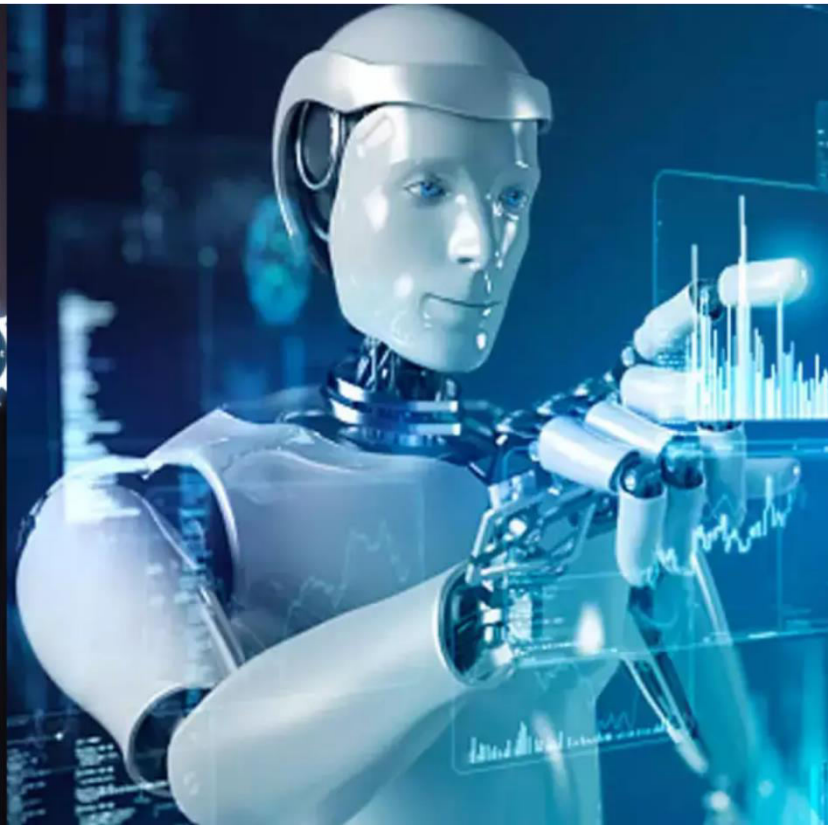
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 5, May 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405303|

Court Management System: A Web-Based Approach using Python and Django

Vismitha V, Afrin Fathima.R, Abhidev .P.S, Rahul.B, A.Renuka

Department of Computer Science, Nethaji Memorial Arts and Science College, Nemmara, University of Calicut, Kerala, India

Head of the Department of Computer Science, Nethaji Memorial Arts and Science College, Nemmara, University of Calicut, Kerala, India

ABSTRACT: -Access to justice has become a critical issue injustice systems worldwide, with technology increasingly viewed as a potential facilitator. This paper presents the development of a Court Management System (CMIS) - a web-based application designed to streamline judicial processes using Python and the Django framework. The system addresses inefficiencies in traditional court management by offering an automated, centralized platform for handling cases, scheduling, and communication between stakeholders. Role- based functionalities are implemented for judges, lawyers, court staff, and clients, enabling them to perform their respective tasks efficiently. The study follows the Agile Development Methodology, allowing direct interaction with users throughout the development process to ensure the system meets their requirements. Through automation of routine tasks such as case assignments, courtroom scheduling, and client-lawyer communications, the system reduces errors, saves time, and enhances productivity. Key modules include Client Module, Judge Module, Staff Module, and Lawyer Module, each designed with specific functionalities to facilitate respective user roles. The Court Management System provides an efficient solution for managing interactions between various stakeholders in the judicial system, ensuring transparency, accuracy, and ease of access, ultimately leading to a more organized and reliable judicial process.

KEY WORDS: Court Management, Django, Python, Judiciary System, Web Application, Case Management

I. INTRODUCTION

The Court Management System is a web-based application developed using Python and the Django framework to modernize and streamline court operations. It addresses the inefficiencies of traditional, manual systems by offering an automated and centralized platform for managing judicial processes. This system is designed to ensure transparency, accessibility, and efficiency in handling cases, scheduling, and communication between various stakeholders.

The platform supports role-based functionalities, catering to the needs of judges, lawyers, court staff, and administrators. Judges can manage case histories, monitor schedules, and oversee staff interactions. Lawyers are equipped to handle case bookings, submit cases to court, and receive real-time updates. Staff members can update case statuses, allocate courtrooms, and process payments, while clients can track their cases and communicate with their lawyers.

By automating routine tasks such as case assignments, courtroom scheduling, and client- lawyer communications, the system reduces errors, saves time, and enhances productivity. The Court Management System provides an intuitive interface and secure environment, ensuring a seamless experience for all users. The system offers a comprehensive solution to the challenges faced in court management, paving the way for a more organized, reliable, and efficient judicial system.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405303|

II. AIM

The aim of the Court Management System is to design and develop a centralized, web-based platform using Python and the Django framework to streamline judicial processes. This system seeks to automate and enhance the efficiency of court operations by providing role- specific functionalities for judges, lawyers, staff, and clients. The primary objective is to replace traditional, manual methods of court management with a modern, secure, and user-friendly solution that facilitates case handling, scheduling, courtroom allocation, and communication. By reducing errors, improving transparency, and ensuring accessibility, the system aims to create a more efficient and organized judicial environment.

