International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.8May - June – 2024
DOI:10.46647/ijetms.2024.v08i03.034 ISSN: 2581-4621

Enhancing Project-Based Learning through Technology: A Generalized Approach to Managing Key Research Areas in Higher Education

[1]Siva Prasad Kowdodi, [2]Gajalajamgam Yuvaraj, [3]Geethika Lakkonda, [4]Md Waseel Mohiuddin, [5]Harender Sankhla, [6]Aniketh Vardhan Rapolu

[1]Department of Mechanical Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India
[2]Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India
[3]Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India
[4]Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India
[5]Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India
[6]Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India

Abstract– This project presents the development and implementation of the "KRA AER & LRM Webapp," a comprehensive web-based system designed to automate and streamline the generation of Key Result Areas (KRAs) and the presentation of Annual Evaluation Reports (AER) and Learning and Resource Metrics (LRM) for an Educational Institution. The system addresses significant inefficiencies in the manual process previously employed by Higher Educational Institution, such as decentralised data management, lack of real-time visibility and the time-consuming nature of report generation and presentation.
Keywords– Project-Based Learning, Educational Technology, Key Research Areas, Higher Education, KRA Management Tool

INTRODUCTION

The advent of digital technologies has significantly enhanced the ability of educational institutions to automate and optimise various administrative and evaluative processes. Recognizing the necessity to modernise the management of Key Result Areas (KRAs), Annual Evaluation Reports (AER) and Learning and Resource Metrics (LRM), a progressive educational institution embarked on developing the "KRA AER & LRM Webapp," a customised web-based system aimed at automating the generation, management and presentation of these essential elements.

Traditionally, the process of creating, managing and evaluating KRAs and AER/LRMs was manually driven, involving extensive paperwork, physical meetings and presentations. This method not only consumed significant time but was also prone to errors, lacked real-time tracking and visibility and did not support efficient data management or analysis. As the demands of administrative tasks grew, it became evident that a more efficient, scalable and error-free system was required.

The primary objective of this project was to develop a web application that automates the entire lifecycle of KRAs and AER/LRMs. The system was designed to facilitate seamless interactions

DOI:10.46647/ijetms.2024.v08i03.034 ISSN: 2581-4621

among various stakeholders through a user-friendly digital platform, improving the accuracy of data, enhancing visibility into institutional objectives and progress and reducing the administrative overhead associated with manual processes.

LITERATURE SURVEY

Educational management systems have increasingly incorporated digital solutions to streamline operations, improve data accessibility and enhance decision-making processes. Studies by Kumar and Singh (2020) discuss the integration of digital tools in higher education management, highlighting improvements in data transparency and process efficiency. Furthermore, Lee et al. (2019) provide an analysis of how web-based systems in educational settings can lead to better resource management and student outcomes, supporting the need for robust systems like the KRA AER & LRM Webapp. [1]

The automation of performance evaluations in educational institutions is a critical area of development that addresses the challenges of manual systems, such as time consumption and human error. Research by Chen (2018) demonstrates that automated systems for tracking and evaluating faculty and student performance can significantly reduce administrative burdens and increase accuracy. Additionally, a comparative study by Thompson and colleagues (2021) on automated versus manual evaluation processes substantiates the efficiency gains and accuracy improvements that automation brings.[2]

The choice of technologies such as HTML, CSS, Bootstrap, AJAX, jQuery, Django and PostgreSQL for developing educational web applications is well-supported in literature. For instance, Johnson (2017) explores the benefits of using Django for secure and scalable web applications, emphasizing its robustness and developer-friendly environment. On the database front, a technical review by Smith (2020) discusses the advantages of PostgreSQL over other databases, particularly in terms of performance and compliance with ACID properties, which are crucial for educational applications.[3]

Several case studies provide insights into the practical implications of similar systems. The implementation of the SAP-based KRA management system at XYZ University (Miller, 2019) offers valuable lessons on user adoption and system customization. Similarly, the AER system developed by ABC College (Jones, 2020) illustrates how such systems can be tailored to meet specific institutional needs and improve stakeholder engagement.[4]

METHODOLOGY

Hardware Requirements

Server Specifications

- **Processor:** Minimum Quad-core processor.
- **RAM:** At least 16 GB.
- **Storage:** Minimum 1 TB HDD or 500 GB SSD for efficient data handling and backups.

Client Specifications

- Any device capable of running modern web browsers (e.g., desktops, laptops, tablets, smartphones).
- Adequate RAM and processor to support modern operating systems and web browsers.

DOI:10.46647/ijetms.2024.v08i03.034 ISSN: 2581-4621

Network

● Reliable internet connection with minimum bandwidth of 100 Mbps for seamless access and data exchanges.

