International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 03 | Mar 2024 www.irjet.net p-ISSN: 2395-0072

"College Student – Admin Portal"

Dattashri Balbude[1], Runank Patil[2], Shrihari Dhore[3]
Prof. Prachi Bhure

Student of, dept of Computer Engineering, Swaminarayan Siddhanta Institute of Technology, Kalmeshwar Nagpur, Maharashtra, India.
H.O.D., Dept of Computer Engineering Swaminarayan Siddhanta Institute of Technology, Kalmeshwar Nagpur, Maharashtra, India.

Abstract - *Our project is focused on revolutionizing the college experience for students and optimizing administrative workflows. We are developing an extensive web application that comprises a React-admin-dashboard frontend and a backend supported by either MongoDB or PostgreSQL, building upon our previous work. Central to our project is the creation of a student dashboard that caters to the needs of both students and administrators. This dashboard will facilitate secure user authentication and grant access to vital information such as departmental circulars and college announcements, crucial for students' academic journeys. The student dashboard will act as a central hub for accessing exam results, internal assignment deadlines, scholarship details, exam schedules, an online library catalog, and early notifications about extracurricular activities. This intuitive interface ensures students remain well-informed about important dates, requirements, and opportunities, thereby enhancing their overall college experience. Following the Model-View-Controller (MVC) architectural pattern, our project prioritizes creating visually appealing and interactive user interfaces using React. Leveraging Java as the technology stack, our web application is robust and feature-rich, promoting increased student engagement and enabling effective documentation of daily activities. Ultimately, our goal is to bridge the communication gap between students and college administration through a feature-rich student dashboard within an integrated web application framework. This initiative empowers students with necessary academic information and encourages active participation in college life, while also simplifying administrative tasks for the benefit of both students and staff.*

Key Words: **React.js, Portal, java, PostgreSQL, MongoDB, mySQL**

1.INTRODUCTION

This In today's dynamic educational environment, technology plays a pivotal role in revolutionizing the college experience and optimizing administrative operations. Our project represents a significant step forward, leveraging previous groundwork to develop a comprehensive web application poised to transform student-institution interactions. With a frontend React-admin-dashboard and a robust backend supported by MongoDB or PostgreSQL, our initiative aims to redefine the college experience. At its heart, our project introduces a groundbreaking student dashboard

tailored for students and administrators. This dashboard streamlines user authentication, ensuring secure access to essential resources like departmental circulars, pivotal for students' academic journeys.The student dashboard acts as a central hub, providing critical information such as exam results, assignment deadlines, scholarship details, and extracurricular updates. Through an intuitive interface, it empowers students with timely information, enriching their college experience. Following the Model-View-Controller (MVC) pattern, we prioritize creating an aesthetically pleasing and interactive interface using React. Our Java-based technology stack ensures reliability across diverse functionalities, promoting student engagement and facilitating efficient administrative processes. In essence, our project bridges the gap between students and administration, delivering a feature-rich student dashboard within a cohesive web application framework. This endeavor fosters academic success, active engagement, and streamlined operations, ushering in a new era of educational synergy and innovation.

1.1 Core Features

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

1.2 User Interface Design

Model-View-Controller (MVC) Architecture:

Our project follows the Model-View-Controller (MVC) architectural pattern, emphasizing a clear separation of concerns for modularity and maintainability.

The MVC structure ensures scalability and flexibility, with implementation as follows:

Model: Represents data and business logic, including student/admin data, circulars, and exam results, managed in MongoDB or PostgreSQL for data integrity.

View: Encompasses user interface elements built with React for visually appealing and interactive components, ensuring a user-friendly experience.

Controller: Acts as an intermediary handling user interactions, requests processing, and real-time updates via React, ensuring secure access and functionality.

React GUI Components:

Utilizing React, our project creates modular and reusable GUI components for an engaging user interface, including:

User Authentication: Robust system for secure dashboard access.

Dashboard Widgets: Display vital information like exam results and deadlines, designed to be visually appealing and informative.

Responsive Design: Adaptable to various devices for a seamless experience.

Intuitive Navigation: User-friendly menus for easy information access.

User Experience:

Our focus is on delivering an exceptional user experience through:

Intuitive Design: Easy navigation and information accessibility.

Interactivity: Interactive elements for effortless actions like checking results or browsing catalog.

Performance: Optimized for quick loading and responsive interactions.

Security: Strong measures to safeguard user data.

Accessibility: Designed to be inclusive for all users, ensuring usability for everyone.

2. Technology Stack Details

To implement the College Experience Enhancement project in Java, follow these steps:

Initialize Web Application Components:

Create a ReactAdminDashboard class to define the front-end component using React.

Implement methods and functionality within ReactAdminDashboard to create a rich GUI.

