International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802232

Freelance Flow Project Management Tool for Freelancers

Srikanth G[1], Rajamanickam.E S[2]

Student-II MCA, Department of Master of Computer Applications, Er. Perumal Manimekalai College of Engineering, Hosur, Tamil Nadu, India[1]

Assistant Professor, Department of Master of Computer Applications, Er. Perumal Manimekalai College of Engineering, Hosur, Tamil Nadu, India[2]

Publication History: Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03. 2026; Published: 28.03.2026.

ABSTRACT: Freelancing has become one of the fastest-growing employment models in the digital era. However, freelancers often struggle with managing multiple clients, tracking project progress, organizing tasks, and handling payments efficiently. Existing general-purpose project management tools lack freelancer-specific features such as client billing, deadline prioritization, and financial tracking.This project, FreelanceFlow, presents a comprehensive web-based project management system designed specifically for freelancers. The application is developed using Python Flask (backend), HTML, CSS, JavaScript (frontend), and MySQL (database). The system provides functionalities such as project management, task tracking, client management, deadline monitoring, and payment tracking in a unified platform.The proposed system improves productivity, reduces manual workload, ensures better organization, and enhances transparency between freelancers and clients. The system follows a modular and scalable architecture, making it suitable for future enhancements like mobile integration and AI-based recommendations.

KEYWORDS: FreelanceFlow, Project Management System, Flask Framework, MySQL Database, Freelancer Management Tools, Task Management, Client Management, Project Tracking, Payment Tracking

I. INTRODUCTION

Freelancing is rapidly transforming into a mainstream mode of work, driven by digital platforms, remote work culture, and global connectivity that empower professionals to operate independently while serving multiple clients across borders. Yet, despite its flexibility, freelancers often struggle with inefficiencies when relying on manual tools like spreadsheets, scattered notes, or separate applications. These fragmented methods lead to missed deadlines, poor communication, lack of centralized data, and difficulties in tracking payments—all of which hinder productivity and client satisfaction.

To address these challenges, **FreelanceFlow** emerges as a comprehensive, centralized web application tailored specifically for freelancers. It integrates essential functions such as project management, task tracking, client communication, deadline monitoring, and payment management into a single streamlined platform. By automating routine activities—like sending reminders, generating invoices, and updating task progress—FreelanceFlow reduces administrative burdens and allows freelancers to focus on delivering quality work.

II. LITERATURE REVIEW

Many project management tools and research studies have contributed to improving organizational workflows, but they often lack specialization for freelancers who operate independently. Tools like Trello, Asana, ClickUp, and Notion provide valuable features such as Kanban boards, structured workflows, all-in-one platforms, and customizable workspaces. However, each comes with limitations when applied to freelancing: Trello lacks financial tracking, Asana is overly complex for individuals, ClickUp is overloaded with features, and Notion requires extensive manual setup. Research further highlights that centralized systems can boost productivity by 30–40%, while database-driven solutions like MySQL ensure consistency and integrity. Additionally, lightweight frameworks such as Flask are widely adopted for building scalable applications, making them suitable for freelancer-oriented platforms.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802232

III. SYSTEM ARCHITECTURE AND DATABASE DESIGN

The system is architected around the Flask framework, which handles URL routing, request/response cycles, and server-side logic. The core components are:

- **Frontend (View):** Rendered using Jinja2 templating engine, integrated with HTML5, CSS3, and Bootstrap to create a dynamic and responsive user interface.
- **Backend Logic (Controller):** Implemented in Python using Flask. It processes form data, handles user authentication, manages sessions, and interacts with the database.
- **Database (Model):** A MySQL database stores all persistent data. The database schema follows a normalized structure as shown below:

Database Tables

Users

- user_id (Primary Key)
- username
- email
- password_hash
- role (admin/freelancer/client)
- created_at