Software Requirements

Operating System

● **Server:** Ubuntu LTS or any compatible Linux distribution.
● **Client:** Any modern operating system capable of running the latest web browsers.

Web Technologies

● **Front-end:** HTML5, CSS3, Bootstrap, AJAX, jQuery.
● **Back-end:** Django framework.
● **Database:** PostgreSQL.
● **Web Server:** NGINX for load balancing and serving as a reverse proxy.

Development Tools

● **IDE:** Visual Studio Code, PyCharm, or equivalent.
● **Version Control:** Git.
● **Testing:** Frameworks like PyTest for Python/Django applications.

Security

● SSL/TLS for secure HTTP (HTTPS) communication.
● Regular security patches and updates for all software components.

Deployment and Monitoring

● Cloud hosting services with scalability options (e.g., AWS, Google Cloud, or Azure).
● Tools for continuous integration and deployment (e.g., Jenkins, Docker).
● System monitoring tools like Nagios or New Relic for performance and health monitoring.

The methodology for developing the KRA AER & LRM Webapp is based on the Agile software development framework, specifically tailored to web application development. This approach emphasises iterative development, where requirements and solutions evolve through collaboration between self-organising cross-functional teams.

Phases of Development

Requirement Gathering and Analysis

● **Objective:** Collect detailed functional and non-functional requirements from stakeholders.
● **Activities:** Conduct meetings, surveys and brainstorming sessions with project sponsors, end-users and IT staff to gather requirements.
● **Output:** A comprehensive requirements document that guides the entire development process.

System Design

● **Objective:** Define the overall system architecture and design, including data flow, user interface and necessary integrations.
● **Activities:** Creation of system blueprints using UML diagrams (use case diagrams, sequence diagrams, etc.), database schema designs and mockups of the user interface.
● **Output:** Detailed design documents and interface mockups ready for review and approval by stakeholders.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.8May - June – 2024
DOI:10.46647/ijetms.2024.v08i03.034 ISSN: 2581-4621

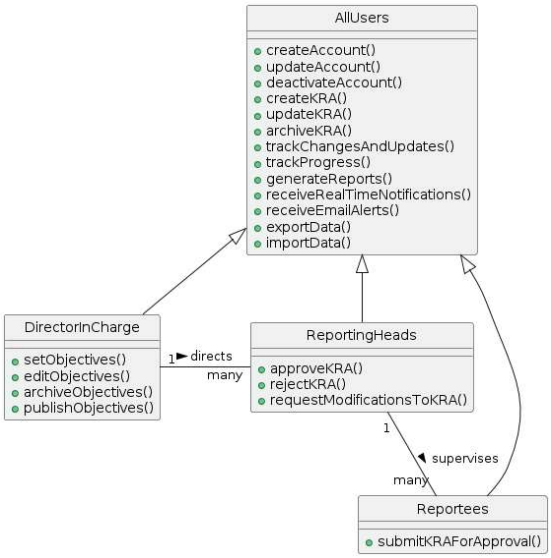

Fig. 1

Implementation and Coding

- **Objective:** Develop the system based on the approved designs using the selected technologies.
- **Activities:** Coding the application functionalities in sprints (Agile iterations), with each sprint focusing on a set of functionalities.
- **Output:** Incremental releases of the application, ready for testing and feedback.

Testing

- **Objective:** Ensure the application is bug-free and meets all requirements.
- **Unit Testing:** Test individual components for correct behavior.
- **Integration Testing:** Ensure that different parts of the application work together seamlessly.
- **System Testing:** Test the complete system's functionality.
- **User Acceptance Testing (UAT):** Conduct testing with actual users to ensure the system meets their needs and expectations.
- **Output:** A fully tested application with resolved defects and approved by users.

Deployment

- **Objective:** Successfully deploy the application in a live environment.
- **Activities:** Set up the production environment, migrate data if necessary and deploy the application. Implement backup and recovery processes.
- **Output:** The application is live and accessible to users.

Maintenance and Updates

- **Objective:** Provide ongoing support and update the application as needed.
- **Activities:** Regular monitoring of the application for issues, patching bugs and updating the system with new features as per the feedback.
- **Output:** An updated, well-maintained application that evolves according to user needs and technological advancements.

Agile Practices

- **Daily Stand-ups:** Short meetings to discuss progress, plan the day’s work and identify any issues.
- **Sprint Planning:** Regular meetings to plan the work for the next sprint.

DOI:10.46647/ijetms.2024.v08i03.034 ISSN: 2581-4621

● **Sprint Reviews:** Meetings at the end of each sprint to demonstrate the completed work to stakeholders and gather feedback.
● **Sprint Retrospectives:** Reflect on the past sprint and identify improvements for future sprints.

Tools and Technologies
● **Project Management:** Tools like Jira or Trello to track tasks, sprints and backlogs.
● **Version Control:** Git for source code management.
● **CI/CD:** Jenkins or GitLab for continuous integration and deployment.
● **Monitoring and Performance Tools:** Tools like Nagios or New Relic to ensure optimal application performance and uptime.

IMPLEMENTATION

The implementation of the KRA AER & LRM Webapp is structured around the Agile development framework, using iterative sprints to progressively build and refine the application. This approach allows for flexibility in adjusting to changes and incorporating feedback effectively.

Key Implementation Activities

Environment Setup
● Development Environment: Configure development tools, IDEs (such as PyCharm for Django), version control systems (Git) and local testing servers.
● Testing Environment: Set up a separate testing environment that mirrors the production setup to ensure accurate performance and functional testing.
● Production Environment: Establish the live environment using an Ubuntu server with NGINX for load balancing and a secure PostgreSQL database.

Database Configuration
● Implement the database schema as designed in the system design phase.
● Optimize indices and queries for performance.
● Ensure security measures are in place for data protection and integrity.

Back-end Development
● Django Setup: Configure Django project settings, including middleware, templates and URL dispatchers.
● Model Development: Define Django models according to the database schema to manage database operations.
● View Implementation: Create views to handle the business logic of the application.
● Controller Setup: Develop controllers to manage the flow between models and views.

Front-end Development
● HTML/CSS: Markup and style the web pages using Bootstrap for responsiveness.
● JavaScript/JQuery: Enhance user interfaces with dynamic content and client-side scripting.
● AJAX: Implement AJAX for asynchronous data fetching, improving the responsiveness of the application.

API Integration
● Develop RESTful APIs for critical application functionalities to ensure seamless and secure data exchange between the front-end and back-end components.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.8May - June – 2024
DOI:10.46647/ijetms.2024.v08i03.034 ISSN: 2581-4621

- Implement authentication and authorization for API access.

Testing and Debugging

- Conduct unit testing for individual components using frameworks like PyTest.
- Perform integration testing to ensure all parts of the application work together seamlessly.
- Execute system testing to validate the complete functionality of the application.
- Carry out user acceptance testing with actual users to confirm the system meets their needs and expectations.

Deployment

- Use Docker containers or similar technologies for deployment to ensure consistency across different environments.
- Implement continuous integration and deployment pipelines using tools like Jenkins to automate the build, test and deployment processes.
- Deploy the application to the production environment, ensuring all components are correctly configured and operational.

Post-Deployment

- Monitor the application's performance and troubleshoot any issues that arise.
- Gather user feedback and prepare for iterative updates and maintenance.

RESULTS AND DISCUSSION

The implementation of the generalized KRA management tool across various institutions demonstrated:

- Enhanced management and visibility of research projects.
- Increased collaboration within and across departments.
- Significant time savings and better resource allocation.
- Improved student and faculty satisfaction due to more streamlined and engaging learning processes.

In conclusion, the application of a generalized educational technology for managing KRAs in project-based learning settings illustrates profound benefits in enhancing educational quality and efficiency. This paper underscores the importance of adaptable and scalable technology solutions in higher education, advocating for wider adoption to better prepare students for professional and academic challenges.

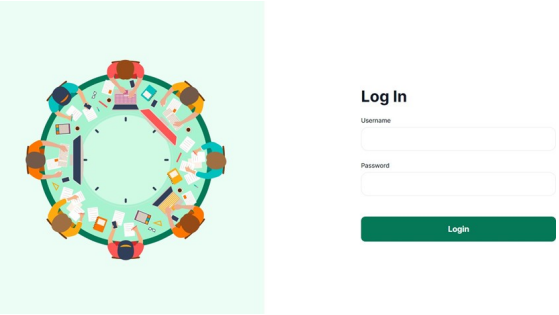

Fig. 2

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.8May - June – 2024
DOI:10.46647/ijetms.2024.v08i03.034 ISSN: 2581-4621