III. EXISTING SYSTEM

The current court management systems suffer from several inefficiencies and limitations that hinder the effective functioning of judicial processes. These systems are largely manual or rely on outdated technologies, resulting in time-consuming, error- prone, and resource-intensive operations. Key challenges include:

1. **Paper-based case handling**: Leading to document loss, mismanagement, and delays in accessing case files.
2. **Lack of centralized digital storage**: Complicating record retrieval and hindering effective case tracking.
3. **Inefficient communication**: Between stakeholders, with important notifications and updates often being delayed or overlooked.
4. **Scheduling conflicts**: Manual courtroom allocation and hearing schedules frequently lead to overlaps and inefficiencies.
5. **Limited accessibility**: Requiring physical visits to the court to access information, submit documents, or make payments.
6. **Outdated payment processes**: Managed manually, involving long queues, delays, and errors in payment tracking.
7. **Lack of transparency**: Clients have little visibility into their case progress.
8. **No real-time updates**: Stakeholders remain unaware of critical developments or changes in case schedules or judgments.

These issues highlight the pressing need for a robust and automated Court Management System that can provide centralized case management, seamless communication, real-time updates, digital payment options, and enhanced transparency for all stakeholders involved.

IV. PROPOSED SYSTEM

The Court Management System is a web-based application built using Python and Django, aiming to modernize and optimize court operations. This system centralizes case management, scheduling, and communication, replacing outdated manual processes. It supports various roles, including judges, lawyers, court staff, and clients, each with specific functionalities to streamline their tasks.

Clients can track case progress in real-time, securely communicate with their lawyers, upload necessary documents, make payments, and receive automated notifications for hearings and deadlines. Judges can easily manage their caseload, schedule hearings, record judgments, and communicate with relevant stakeholders. Court staff can register cases, allocate courtrooms, update case statuses, process payments, and manage notifications for all parties. Lawyers can manage clients, submit legal documents, track case updates, and communicate with judges and staff.

By automating routine tasks and providing real-time updates, the system enhances efficiency, accuracy, and transparency, ensuring an organized and secure court environment. This system ultimately aims to reduce errors, save time, and improve productivity in judicial operations.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405303|

V. RESEARCH METHODOLOGY

The methodology used for the project development is the Agile Development Methodology. This methodology was chosen because the project needs to deal directly with clients and users, allowing developers to understand their requirements and incorporate feedback after each iteration. The agile approach enables flexibility andadaptation throughout the development process, ensuring that the final system meets user needs effectively.

VI. SYSTEM CONFIGURATION

Hardware Requirements:

- Processor: i3 or higher
- RAM: 4 GB minimum
- Hard Disk: 100 GB minimum
- Internet Connection
- Standard Input/Output Devices

Software Requirements:

- Language: Python
- IDE: VS Code
- Front End: Django
- Backend (Database): SQLite
- Other Technologies: HTML, CSS, JavaScript

VII. SYSTEM ANALYSIS AND DESIGN

1.1 Module Description

The proposed system has four modules:

1. Client Module

- Case Progress Tracking: Real-time monitoring of case status, updates on hearings, judgments, and case progress milestones.
- Secure Communication: Direct communication with assigned lawyers through a messaging system.
- Document Management: Upload and view
- case-related documents and evidence.
- Payment Management: Secure online payments for court fees with detailed payment history.
- Notification System: Automated alerts for upcoming hearings, deadlines, and case updates.
- Access to Legal Resources: Guidance on legalprocedures and general information about the judicial process.
- Case Submission: Initiate new cases by submitting details and supporting documents.
- User Profile Management: Update personal information and preferences.

2. Judge Module

- Case Assignment Overview: Dashboard of assigned cases, sorted by priority and status.
- Case Details Management: View evidence, statements, and documentation related to each case.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405303|

- Courtroom Scheduling: Schedule hearings and allocate courtroom slots.
- Judgment Recording: Input case rulings, judgments, and final case closures.
- Communication Tools: Communicate with lawyers and staff regarding case updates.
- Staff and Lawyer Supervision: Monitor activities such as case updates and lawyer submissions.

3. Staff Module

- Case Registration: Record new cases, including parties involved, case types, and documentation.
- Case Updates: Update case statuses and provide real-time information.
- Courtroom Allocation: Assign appropriate courtrooms based on availability.
- Payment Processing: Manage court-related payments, generate receipts, and track dues.
- Notification Management: Send alerts to judges, lawyers, and clients about hearings and updates.
- System Coordination: Bridge between judges, lawyers, and clients by managing requests and responses.

