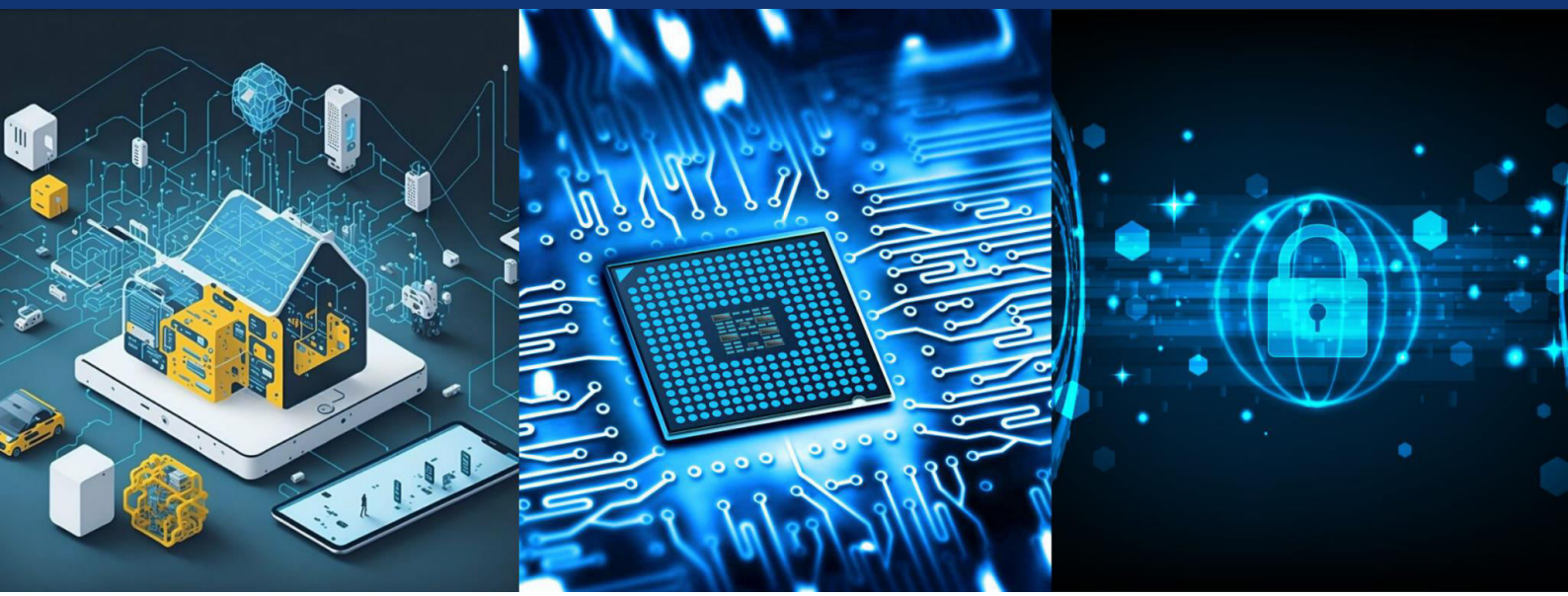
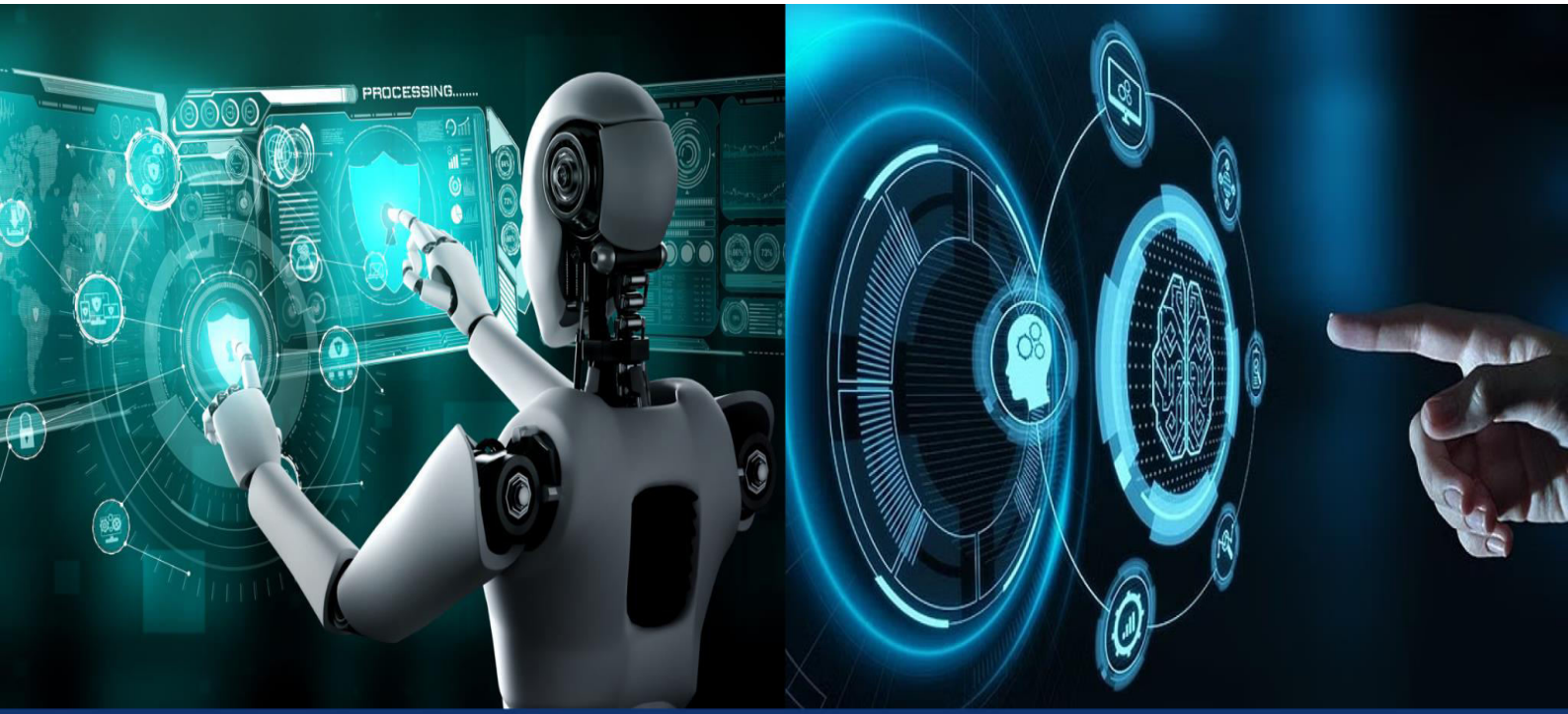
ISSN(O): 2320-9801
ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

