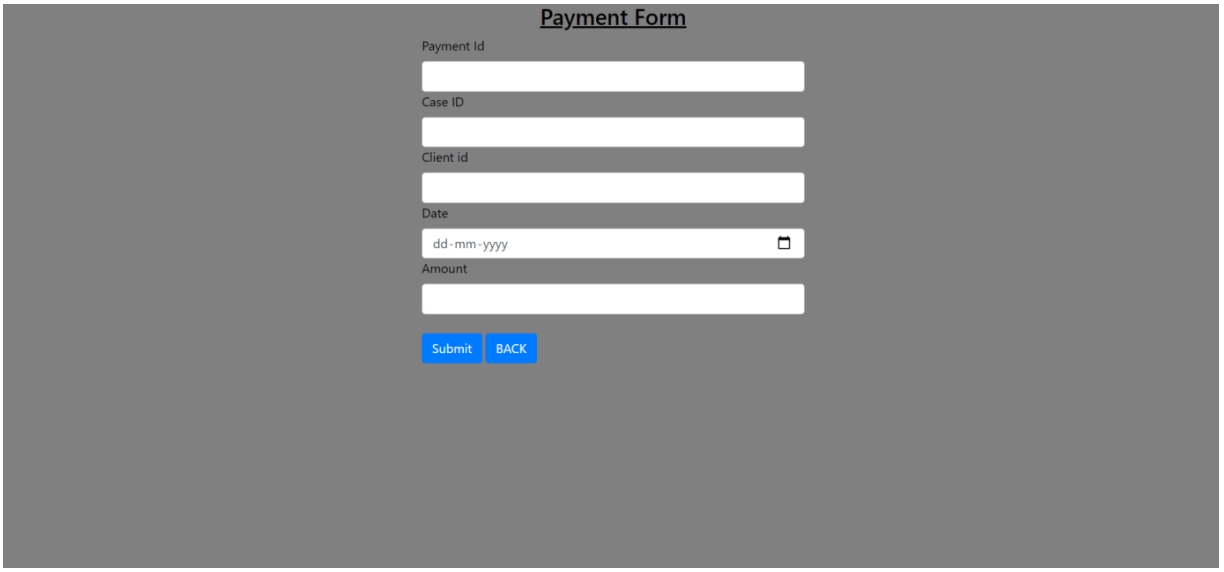

Fig-5.11: Snapshot of the registration form of payment.

The following snapshot contains the details of payments paid by the client.

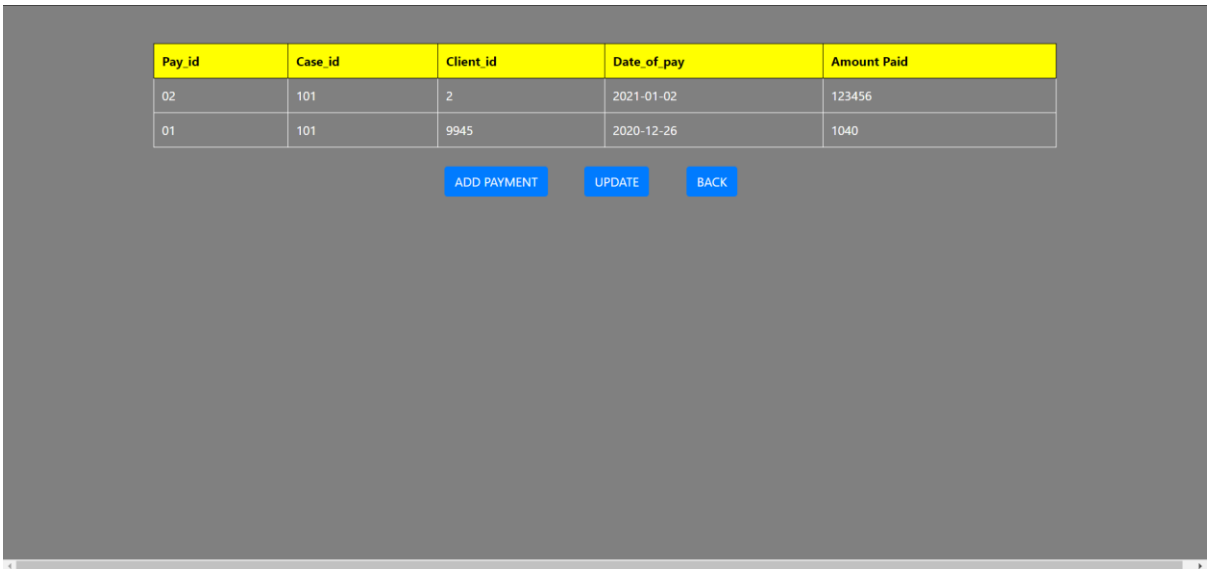

Fig-5.12: Snapshot of the details of the payment.