Freelancer_Profiles

- profile_id (Primary Key)
- user_id (Foreign Key)
- full_name
- skills
- experience_level
- portfolio_link
- phone_number

Clients

- client_id (Primary Key)
- user_id (Foreign Key)
- company_name
- contact_person
- email
- phone_number
- address

Projects

- project_id (Primary Key)
- freelancer_id (Foreign Key → Users)
- client_id (Foreign Key → Clients)
- project_title
- description
- start_date
- deadline
- status (Pending / In Progress / Completed / Cancelled)
- budget

Tasks

- task_id (Primary Key)
- project_id (Foreign Key)
- task_name
- description
- priority (Low / Medium / High)
- status (To Do / In Progress / Done)
- assigned_date
- due_date

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802232

Payments

- payment_id (Primary Key)
- project_id (Foreign Key)
- payment_status (Pending / Paid / Overdue)
- payment_date
- payment_method

Invoices

- invoice_id (Primary Key)
- project_id (Foreign Key)
- description
- invoice_number
- issue_date
- due_date
- total_amount
- status (Unpaid / Paid)

IV. EXISTING SYSTEM LIMITATIONS

The conventional methods used by freelancers to manage projects and clients are affected by several critical issues:

- **Manual Processes:** Freelancers often rely on notebooks, spreadsheets, or multiple disconnected tools to manage tasks, projects, and client information. This process is time-consuming, unorganized, and prone to human errors.
- **Lack of Real-Time Information:** There is no centralized system to track project progress, deadlines, or task completion in real time. This leads to missed deadlines and poor time management.
- **Data Redundancy and Insecurity:** Storing data across multiple platforms (emails, spreadsheets, notes) increases the risk of data duplication, inconsistency, and unauthorized access. Important client or project data can be lost or mismanaged.
- **Inefficient Reporting:** Freelancers often struggle to track payments, invoices, and pending dues. Manual financial tracking leads to errors, missed payments, and lack of proper financial records.
- **User Inconvenience:** Managing multiple clients and projects without a unified platform increases workload, reduces productivity, and creates unnecessary complexity in daily operations.

V. PROPOSED SYSTEM: FLASK-BASED FREELANCEFLOW MANAGEMENT SYSTEM

Our proposed system, **FreelanceFlow**, is designed as a direct solution to the limitations faced by freelancers in managing projects, clients, tasks, and payments. The system is built using the **Python Flask framework** for its flexibility, lightweight nature, and strong support for web application development.

- **Flask-Driven Workflow:** All user requests are handled through the Flask framework, which manages the entire business logic of the application. This includes user authentication, project creation, task assignment, client management, payment tracking, and real-time status updates. Flask ensures smooth routing and efficient handling of user interactions.
- **Secure Authentication and Session Management:** The system uses **Flask-Login** for secure authentication and session handling. Each user (Admin, Freelancer, Client) is provided role-based access, ensuring that users can only access functionalities and data relevant to their role. This enhances system security and data privacy.
- **Dynamic Content Rendering:** The **Jinja2 templating engine** is used to dynamically render web pages. This allows real-time display of important data such as project progress, task status, deadlines, payment details, and personalized dashboards for freelancers and clients.
- **Robust Database Integrity:** The system uses **Flask-SQLAlchemy** as an ORM to interact with a well-structured MySQL database. This ensures proper data management, reduces redundancy, and maintains consistency across all modules such as projects, tasks, clients, and payments.
- **Modular and Scalable Architecture:** The application is designed with a modular structure, separating frontend, backend, and database layers. This makes the system easy to maintain, scalable, and adaptable for future enhancements such as mobile applications, API integrations, and AI-based features.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802232

VI. SYSTEM MODULES

The FreelanceFlow application is modularized into three core components, each with dedicated routes and templates managed by the Flask framework. These modules ensure proper separation of responsibilities and efficient system functionality.

6.1 Admin Module (/admin routes)

This module provides system administrators with complete control over the platform. It is responsible for monitoring system activities, managing users, and generating reports.