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



Impact Factor: 8.771 **Volume 13, Issue 5, May 2025**

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305176

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

Digital Studio ERP using React in Full Stack Development

R J Poovaraghan [1], R Srinivasan [2], Pavithra S[3], Thrishulla A[4], Harini A[5], Jananee G[6]

Assistant Professor, Dept. of Information Technology,Jaya Engineering College, Chennai, Tamil Nadu, India [1-2]

UG Student, Dept. of Information Technology, Jaya Engineering College, Chennai, Tamil Nadu, India[3-6]

ABSTRACT: Full Stack Development involves building both the frontend and backend of web applications using technologies like HTML, CSS, JavaScript, React, Node.js, .NET Core, and MySQL. Developers manage UI/UX, APIs, databases, security, and deployment, facing challenges like tech overload, scalability, and debugging across layers. The current business system relies on manual processes, disconnected tools, and lacks automation, causing inefficiencies, errors, and poor scalability. Client requests are managed via emails or calls, while employees use spreadsheets, making tracking and collaboration difficult. To resolve these issues, a modern web application is proposed. It offers centralized automation for client requests, employee task tracking, and admin control. With modules like a public website, client portal, employee portal, and admin portal, it enhances transparency, real-time updates, and operational efficiency. Built using React.js, .NET Core Web API, MySQL, and Azure DevOps, the system ensures seamless communication, security, and scalability, transforming business and task management into an efficient digital workflow.

KEYWORDS: Full Stack Web Development, Freelancer Management System, React.js, NET Core Web API, MySQL, Azure DevOps,Security

I.INTRODUCTION

Full Stack Web Development encompasses three major domains: Front-End Development, Back-End Development, and DevOps & Deployment. These domains work together to build complete, functional, and scalable web applications.

Front-End Development, also known as client-side development, is focused on creating the user interface and ensuring a seamless user experience. It involves using core technologies like HTML for structuring content, CSS for styling and layout, and JavaScript for adding interactivity. Modern frameworks such as React.js, Angular, and Vue.js further enhance front-end development by allowing developers to build reusable components and create dynamic, responsive web interfaces. Front-end developers prioritize accessibility, responsiveness, and compatibility across various devices and browsers to ensure that users have a smooth and consistent experience.

