Ph.D. Management Portal

Ayush Kumar Satish (2020133)

Atyant Sony (2020039)

Yuvraj Singh (2020270)

Introduction

In the realm of academic management, it is imperative to efficiently oversee the progress of PhD students within educational institutions. The current manual processes of supervising PhD candidates, which encompass tasks like thesis assessment, comprehensive exams, annual evaluations, contingency points management, token management, exam invigilation allocation, and stipend management, are not only labor-intensive but also susceptible to errors. In response to these challenges, we are in the process of developing a comprehensive software solution that will streamline these administrative procedures and seamlessly maintain academic records.

Purpose

The principal aim of our software system is to modernize and simplify the management of academic activities related to PhD students. This is done with the intention of enhancing overall efficiency and accuracy throughout the process. The system is designed to automate various facets of PhD program administration, including the tracking of attendance, stipend disbursements, annual evaluations, comprehensive examinations, exam invigilation allocation, contingency points management, token management and the evaluation of PhD theses. This approach will alleviate administrative burdens and provide instant access to critical data, empowering academic institutions to make well-informed decisions.

Types of Users

The system will cater to three distinct user roles, each assigned specific responsibilities and privileges:

- Ph.D. Administrator:
    - The administrator holds overarching control over the PhD program and its management.
    - Responsibilities include overseeing all system functionalities, user management, and ensuring the system's overall integrity.

    - Monitoring and managing all aspects of the Ph.D. program, including attendance tracking, stipend disbursements, evaluations, contingency claims, exam invigilation allocations, and thesis assessments.
- Professor:
    - Professors are responsible for supervising PhD students under their guidance.
    - Access to relevant data pertaining to the students they supervise, such as attendance records, evaluations, and thesis assessments.
    - The ability to validate contingency claim requests submitted by their assigned PhD students.
- Student
    - PhD students have access to personalized information and functionalities related to their academic journey.
    - Check the status of contingency claim requests, including validations by their respective professors.
    - Raise new contingency claim requests and monitor the balance of contingency points.

Each user type will experience a tailored interface providing access only to the functionalities pertinent to their role, ensuring a streamlined and user-friendly experience for administrators, professors, and students alike. The authentication and authorization mechanisms will enforce the appropriate access levels for each user type, maintaining the confidentiality and integrity of sensitive academic data.

User Functionality

User Type	Functionality
Ph.D. Administrator	● Management of Student Records ● Stipend Processing ● Annual Reviews ● Comprehensive Examination Records ● Thesis Assessment & Evaluation ● Exam Invigilation Allocation ● Contingency Claims Validation and Logging
Professor	● Validate Contingency Points Claims of Students working under the Professor
Student	● Raise New Contingency Claims ● Track the status of Contingency Claims ● Check their Contingency Points Balance

Tech Stack

The technological foundation for the Ph.D. Management Portal project is carefully selected to ensure robust performance, scalability, and a modern user interface. The backend is developed using Django, a high-level Python web framework renowned for its versatility and efficiency. The Django Object-Relational Mapping (ORM) system facilitates seamless interaction with the database, enhancing data management and retrieval. On the front end, we leverage ReactJS to create a dynamic and responsive user interface. ReactJS, known for its component-based architecture, provides an interactive and efficient user experience. Complementing ReactJs, TailwindCSS is employed for frontend styling, offering a utility-first approach that enhances the flexibility and maintainability of the project's design. This tech stack amalgamates the robust backend capabilities of Django with the interactive and visually appealing frontend powered by ReactJS and TailwindCSS, ensuring a comprehensive and modern solution for the efficient management of academic activities within the Ph.D. program.

Functional Requirements Overview

User	Use Case Name	Description	Priority
All	User Authentication and Access Control	Users can securely log in to the portal	High
Admin	Student Profile Management	Admin can view, edit, and add new student profiles	High
Admin	Stipend Management	Admin can review and approve stipend disbursements based on the professor's review	High
Admin	Academic Reviews Overview	Admin can view, edit, and add new performance reviews based on professors's review	High
Admin	Comprehensive Exam and Thesis Evaluation Database	Admin can maintain information related to comprehensive exams and thesis evaluations in the portal	High
Admin	Exam Invigilation Allocation	Admin can allocate exam invigilation to students efficiently using an automated algorithm	High
Admin	Validate Contingency Claims	Admin can validate contingency claims of all students	Mid
Professor	Validate Contingency Claims	Professor can validate contingency claims of students working under him	Mid
Student	Claim Contingency Points	Students can raise new contingency claims in the portal	Mid
Student	Track Contingency Claim Process	Students can track the status of claims raised by them in the portal	Low
Student	Check Contingency Points balance	Students can check their contingency points balance	Low

Non-Functional Requirements

User Interface Requirements

- **User-Friendly & Responsive Interface:** The portal shall have an intuitive, user-friendly and responsive interface to ensure ease of use on every type of device.
- **Cross-Browser Compatibility:** The portal shall be compatible with major web browsers (e.g., Chrome, Firefox, Safari, Edge) to provide a consistent user experience.

Performance Requirements

- **Response Time:** The portal shall have a maximum response time of 3 seconds for all user interactions to ensure a responsive user experience.
- **Scalability:** The system shall be designed to handle a growing number of data records without significant performance degradation.

Design Constraints

- **Security:** The system shall implement security measures, including encryption, authentication, and authorization, to protect user data and maintain data confidentiality.
- **Data Backup and Recovery:** Regular data backups and a disaster recovery plan shall be implemented to ensure data integrity and availability.
- **Data Privacy:** The system shall comply with data privacy regulations to protect user data privacy and confidentiality.
- **Documentation:** Comprehensive technical documentation shall be maintained to facilitate future maintenance and development efforts.
- **Testing:** Rigorous testing procedures, including unit testing, integration testing, and user acceptance testing, shall be conducted to ensure software quality and reliability.
- **Support and Maintenance:** Ongoing support and maintenance shall be provided to address issues, apply updates, and ensure the portal's continued functionality and security.

These non-functional requirements are essential for ensuring the usability, performance, security, and maintainability of the Ph.D. Management Portal.