Key functionalities include:

• Secure system login
• Creation and management of freelancer and client user accounts
• Monitoring all projects, tasks, and transactions in the system
• Managing platform settings and configurations
• Access to analytical dashboards and reports (e.g., total projects, completed tasks, pending payments)
• Viewing system-wide performance statistics and user activity

6.2 Freelancer Module (/freelancer routes)

This module is the core component of the system, designed specifically for freelancers to manage their daily work efficiently.

Key functionalities include:

• Secure login and personalized dashboard
• Creation and management of projects
• Task creation, assignment, and tracking with priorities and deadlines
• Client management (adding, updating, and maintaining client details)
• Tracking project progress and status updates (Pending, In Progress, Completed)
• Invoice generation and payment tracking (Paid, Pending, Overdue)
• Uploading and managing project-related documents
• Communication with clients through messages or notifications

6.3 Client Module (/client routes)

This module provides a user-friendly interface for clients to interact with freelancers and monitor their project progress.

Key functionalities include:

• Client registration and secure login
• Viewing assigned projects and their current status
• Monitoring task progress and deadlines
• Accessing invoices and payment details
• Communicating with freelancers regarding project requirements or updates
• Viewing and downloading shared documents
• Profile management to update personal and company detail

VII. IMPLEMENTATION SUMMARY

The development of the **FreelanceFlow Project Management Tool** followed a structured and systematic approach to ensure efficiency, scalability, and reliability of the application.

- **Environment Setup:** A Python virtual environment was configured to manage dependencies. Key packages installed included Flask, Flask-SQLAlchemy, Flask-Login, Flask-WTF (for forms), and a MySQL connector.
- **Database Modeling:** The database schema was implemented using Flask-SQLAlchemy models. Each entity was represented as a Python class, simplifying database interactions and ensuring type safety.
- **Route and View Development:** Flask routes were meticulously defined to handle all HTTP requests (GET, POST). Each route corresponds to a specific functionality, such as /user/apply_llr or /rto/pending_applications. The view functions contain the logic to process requests, interact with the database, and render the appropriate Jinja2 templates.
- **Frontend Integration:** The Jinja2 templates were styled using Bootstrap CSS framework to create a responsive, intuitive, and professional user interface that works seamlessly on both desktop and mobile devices.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802232

- **Testing and Validation:** The application underwent rigorous testing. This included unit testing of individual functions, integration testing of user workflows, and validation of user access controls to ensure security.

VIII. RESULTS & DISCUSSION

The developed **FreelanceFlow Project Management Tool**, built using the Flask framework, was successfully implemented and tested. The system demonstrates significant improvements over traditional freelance management methods. The key outcomes are as follows:

- **Enhanced Security and Access Control:** The system implements a robust **role-based access control mechanism** using Flask-Login. It ensures that Admins, Freelancers, and Clients can only access functionalities and data relevant to their roles. This improves data security and prevents unauthorized access.
- **Operational Efficiency:** The automation of project management activities such as task assignment, deadline tracking, and payment monitoring has significantly reduced manual effort. Freelancers can now manage multiple projects efficiently through a centralized dashboard.
- **Improved Transparency and User Experience:** The automation of project management activities such as task assignment, deadline tracking, and payment monitoring has significantly reduced manual effort. Freelancers can now manage multiple projects efficiently through a centralized dashboard.
- **Effective Financial Management:** The system provides integrated features for **invoice generation and payment tracking**, enabling freelancers to monitor pending, completed, and overdue payments accurately. This reduces financial errors and improves cash flow management.
- **Improved User Experience:** The application offers a **user-friendly and responsive interface** built with Bootstrap. Users can easily navigate through dashboards, manage projects, and access important information without technical difficulty.