Back-End Development, or server-side development, is responsible for the underlying logic, database management, and user authentication. It processes incoming requests from the front-end, interacts with the database, and sends appropriate responses back to the client. Popular back-end programming languages and frameworks include Node.js for JavaScript-based development, Python with frameworks like Django and Flask, Java with Spring Boot, and PHP with Laravel. Data storage and retrieval are handled by databases, which can be either SQL-based (such as MySQL and PostgreSQL) or NoSQL-based (like MongoDB and Firebase). Communication between the front-end and back-end is facilitated through APIs, such as RESTful APIs is the Security is a key concern in back-end development, with practices like JWT authentication, OAuth protocols, and password hashing used to protect user data and application integrity.

The DevOps & Deployment domain ensures the smooth delivery, scalability, and maintenance of web applications. This includes version control using systems like Git and platforms such as GitHub for collaboration and tracking changes. Cloud platforms like Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure provide infrastructure for hosting and scaling applications. Tools like Docker and Kubernetes are used for containerization and orchestration, allowing applications to run consistently across different environments. Continuous Integration and Continuous Deployment (CI/CD) pipelines automate the process of testing and deploying code, which enhances development efficiency and reduces the chances of errors. Web servers like Nginx and Apache manage

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305176

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

incoming traffic and ensure efficient content delivery.

II. SYSTEM MODEL AND ASSUMPTIONS

The system architecture is based on Clean Architecture principles, ensuring a well-organized and maintainable structure by dividing the application into four main layers. The Presentation Layer is developed using React.js, which handles all client- side interactions and delivers a responsive user interface. The Application/Domain Layer is built with .NET Core Web API, responsible for executing business logic, managing workflows, and processing requests from the frontend. Data management is handled by the Data Layer, using a MySQL database to securely store user information, tasks, and transaction records. The Infrastructure Layer supports the system with cloud storage through AWS S3 and deployment via Azure DevOps, enabling scalability and efficient resource management.

Several assumptions guide the system's design and functionality. It is assumed that all users—Admins, Clients, and Employees— access the platform through authenticated login mechanisms, ensuring secure and role-based access to features. A stable internet connection is considered available for all interactions with the platform. Freelancers are able to browse available tasks, place bids, and submit completed work digitally through the system. Admins play a central role in overseeing operations, managing task assignments, and handling approvals, ensuring smooth workflow across users. The platform also includes integrated, secure payment processing to manage financial transactions between clients and freelancers.

This layered and modular approach not only enhances system security, performance, and scalability but also makes development, testing, and maintenance more efficient. The use of modern technologies and cloud-based tools ensures that the platform can handle a wide range of user interactions and data processing tasks effectively, supporting a reliable and user-friendly digital environment for freelance project management.

EFFICIENT COMMUNICATION

Efficient communication is a core feature of the proposed system, designed to streamline collaboration and ensure smooth workflow management. This is achieved through multiple integrated mechanisms that promote clarity, responsiveness, and accountability. Real-time notifications keep users instantly informed about critical updates such as new task assignments, bid statuses, and approaching deadlines, reducing the chances of missed information or delays. The platform provides role-specific dashboards for Admins, Clients, and Employees, allowing each user to view and manage information relevant to their responsibilities in a centralized and organized manner. A built-in task tracking system offers visibility into project progress, including updates on milestones and completion status, enabling users to monitor deliverables efficiently. In-app messaging and alerts eliminate the need for external communication tools, keeping all conversations and updates within the platform to enhance data security and accessibility. Additionally, the system employs structured APIs that facilitate seamless and consistent data exchange between the front-end and back-end components, ensuring that all interactions and updates reflect accurately across the application. These communication features collectively enhance coordination among users, reduce turnaround time, and improve overall productivity by ensuring that everyone involved in a project stay informed and aligned at every stage.

SECURITY

Security is a foundational pillar of the proposed system, integrated at every level to protect user data, ensure that access is limited to authorized individuals, and maintain the integrity and confidentiality of workflows. Given the growing landscape of cybersecurity threats, the system adopts a multi-layered security architecture that combines strong user authentication, robust data protection mechanisms, secure infrastructure deployment, and vigilant session management. This comprehensive approach is designed to address both internal and external risks, creating a secure environment that fosters trust among users and stakeholders.