The following snapshot contains the sign-up page of Client.

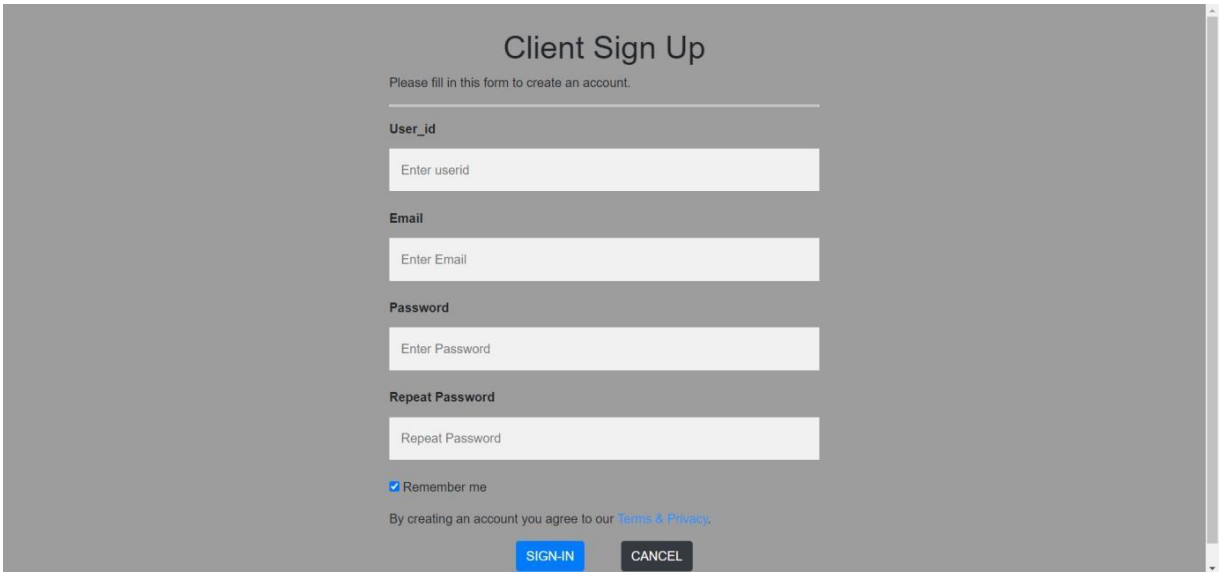

Fig-5.13: Snapshot of the Signup page of Client.

The following snapshot contains the window where the Client enters the case id to look for the hearing dates of his case in the court.

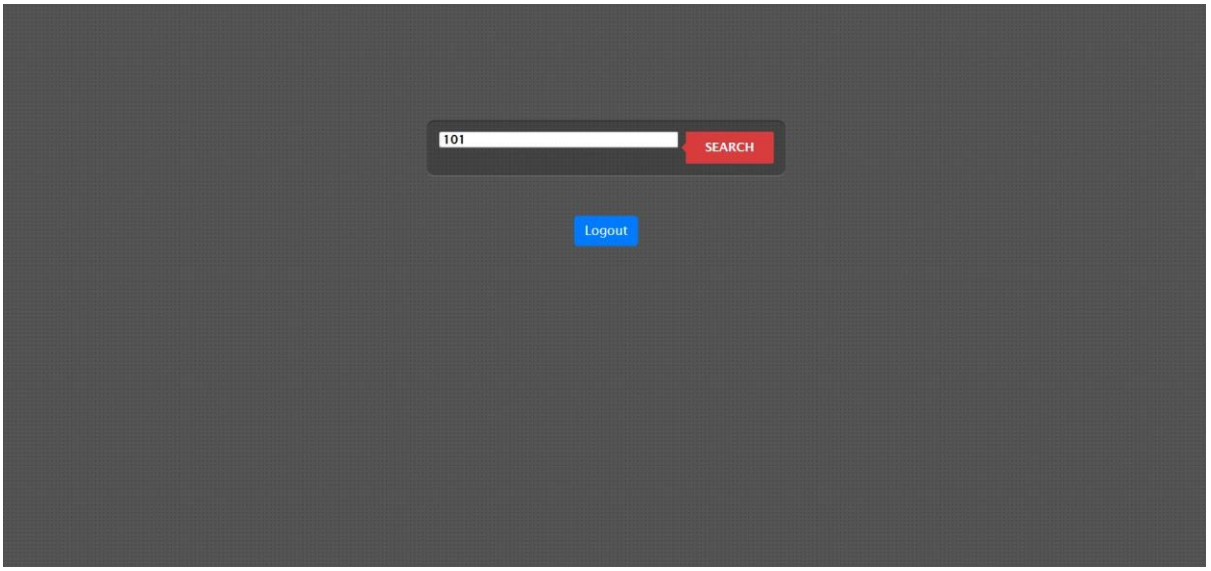

Fig-5.14: Snapshot of the search window.