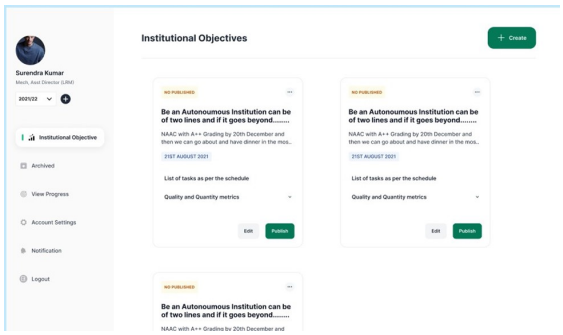

Fig. 3

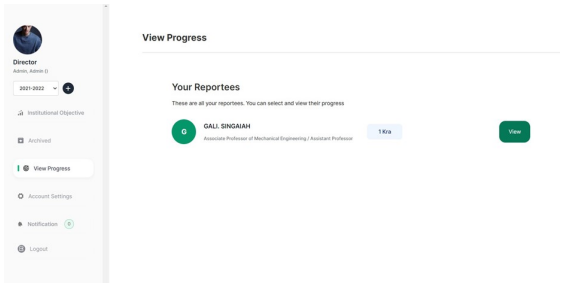

Fig. 4

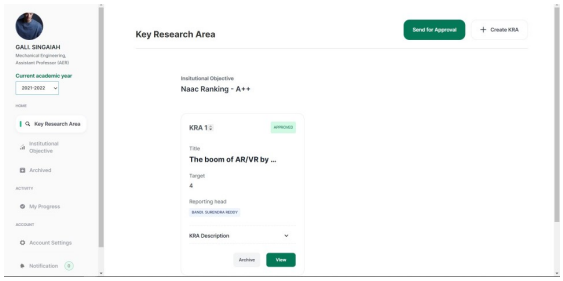

Fig. 5

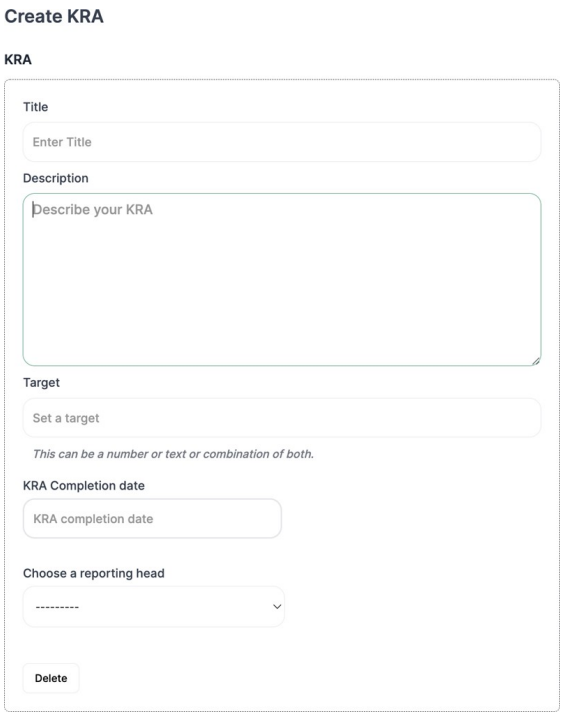

Fig. 6

DOI:10.46647/ijetms.2024.v08i03.034 ISSN: 2581-4621

CONCLUSION

This project represents a strategic initiative aimed at enhancing the efficiency, effectiveness and responsiveness of educational administration processes through the implementation of the "KRA, AER and LRM Web Application." As educational institutions continue to expand and evolve, the need for a robust, scalable and secure system to manage Key Result Areas (KRAs), Annual Evaluation Reports (AERs) and Learning and Resource Metrics (LRMs) has become imperative.

With the successful implementation of this web application, the institution is well-positioned to further leverage technological advancements. Future enhancements can include the integration of artificial intelligence to predict educational trends and outcomes, or the expansion of mobile capabilities to increase accessibility for all users.

REFERENCES

[1] Stuart J. Barnes and Andrew D. Pressey. Technology and educational innovations in higher education: Harnessing the potential of digital platforms. Journal of Educational Administration, 2019.
[2] Jakob Nielsen. User experience requirements for e-learning environments.Computers & Education, 2018.
[3] Ronald K. Larson. Building effective educational applications using api architectures. Journal of Educational Technology Systems, 2020.
[4] Paul Y. Kim and Thomas Keller. Security practices in educational software systems.Software, Practice & Experience, 2017.
[5] Derek Thompson. Scalability strategies for web-based educational tools. The Internet and Higher Education, 2018.
[6] Michael Hartley and John Benington. Performance indicators and educational management: A practical guide for effective usage. Educational Management Administration & Leadership, 2019.
[7] Elizabeth Sanchez and Michael Peters. Automation in education reporting: Implications for schools. Journal of Educational Administration, 2021.
[8] David Anderson and John Kanban. Agile management for software engineering: Applying the theory of constraints for business results. Educational Technology Research and Development, 2017.
[9] Poonam Gupta and Arumugam Seetharaman. The role of cloud computing in higher education. International Journal of Information Management, 2019.
[10] Carla Morales and Robert Baker. Sustainable educational technology: Perspectives on long-term project viability. Computers & Education, 2020.