At the core of the system's access control strategy lies Role-Based Access Control (RBAC), which defines and enforces user permissions based on clearly defined roles: Admins, Clients, and Employees. This ensures that each user is only granted access to the data and functionality required for their role, minimizing the potential for unauthorized access or accidental exposure of sensitive information. For instance, Admins can configure system-wide settings, manage user accounts, and monitor overall system performance. Clients, on the other hand, are restricted to viewing and managing their own data, without visibility into other users' information. Employees are given access relevant to their assigned tasks and projects, with no unnecessary exposure to administrative or client-level functionalities. This fine-grained

control model significantly reduces the attack surface and enforces the principle of least privilege, a core tenet of modern security design.

To further enhance identity verification, the system incorporates Multi-Factor Authentication (MFA), adding a critical layer of security beyond standard username and password login mechanisms. MFA requires users to authenticate through an additional method—such as a one-time code sent to a registered mobile device or email—before granting access. This measure ensures that even if login credentials are compromised, unauthorized users cannot gain entry without the second form of verification. MFA is applied particularly to high-privilege roles and sensitive operations, strengthening the system's defense against account takeovers and brute-force attacks.

Data protection is another cornerstone of the platform's security strategy. All sensitive data, including user credentials, personal identifiable information (PII), and financial records, are encrypted both at rest and in transit. The system uses industry-standard encryption protocols such as AES-256 for data storage and TLS 1.2+ for network communications. This ensures that even if data is intercepted or accessed improperly, it remains unreadable and secure. Regular key rotations and secure key management practices are also implemented to maintain the integrity of encryption protocols over time.

The deployment pipeline is designed with security as a primary consideration. Built using Continuous Integration and Continuous Deployment (CI/CD) practices on Microsoft Azure, the system automates code testing, validation, and approval before releasing updates to production. Each code commit is subjected to automated static code analysis, unit testing, and security scanning to identify potential vulnerabilities early in the development lifecycle. This automated pipeline not only accelerates deployment but also reduces the likelihood of introducing exploitable flaws into the production environment. Secure deployment practices, including environment segregation and least-privilege service accounts, further ensure that production resources remain insulated from development and testing environments.

For file storage, the system leverages Amazon Web Services (AWS) S3 buckets configured with strict access control policies. These policies enforce identity and access management (IAM) rules that allow only authenticated and authorized users to access or manipulate stored files. S3 bucket encryption is enabled by default, and access logs are maintained for audit purposes. The use of S3 also allows for integration with AWS's broader suite of security tools, including anomaly detection and threat intelligence, offering an additional layer of infrastructure-level protection.

III. RESULT AND DISCUSSION

Fig A1.1, Login page interface of the EDIT OK platform, featuring username/email and password fields with login and Google sign-in options.