The following snapshot contains the result window of hearing dates of cases of the client.

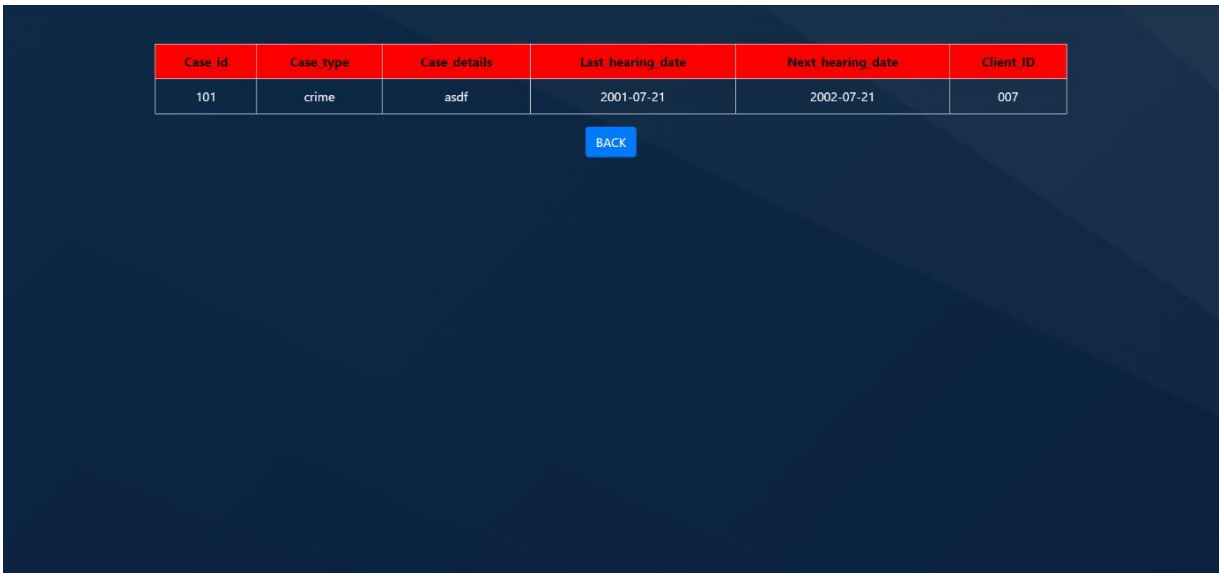

Fig-5.15: Snapshot of the result window.

CONCLUSION & FUTURE ENHANCEMENT

Conclusion

Court Case Management System is only a humble venture to satisfy the needs to manage the work of an Advocate. Several user-friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the Advocate. The objective of this project is to provide the status of the case i.e., respective hearing dates of his Case in the Court that enables the Client to know about his Case Proceedings in the Court and The Advocate needs to update the Hearing dates as the Case progresses in the court . This application can be used by both Advocate and Client to overcome and managing the information of the Case.

Future Enhancement

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- The Client can search for appropriate Advocate before giving his Case to Advocate.
- More advance software can be given for Court Case Management System including more facilities.
- This platform can be hosted on the online servers to make it accessible worldwide.
- Notifications can be received from Client to Advocate after searching for an appropriate an Advocate.
- Implement backup mechanism for taking backup of database on regular basis on different servers.

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here Client can only view the Case hearing dates updated by his Advocate. Enhancements can be done to view availability of his Advocate, Submission of required documents can be done online and also payment can be done online.
I have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.

BIBLIOGRAPHY

Web References

1. https://www.w3schools.in/dbms/intro/

   Introduction to DBMS

2. https://www.w3schools.com/sql/sql_intro.asp

   Introduction to SQL

3. https://dev.mysql.com/doc/

   MySQL documentation

4. https://dev.mysql.com/doc/

   PHP documentation

5.https://dev.mysql.com/doc/refman/8.0/en/stored-routines.html

  Stored Procedures documentation

6. https://stackoverflow.com/

   Stack overflow

7. https://www.coursera.org/learn/database-applications-php

   Database applications in php

8. https://www.coursera.org/learn/web-applications-php

   Web applications in php

Book References

1. Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson