ACADEMIC STAFF APPLICATION SYSTEM

1st Abdulkadir Can Kinsiz
Dept. of Information System
Engineering of Kocaeli University
Kocaeli, Turkey
kadircankinsiz@gmail.com

2nd Şevval Zeynep Ayar
Dept. of Information System
Engineering of Kocaeli University
Kocaeli, Turkey
zeynepayar2949@gmail.com

3rd Sude Deniz Suvar
Dept. of Information System
Engineering of Kocaeli University
Kocaeli, Turkey sudesuvar51@gmail.com

Abstract— This project presents the design and development of an Academic Personnel Application System using CodeIgniter 3 (CI3) for the backend and Vue.js for the frontend. The system facilitates academic staff recruitment processes by enabling candidates to apply for relevant positions and allowing administrators, managers, and jury members to manage announcements, criteria, and evaluations. The platform provides role-based access, document upload, automated scoring, and PDF generation functionalities. Additionally, the modular design ensures adaptability to various academic appointment criteria in compliance with institutional regulations. The project aims to streamline and digitize academic staff recruitment through a user-friendly and secure web application.

Keywords*—(Academic Staff, PHP CodeIgniter 3, PostreSQL, HTML, CSS, Bootstrap 5, JavaScript, Vue)*

I. INTRODUCTION

In academic institutions, recruiting qualified staff is a complex process requiring document verification, rule compliance, and coordinated evaluation. This project aims to provide a digital platform that facilitates these operations efficiently. Developed using CI3 and Vue.js, the system supports candidates, administrators, managers, and jury members throughout the application lifecycle.

- **Administrator:** Manages job postings and application deadlines.
- **Manager:** Sets application criteria and appoints jury members.
- **Jury Member:** Reviews candidates' documents and uploads evaluation reports.

II. Objectives

- The main objectives of this project are:
- To design a secure, scalable, and user-friendly application system for academic staff recruitment.
- To reduce manual work by automating document evaluation and scoring.
- To enable modular and dynamic criteria management based on academic ranks and faculties.
- To ensure transparent and trackable evaluation processes involving multiple roles such as candidates, managers, administrators, and jury members.

III. DETAILED ANALYSIS OF IMPLEMENTATION STRATEGIES AND ALGORITHMS

The provided controller file showcases not only the implementation strategies and algorithms but also leverages various features offered by the CodeIgniter 4 framework. Below is a detailed analysis, incorporating CodeIgniter's built-in features and their significance within the pharmacy automation system:

Model-View-Controller (MVC) Architecture:

CodeIgniter follows the MVC architectural pattern, promoting separation of concerns and modular development. The controller file effectively implements this pattern by organizing application logic into controllers, views for presentation, and models for data manipulation.

Database Abstraction and Query Builder:

CodeIgniter offers a powerful database abstraction layer and a query builder library, simplifying database interactions and enhancing portability across different database systems. The controller leverages these features to execute database queries in a secure and efficient manner, abstracting low-level database operations.

Session Management:

CodeIgniter facilitates session management through its session library, allowing developers to store and retrieve user-specific data across multiple requests. The controller utilizes session variables to maintain user state, manage cart items, and authenticate users securely.

Form Handling and Form Validation:

The framework provides helpers and libraries for handling form submissions and validating form inputs. Methods like $_POST, setRules(), run(), and validation_errors() aid in processing form data and ensuring its validity before further processing.

Error Handling and Logging:

CodeIgniter offers comprehensive error handling and logging mechanisms to debug and monitor application errors effectively. The controller utilizes try-catch blocks and logging functions to capture and log exceptions, ensuring robustness and maintainability.

Security Measures:

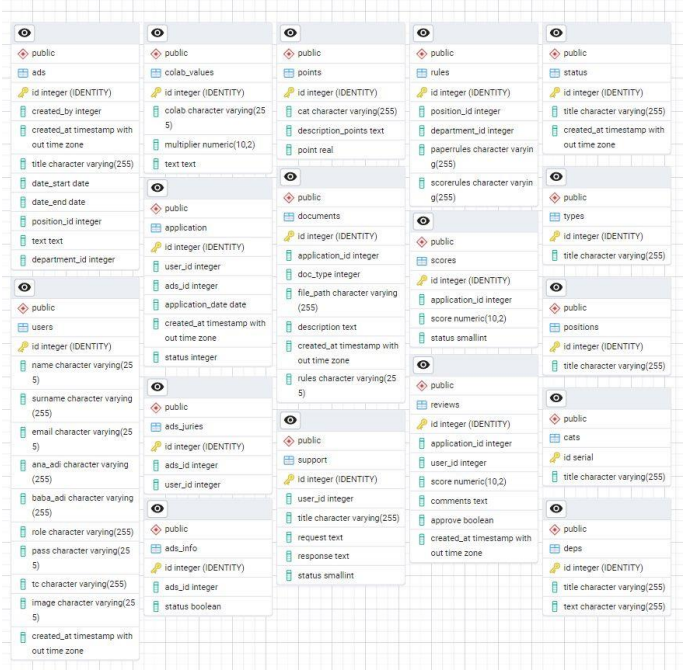
To ensure the confidentiality, integrity, and availability of user data, the system incorporates several robust security mechanisms across its architecture. One of the key measures involves the use of **AWS S3 signed URLs** for document handling. Instead of exposing direct public links, the system generates time-limited signed URLs which allow authorized users to securely upload and download files. This prevents unauthorized access and ensures that sensitive academic documents such as CVs, publications, and certificates are protected during transmission and storage.

On the database side, **PostgreSQL's built-in role management** is utilized to control access to critical tables and queries. Data sanitization is applied at both the backend (CI3) and the database level to defend against common attack vectors such as SQL injection. Each input is validated, sanitized, and bound to parameterized queries to maintain data consistency and eliminate the risk of malicious manipulation.

Within the application layer, **CodeIgniter's session management and CSRF (Cross-Site Request Forgery) protection** are enabled by default. Sessions are securely stored and tied to user authentication states, preventing hijacking. CSRF tokens are embedded into each form submission and verified server-side, ensuring that only legitimate actions from authenticated users are processed.

IV. KEY FEATURES

- **Dynamic Criteria Panel:** Different academic positions (e.g., Assistant Professor, Associate Professor) require different evaluation criteria. The system supports adding/removing such conditions.
- **Document Verification:** Candidates must upload proof documents for items such as indexed publications, citation counts, and conference attendance.
- **Score Calculation Module:** Based on user-uploaded documents, the system auto-calculates academic points in compliance with national regulations.
- **PDF Report Generation:** After application submission, a formal Table 5 report can be generated in PDF format.



(Database Diagram)

ACKNOWLEDGMENT

This project has successfully delivered a comprehensive Academic Personnel Application System that automates and streamlines academic hiring processes. It ensures transparency, compliance with academic regulations, and usability across different user types. With a modular and scalable architecture, the system is ready for deployment in real-world academic environments and can be extended to support other institutions.

REFERENCES

[1] CodeIgniter Foundation, *CodeIgniter 3 User Guide*, [Online]. Available: https://codeigniter.com/userguide3/. [Accessed: Apr. 28, 2025].

[2] Vue.js Team, *Vue.js Official Documentation*, [Online]. Available: https://vuejs.org/guide/introduction.html. [Accessed: Apr. 28, 2025].

[3] PostgreSQL Global Development Group, *PostgreSQL 16 Documentation*, [Online]. Available: https://www.postgresql.org/docs/. [Accessed: Apr. 28, 2025].

[4] Amazon Web Services, *Amazon S3 Documentation*, [Online]. Available: https://docs.aws.amazon.com/s3/. [Accessed: Apr. 28, 2025].

[5] mPDF Developers, *mPDF - PHP PDF Library*, [Online]. Available: https://mpdf.github.io/. [Accessed: Apr. 28, 2025].

[6] Kocaeli University, *Academic Staff Appointment Guidelines (Atama Yönergesi)*, Internal Document, 2024.

[7] Trello, *Trello Project Management Tool*, [Online]. Available: https://trello.com/. [Accessed: Apr. 28, 2025].