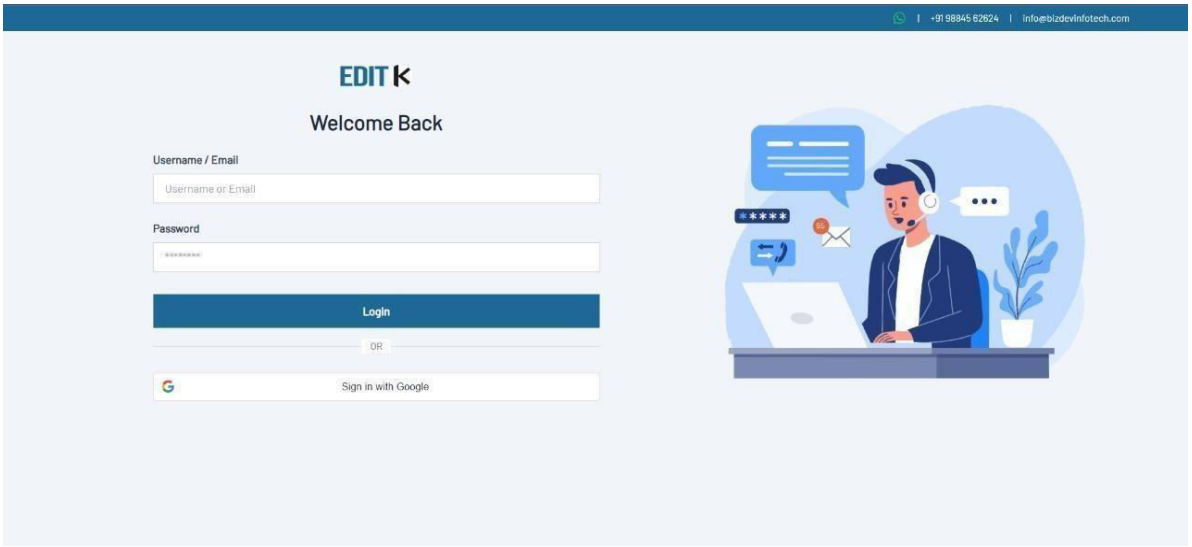


Fig A1.1 Login page

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305176

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

Fig A1.2 , Homepage of the EDIT OK platform showcasing a team of professional editors and their specialization in album designing, video editing, photoretouching, and short video creation

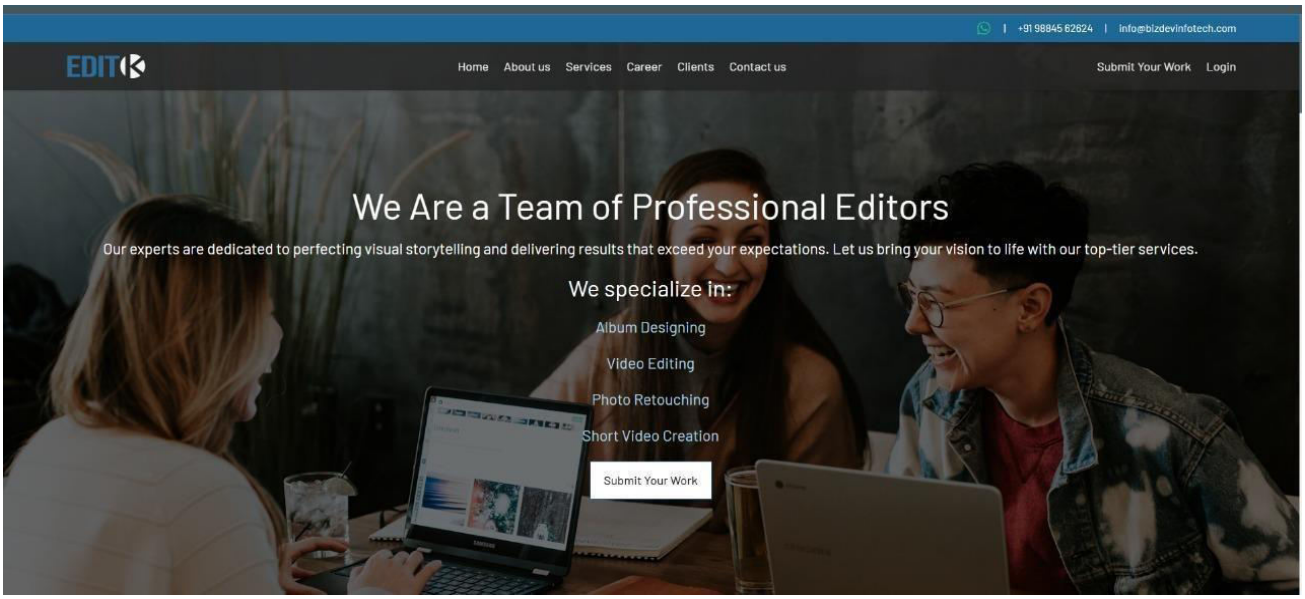


Fig A1. Sign Up

Fig A1.3, Admin dashboard of the Edit OK platform displaying order and employee and customers statistics with status breakdowns. It also shows recent user activities including logins and task updates.