4. Lawyer Module

- Client Management: Register and view client profiles, case histories, and documents.
- Case Booking Management: Accept or reject case requests from clients or the court.
- Document Submission: Upload and submit evidence, affidavits, and legal documents.
- Case Updates and Tracking: Receive real-time updates about case hearings and status changes.
- Communication: Send queries or updates to staff, judges, and clients.
- Performance Dashboard: View summary of ongoing and closed cases, success rates, and upcoming tasks.

1.2 Database Design

The database design follows normalization techniques to ensure efficient data storage and retrieval. All tables have been normalized up to the second normal form to minimize redundancy and interdependency. Key tables include:

1. **Judge Register**: Stores judge information including username, email, contact details, and credentials.
2. **Staff Register**: Contains staff member details such as username, contact information, and login credentials.
3. **Client Register**: Maintains client profiles including personal details and credentials.
4. **Case**: Stores case details such as case ID, client name, case type, description, filing date, and status.
5. **Lawyer**: Contains lawyer information including expertise, experience, and contact details.
6. **Case Booking**: Records booking details including client information, case details, and booking status.
7. **Court Register**: Comprehensive table containing case details, client information, lawyer details, and defendant information.
8. **Payment**: Tracks payment transactions Including amount, status, and payment method.
9. **Room**: Contains information about courtroom allocation.
10. **Message**: Stores communication between different stakeholders.

VIII. SYSTEM TESTING

- The testing process Validation of database operations (insertion, updating, deletion)
- Form validations (email format, phone number format, required fields)
- Character limit checks
- System response to inputs involved several phases to ensure the system's reliability and functionality:

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405303|

1.3 Unit Testing

Each module was tested individually to verify its functionality:

1.4 Integration Testing

Modules were integrated and tested as a whole:

- System response after module integration
- Page linking functionality
- Database connectivity across modules
- Cross-module operations and data consistency

1.5 Validation Testing

Validation testing was performed to ensure the system meets business requirements:

- Input validation
- Business logic verification

1.6 Acceptance Testing

User acceptance testing was conducted with prospective users to ensure the system meets their requirements, particularly regarding input and output screen designs.

IX. IMPLEMENTATION

The implementation follows a systematic approach:

Database Setup: Creating and configuring the SQLite database with all required tables.

1. **Backend Development**: Implementing the business logic using Python and Django.
2. **Frontend Development**: Creating user interfaces using HTML, CSS, and JavaScript.
3. **Integration**: Connecting frontend interfaces with backend logic.
4. **Testing**: Rigorous testing of individual modules and the integrated system.
5. **Deployment**: Setting up the system in the target environment.

X. FUTURE ENHANCEMENTS

Future enhancements for the Court Management System could include:

1. **AI and ML Integration**: Implementing machine learning algorithms for case prediction, resource allocation, and decision- making support.
2. **AI-powered Chatbots**: Assisting clients and lawyers with common queries and providing updates.
3. **Natural Language Processing**: Analyzing legal documents, generating summaries, and identifying trends across cases.
4. **Blockchain Integration**: Creating tamper- proof records of case documents and court rulings.**Predictive Analytics**: Understanding peak times and resource demands for better planning.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405303|

1. **Mobile Application**: Developing a companion mobile app for real-time case tracking and notifications.
2. **Remote Video Hearings**: Implementing integrated tools for virtual court proceedings.
3. **Multi-language Support**: Making the system accessible to a more diverse population.
4. **Advanced Analytics**: Providing detailed reports on case statistics and court efficiency.

XI. CONCLUSION

The Court Management System built using Python and Django offers a modern and efficient solution to the challenges faced by traditional court management systems. By centralizing and automating processes such as case tracking, scheduling, document management, and communication, the system significantly improves the speed, accuracy, and transparency of judicial operations. The role-based modules ensure that each user can efficiently perform their tasks while maintaining security and ease of access.

The system reduces administrative burdens, minimizes errors, and enhances the overall productivity of the judicial system. This web-based platform not only streamlines court procedures but also contributes to a more organized, accessible, and user-friendly judicial environment, making it a valuable tool for modernizing court management practices.

REFERENCES

1. Sommerville, I. (2015). Software Engineering (10th ed.). Pearson Education.
2. Pressman, R. S., & Maxim, B. R. (2019). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.
3. Welling, L., & Thomson, L. (2016). PHP and MySQL Web Development (5th ed.). Addison- Wesley.
4. Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Pearson Education.
5. ISO/IEC 27001:2013. (2013). Information
6. Security Management Systems (ISMS). International Organization for Standardization.
7. Chowdhury, A. (2021). Full Stack Web Development with Django. Packt Publishing.
8. Nielsen, J. (1993). Usability Engineering. Morgan Kaufmann.
9. Django Software Foundation. (n.d.). Django Documentation.URL: https://docs.djangoproject.com
10. National Center for State Courts. (2020). Court Case Management Systems: A Guide for Implementation. URL: https://www.ncsc.org

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details