IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

"Python Full Stack Web Based User Management System"

Prof. Namrata V. Jangam [1], Sushant Dongardive [2]

[1]Faculty Computer Engineering Vidya Prasarini Sabha's Collage of Engineering and Technology, Lonavala

[2]Student Computer Engineering Vidya Prasarini Sabha's Collage of Engineering and Technology, Lonavala

ABSTRACT: Modern web applications require seamless integration between front-end and back-end technologies to provide efficient and user-friendly systems. Traditional standalone applications lack scalability, real-time interaction, and centralized data management.

This paper presents the design and implementation of a Python-based full-stack web application that integrates front-end technologies (HTML, CSS, JavaScript) with a Python back-end framework (Flask/Django) and a relational database system. The system provides secure user authentication, efficient data management using CRUD operations, and a responsive user interface.

The application demonstrates how full-stack development can be used to build scalable and maintainable systems. The implementation ensures proper data handling, user interaction, and system reliability, making it suitable for small-scale web applications and learning purposes.

Keywords: Python, Full-Stack Development, Flask, Web Application, CRUD Operations, User Authentication, Database Management

1.INTRODUCTION

With the rapid growth of web technologies, full-stack development has become essential for building modern web applications. A full-stack application combines both front-end and back-end development to create complete and functional systems.

Traditional applications often lack dynamic interaction and centralized data handling, making them less efficient for real-world usage. Python frameworks such as Flask and Django provide a powerful environment for developing scalable and efficient back-end systems.

This project focuses on developing a Python full-stack web application that allows users to perform operations such as registration, login, and data management. The system integrates front-end technologies with a back-end server and database to provide a smooth user experience.

The aim is to demonstrate the practical implementation of full-stack development concepts and provide a foundational system for future enhancements.

1.1 Literature Review

1. Full-Stack Web Development Using Python (Various Authors)

This study highlights the importance of Python frameworks like Flask and Django in developing scalable web applications. These frameworks simplify backend development and provide flexibility.

2. Web Application Development with MVC Architecture

This paper explains the Model-View-Controller (MVC) design pattern used in web applications to separate logic, UI, and data handling, improving maintainability.

3. Secure Authentication Systems in Web Applications

The research focuses on user authentication mechanisms such as login systems and password hashing to ensure secure access control.

4. Database Management in Web Applications

This study discusses how relational databases like SQLite and MySQL are used for storing and managing application data efficiently.

2.PROPOSED SYSTEM

The proposed system is a Python-based full-stack web application designed to provide secure and efficient data management through a user-friendly interface.

System Modules

User Module

- User registration and login
- Secure password storage
- Session management

Application Server (Backend)

- Handles business logic using Flask/Django
- Processes user requests
- Connects front-end with database

Database Layer

- Stores user credentials and application data
- Supports CRUD operations
- Ensures data consistency

System Working

User Registration:

- User enters details
- Data is stored securely in database

Login Process:

- User enters credentials
- System verifies and grants access

CRUD Operations:

- Users can create, view, update, and delete records

Dashboard:

- Displays stored data in structured format

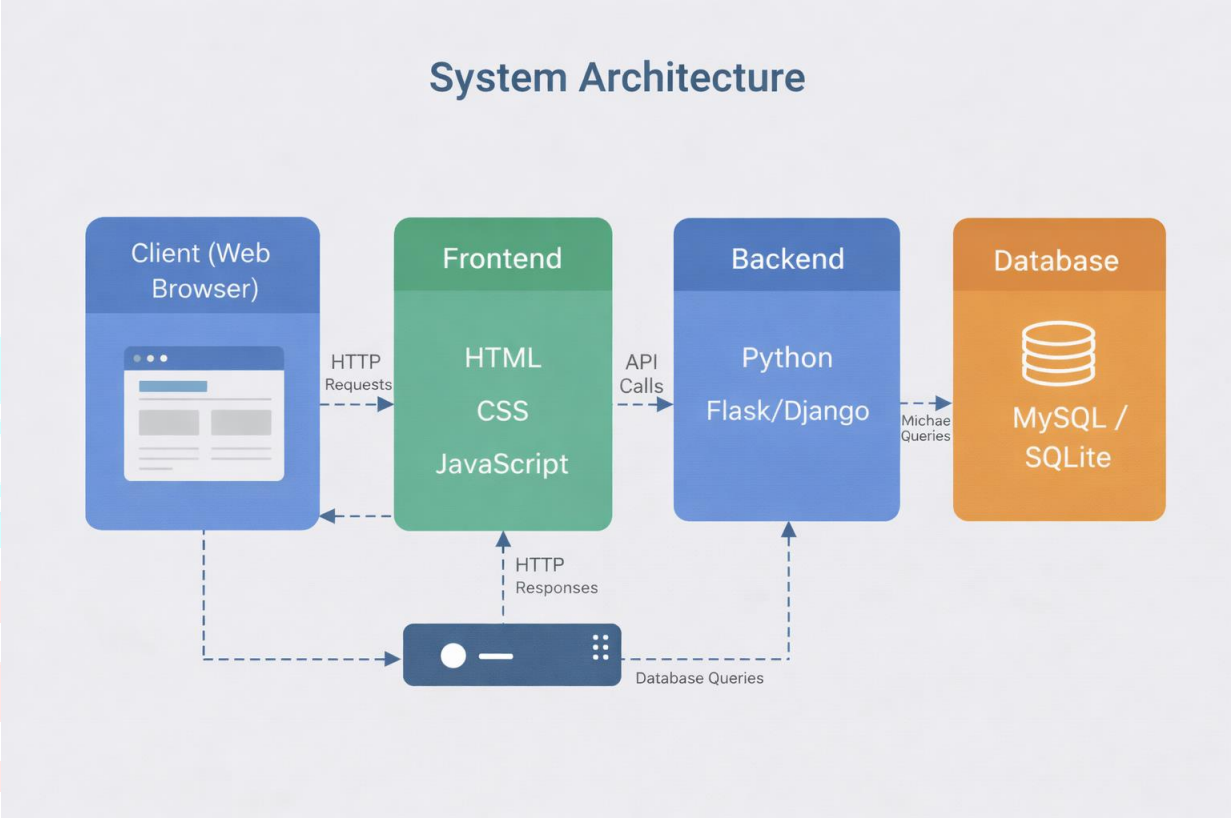

Fig 1. System Architecture

3.IMPLEMENTATION AND RESULT

The system was implemented using Python (Flask/Django), HTML, CSS, JavaScript, and SQLite/MySQL database. The application was tested under different scenarios such as user login, data entry, and record management.

The results show that the system performs efficiently for small-scale applications. The integration between front-end and back-end is smooth, and the application responds quickly to user actions.

Authentication functionality ensures that only authorized users can access the system. CRUD operations were successfully implemented, allowing users to manage data effectively.

The application provides a responsive user interface, making it accessible across different devices. Error handling mechanisms were also implemented to improve reliability.

5.CONCLUSION :

The developed Python full-stack web application successfully demonstrates the integration of front-end and back-end technologies. The system provides secure authentication, efficient data management, and a user-friendly interface.

The project highlights the importance of structured development using MVC architecture and modern web technologies. It serves as a strong foundation for building more advanced applications with additional features such as APIs, cloud deployment, and real-time data processing.

6.REFERENCES:

1. Flask Documentation – https://flask.palletsprojects.com/
2. Django Documentation – https://www.djangoproject.com/
3. W3Schools – https://www.w3schools.com/
4. SQLite Documentation – https://www.sqlite.org/docs.html
5. MDN Web Docs – https://developer.mozilla.org/