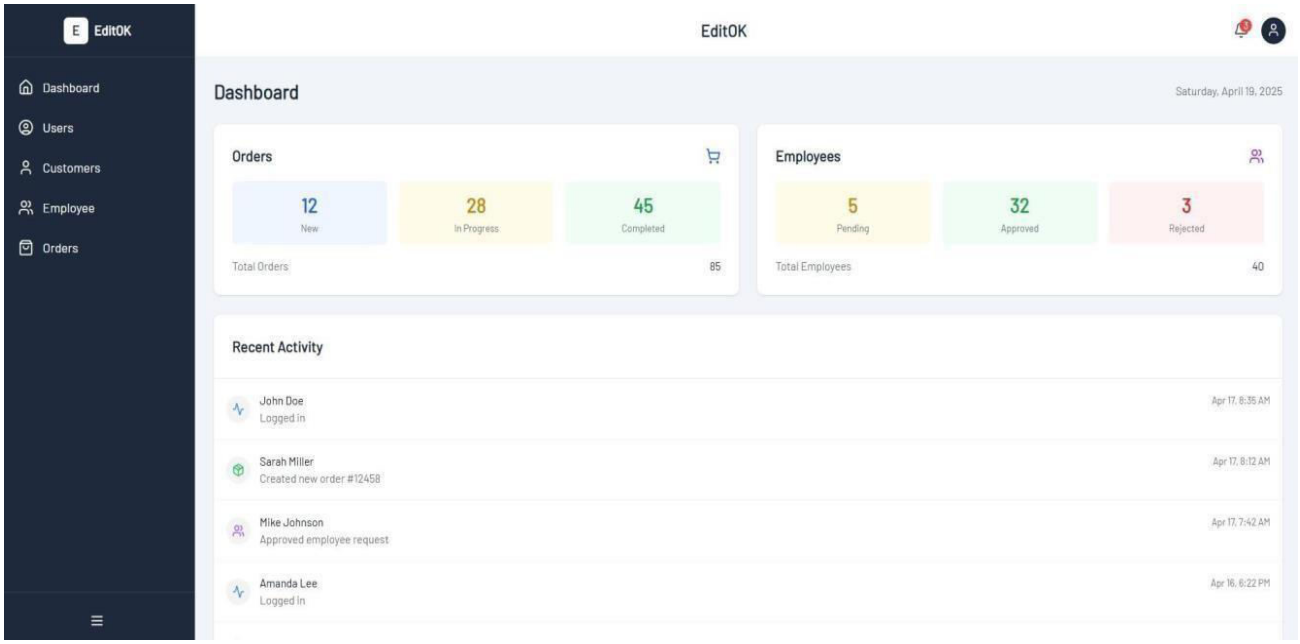


Fig A1.3 Admin portal

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305176

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

Fig A1.4, Employee dashboard of the Edit OK platform Includes Task and Earnings. The sidebar includes navigation to works, profile, and earnings, with recent activity logs listed below.

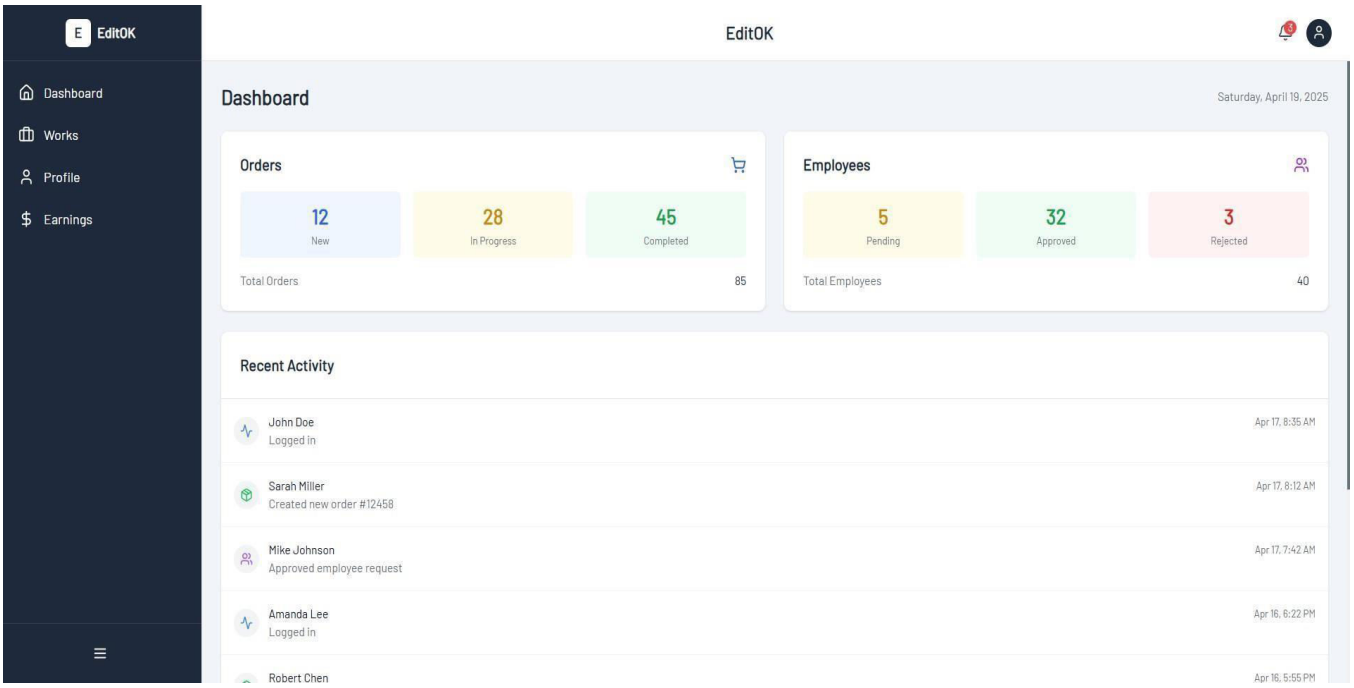


Fig A1.4List Of Employee Dashboard

Fig A1.5, Customer dashboard of the displaying order and employee status summaries. It includes recent activity logs and a sidebar with navigation options for dashboard and orders.

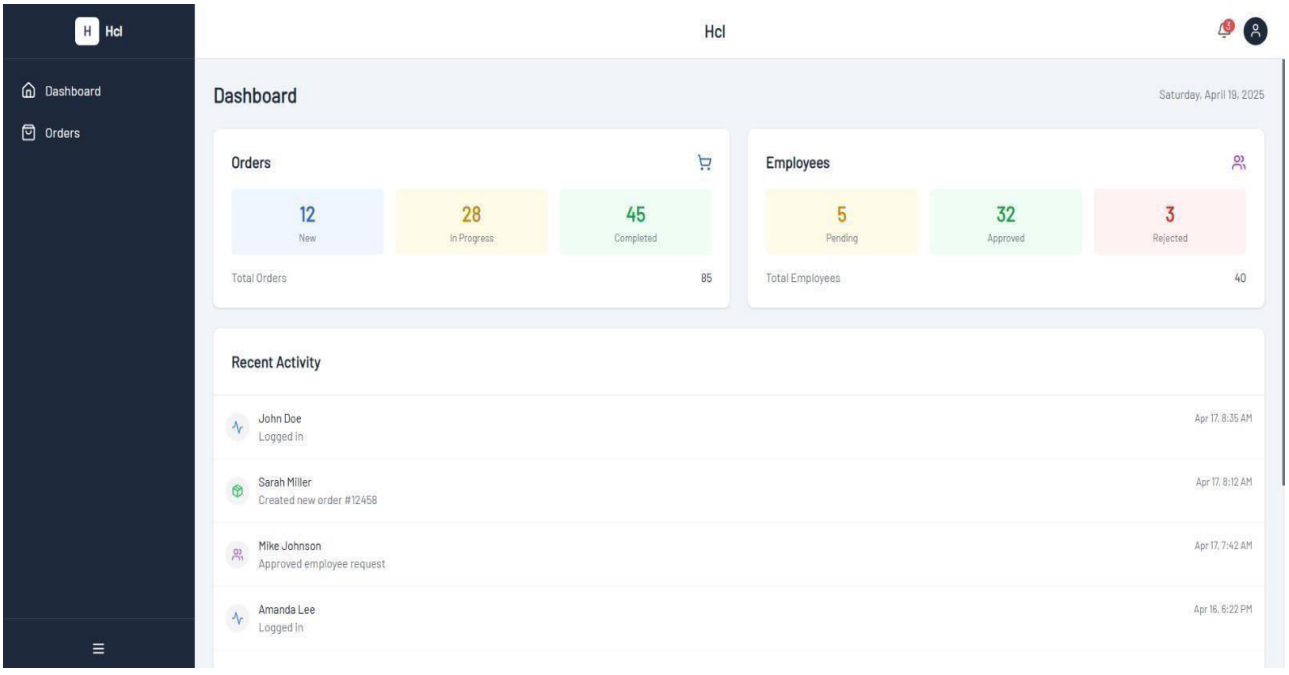


Fig A1.5 Customer Dashboard

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

IV. CONCLUSION

The project introduces a well-structured and efficient management system designed specifically for service-based platforms. It integrates four essential modules—Admin, Client, Employee, and Project each playing a critical role in the overall workflow. This integration ensures seamless coordination among stakeholders, supporting a collaborative environment where tasks are clearly defined, allocated, and tracked. The Admin module serves as the control center, where administrators manage operations, assign tasks based on employee skill sets, and ensure optimal use of resources. This helps in reducing redundancy and improving efficiency across the board. Clients benefit from enhanced transparency and engagement through the Client module. They can monitor the real-time status of their projects, raise queries or concerns, and receive timely responses. This builds trust and encourages better client relationships. On the other hand, the Employee module empowers team members by providing a clear view of their assigned tasks and deadlines. Employees can update their progress, complete tasks methodically, and contribute effectively to the success of each project. The Project module acts as a centralized hub where all project-related data and workflows are maintained. It ensures that project timelines are met and communication gaps are minimized. By reducing manual effort and providing automated task tracking, the system significantly boosts productivity. Moreover, the structured flow of data and user interactions helps prevent errors and delays, resulting in smoother project execution.

