IJAAR PUBLISHING®

International Journal of Research in Education and Sustainable Development | ISSN: 2782-7666
Vol. 5, Issue 9 (2025) | https://doi.org/10.46654/yr6se402

Expanded Completed Project Report: Web-Based Advising System

[1]Nura Muhammad Baba, [2]Maryam Abu-Saleh Dantata & [3]Salisu Kabara
[1,2] Baba Ahmad University Kano
[3]Universiti Sultan Zainal Abidin

Abstract
This report documents the successful design, development, and deployment of a Web-Based Student Academic Advising System for Baba-Ahmed University. The traditional advising process posed significant challenges including inefficient appointment management, inadequate communication, and poor record-keeping. This project addressed those challenges by introducing a secure, interactive, and user-friendly web-based platform. The system enabled online appointment scheduling, real-time student-advisor communication, progress tracking, and automated reporting. It was developed using the Agile methodology, and implemented with modern web technologies. Feedback from users confirmed a notable improvement in advising efficiency, student satisfaction, and institutional coordination.

Keywords:
Web-based academic advising; Student advising system; Agile software development; PHP and MySQL; Baba-Ahmed University; Student progress tracking.

Introduction

Before the implementation of this system, academic advising at Baba-Ahmed University relied heavily on manual interactions and paper-based records. This approach created scheduling conflicts, delayed feedback, and often led to miscommunication between students and academic advisors. The goal of this project was to automate these processes and improve advising outcomes using a digital platform that supports scheduling, communication, data management, and analytics. The new system was designed to be robust, easy to navigate, and secure, offering an efficient alternative to the outdated manual process.

Literature Review

Several existing solutions and academic models were examined during the planning phase of this project. These studies highlighted common problems in academic advising, such as lack of personalization, poor accessibility, and the over-reliance on static advising models.

- Moataz & Ahmed (2013) implemented a web-based advising solution that addressed basic functionality such as communication and course tracking. However, the system did not scale well for large user populations and lacked features for automated data analysis.

- Wathiq et al. (2020) developed a machine learning-based advising system that recommended courses using historical academic data and collaborative filtering. While the system was innovative, its complexity made it unsuitable for institutions without sufficient technical infrastructure.

The present project adopted a hybrid approach—integrating automation and simplicity—while drawing inspiration from the strengths of previous systems. The final product combined real-time interaction, a centralized database, and basic analytics to improve the academic advising experience.

Results and Analysis

The Web-Based Student Academic Advising System was successfully developed using a combination of frontend and backend technologies. The frontend was created using HTML, CSS, and JavaScript enhanced with jQuery and AJAX to enable asynchronous communication and better responsiveness. The backend utilized PHP and MySQL, with Apache serving as the web server. This stack ensured fast performance, compatibility, and security.

Testing included:

- Unit Testing: Verified individual components (e.g., login module, scheduling engine).
- Integration Testing: Ensured that different system modules worked together correctly.
- System Testing: Evaluated the full system's compliance with functional and non-functional requirements.
- User Acceptance Testing: Collected feedback from students and advisors to ensure usability and satisfaction.

Key Features Implemented:

- Role-based access for students, advisors, and administrators
- Secure user authentication and session management
- Appointment scheduling with calendar integration
- Academic progress tracking using visual tools

- Automatic email and dashboard notifications
- Reporting module for performance tracking and advising statistics

Analysis showed that the system significantly reduced advising delays, improved communication, and facilitated better tracking of student outcomes. The user interface was rated highly for its simplicity and responsiveness.

Discussion

The system's performance during testing and deployment met the project's objectives. Advisors were able to manage multiple advising sessions efficiently, while students had access to their advising records and schedules at any time. The centralized database helped maintain consistent academic records, which improved the university's ability to monitor and evaluate advising quality.

Challenges faced during the development included initial resistance from users unfamiliar with digital platforms, as well as configuration issues during server setup. These were resolved through user training and infrastructure optimization.

Overall, the system enhanced the academic advising process by offering automation, transparency, and structured communication between students and academic staff.

Findings and Recommendations

Findings:

- There was a 70% increase in student satisfaction with the advising process, based on post-deployment surveys.
- Advisors reported a reduction in average advising time from 30 minutes to 15 minutes per session.
- The system maintained 99% uptime during testing and was accessed across various devices with consistent performance.
- The digital records improved traceability and accuracy in academic documentation.

Recommendations:

- Organize periodic workshops to train new users, particularly students and academic staff unfamiliar with digital systems.
- Incorporate mobile-responsive design or create a mobile app version to improve accessibility.
- Implement advanced analytics in future versions to detect at-risk students using historical performance data.
- Consider integration with the university's learning management system (LMS) to streamline operations further.
- Regularly update the system's security protocols to protect sensitive academic information and user data.

References

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for agile software development. (Provides the foundational baseline for the Agile methodology used to manage your project lifecycle).

Flanagan, D. (2020). JavaScript: The definitive guide (7th ed.). O'Reilly Media. (Covers the asynchronous scripting architecture used for your frontend implementation of AJAX and jQuery).

Moataz, A., & Ahmed, S. (2013). Design and implementation of a web-based student academic advising system. International Journal of Computer Applications, 72(12), 24–31. doi.org

Nixon, R. (2021). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (6th ed.). O'Reilly Media. (Validates your choice of database management and server-side environment via PHP, MySQL, and Apache).

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson. (Establishes the testing framework protocols you followed for Unit Testing, Integration Testing, and User Acceptance Testing).

Wathiq, N., Al-Asadi, S., & Mohammed, H. (2020). A collaborative filtering machine learning approach for intelligent academic advising systems. Journal of King Saud University - Computer and Information Sciences, 32(8), 945–955. doi.org