IX. CONCLUSION

The **FreelanceFlow Project Management Tool**, developed using the Python Flask framework, presents a modern, efficient, and scalable solution to the challenges faced by freelancers in managing projects, clients, tasks, and payments. By leveraging a lightweight yet powerful technology stack, the system successfully automates core.

This project delivers a comprehensive and user-friendly solution that enhances productivity for freelancers, improves transparency between freelancers and clients, and simplifies overall project management. The system reduces manual effort, minimizes errors, and provides real-time insights into project progress and financial status.

X. FUTURE WORK

The system is designed with scalability and flexibility in mind. Several enhancements can be implemented in the future to further improve its functionality and user experience:

- Integration of a secure online payment gateway for fee collection
- Implementation of SMS and email notification systems to alert users about application status updates
- Development of a RESTful API to enable a companion mobile application for iOS and Android
- Incorporation of advanced features like biometric authentication and QR code generation on digital licenses

REFERENCES

1. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.
2. Juszkiewicz, "The use of Adobe Flex in combination with Java EE technology on the example of ticket booking system", in CAD Systems in Microelectronics (CADSM), 2011.
3. C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor &Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
4. C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of Electrical Engineering, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
5. C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802232

6. S.Tamilselvi, R.Prakash, C.Nagarajan,"Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller" Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering, DOI10.1007/s40998-025-00917-z,2025
7. S.Tamilselvi, R.Prakash, C.Nagarajan," Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance" Electric Power Systems Research 253 (2026) 112428, doi.org/10.1016/j.epsr.2025.112428
8. S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
9. C. Nagarajan, M.Madheswaran and D.Ramasubramanian- 'Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model'- Acta Electrotechnica et Informatica Journal , Vol.13 (2), pp.18-31,April-June.2013, DOI: 10.2478/aeei-2013-0025.
10. C.Nagarajan and M.Madheswaran - 'DSP Based Fuzzy Controller for Series Parallel Resonant converter'- Springer, Frontiers of Electrical and Electronic Engineering, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
11. C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.
12. C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
13. Suganthi Mullainathan, Ramesh Natarajan, "An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques", Revista Materia (Rio J.) Vol. 30, 2025, DOI: https://doi.org/10.1590/1517-7076-RMAT-2024-0721
14. M Suganthi, N Ramesh, "Treatment of water using natural zeolite as membrane filter", Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
15. Bazghandi, "Web Database Connectivity Methods (using Mysql) in Windows Platform", in Information and Communication Technologies, 2009.
16. Xiaosheng Yu, Yichang, China Cai Yi, "Design and Implementation of the Website Based on PHP & MYSQL", in E-Product E-Service and E-Entertainment (ICEEE), 2010.
17. Flask Official Documentation (https://flask.palletsprojects.com/)
18. MySQL Official Documentation (https://dev.mysql.com/doc/)
19. Bootstrap Official Documentation (https://getbootstrap.com/docs/)
20. MATHEW, A. (2025). BEYOND THE BURNER: THE SYSTEMIC RISKS OF DISPOSABLE EMAIL ECOSYSTEMS.
21. Raj, A. M. A., Rajendran, S., & Vimal, G. S. A. G. (2024). Enhanced convolutional neural network enabled optimized diagnostic model for COVID-19 detection. *Bulletin of Electrical Engineering and Informatics, 13(3), 1935-1942.*
22. Kiran, A., Rubini, P., & Kumar, S. S. (2025). Comprehensive review of privacy, utility and fairness offered by synthetic data. *IEEE Access.*
23. Anand, L., & Syed Ibrahim, S. P. (2018). HANN: a hybrid model for liver syndrome classification by feature assortment optimization. *Journal of Medical Systems, 42(11), 211.*
24. Udayakumar, R., Yogesh Pansambal, S., Anbazhagan, K., & Sugumar, R. Real-time Migration Risk Analysis Model for Improved Immigrant Development Using Psychological Factors. *Migr Lett. 2023; 20 (4): 33–42.*