REFERENCES

1. Sharma, R., *Client-Admin Interaction in Web Systems*, International Journal of Computer Applications, ISSN: 0975-8887, Vol. 34(2), pp. 45–49, 2019.
2. Kumar, A., *Project Management Tools in Software Engineering*, Software Engineering Journal, ISSN: 2040- 7459, Vol. 21(3), pp. 22–30, 2020.
3. Singh, M., *Effective Task Allocation Using Admin Modules*, IJCSIT, ISSN: 0975-9646, Vol. 10(1), pp. 12–18, 2021.
4. Jain, R., *Client-Based Web Portals*, International Journal of Computer Science and Technology, ISSN: 2229- 4333, Vol. 9(2), pp. 33–40, 2018.
5. Mehta, D., *Dynamic Project Tracking*, Journal of Information Technology, ISSN: 1741- 1122, Vol. 15(4), pp. 50–55, 2020.
6. Banerjee, T., *Admin Control Systems in Web Projects*, IJARCSSE, ISSN: 2277-128X, Vol.
7. 8(6), pp. 60–65, 2019.
8. Kapoor, S., *Employee Management in IT Solutions*, IOSR-JCE, ISSN: 2278-0661, Vol. 12(2), pp. 22–27, 2021.
9. Thomas, R., *Project Module Design in Client Interfaces*, ACM Transactions on Software Engineering, ISSN: 1049-331X, Vol. 25(7), pp. 100–108, 2019.
10. Gupta, P., *Workflow Automation and Collaboration*, Journal of Computer Networks, ISSN:
11. [11] 2394-4714, Vol. 5(3), pp. 30–36, 2020.
12. Roy, H., *Web-Based Task Assignment Systems*, IJCSMC, ISSN: 2320-088X, Vol. 6(8), pp. 41–47, 2018.
13. Shah, A., *Development Workflow in Client Projects*, International Journal of Emerging Technology Advanced Engineering, ISSN: 2250-2459, Vol. 7(9), pp. 18–24, 2020.
14. Tyagi, S., *Monitoring Project Lifecycle*, IJSR, ISSN: 2319-7064, Vol. 9(4), pp. 50–55, 2020.
15. Desai, P., *Web-Based Systems for Freelancers*, Journal of Computer Applications, ISSN:
16. [16] 0974-1925, Vol. 12(5), pp. 33–39, 2021.
17. [17] Rao, N., *Admin-Controlled Workflow Models*, IJITEE, ISSN: 2278-3075, Vol. 8(12), pp.
18. [18] 60–67, 2020.
19. Narayan, B., *Client Interaction Portals*, Journal of Digital Communication, ISSN: 2162- 7199, Vol. 4(3), pp. 42–48, 2018.
20. Batra, A., *Project-Based Collaboration Systems*, IEEE Access, ISSN: 2169-3536, Vol. 6, pp. 1234–1241, 2020.
21. Yadav, R., *Structured Communication in Admin-Client Models*, JETIR, ISSN: 2349-5162, Vol. 7(5), pp. 11– 18, 2019.
22. [22][10] Kohli, N., *Modern Software Collaboration Tools*, IJRASET, ISSN: 2321-9653, Vol. 6(10), pp. 70–75, 2020.
23. Patil, M., *User Role-Based Access Control*, International Journal of Advanced Research in Computer Science, ISSN: 0976-5697, Vol. 10(7), pp. 34–40, 2019.
24. Sharma, P., *Project Status Modules in IT Systems*, IJCSE, ISSN: 0976-5166, Vol. 8(9), pp. 29–35, 2021.
25. Pillai, S., *Handling Client Requirements Digitally*, International Journal of Computer Applications Technology and Research, ISSN: 2319-8656, Vol. 4(1), pp. 18–22, 2018.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details