Database Interaction:

Define a Database interface with methods for interacting with the database.

Implement MongoDB and PostgreSQL classes that implement the Database interface.

In the chooseDatabase() method, implement logic to select either MongoDB or PostgreSQL based on requirements.

Student Dashboard:

Create a StudentDashboard class that manages the student dashboard.

Add private fields for the front-end (ReactAdminDashboard) and backend (Database) components.

Implement a constructor in StudentDashboard to initialize these components.

Implement methods within StudentDashboard for:

User authentication logic.

Retrieving and displaying departmental and college circulars.

Displaying exam results, deadlines, scholarship details, exam dates, library catalog, and extracurricular activities.

Main Class (CollegeExperienceEnhancement):

In the main method, initialize the web application components:

Create an instance of ReactAdminDashboard.

Use the chooseDatabase() method to select the database backend (MongoDB or PostgreSQL).

Create an instance of StudentDashboard with the initialized front-end and backend components.

Demonstrate the core features of the project by calling the relevant methods in StudentDashboard:

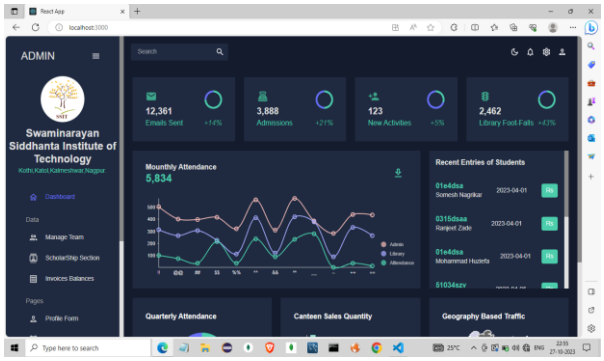
authenticateUser(),displayCollegeCirculars(), displayExamResults(),displayDeadlines(), displayScholarshipDetails(),displayExamDates(), displayLibraryCatalog(),and displayExtracurricularActivities().

Closing the Application:

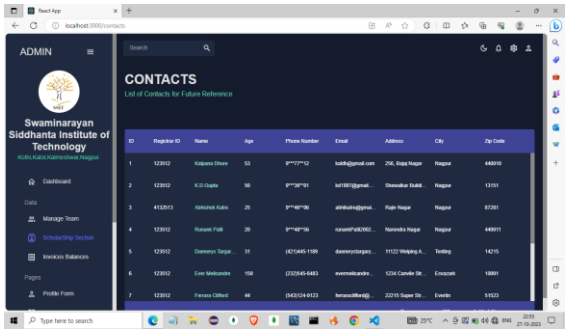
After demonstrating the core features, display a closing message to thank the user for using the College Experience Enhancement application.

This implementation structure allows for a modular and organized Java development approach, ensuring clear separation of concerns between different components of the project. It also provides flexibility in choosing and implementing the database backend and front-end GUI functionality.

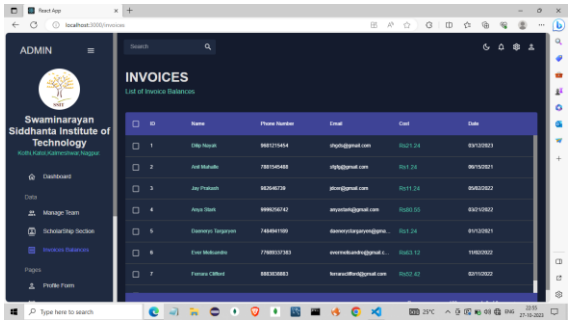
This is a simplified Java program to represent your project based on the provided abstract. You'll need to implement the actual functionalities, integrate with the chosen database, and develop the React Admin Dashboard as per your project's requirements. Additionally, you can further modularize and structure your code according to your project's needs and best coding practices.



Screenshot 1: Dashboard



Screenshot 2: Scholarship Section



Screenshot 3: Accounts Section

3. CONCLUSIONS

Our project, "Bridging the Gap," represents a significant step forward in enhancing the college experience for students and optimizing administrative processes. Leveraging the groundwork laid in the previous year, we are currently developing a comprehensive web application that integrates a front-end React-admin-dashboard with a backend powered by either MongoDB or PostgreSQL.

Central to our project is the creation of a student dashboard that caters to the diverse needs of both students and administrators. This dynamic dashboard simplifies user authentication, ensuring secure access to critical information such as departmental circulars, which profoundly impact students' academic journeys.

The student dashboard acts as a centralized hub, offering access to essential data like exam results, assignment deadlines, scholarship information, exam schedules, an online library catalog, and updates on extracurricular activities. Through an intuitive interface, students can effortlessly stay updated on important dates, requirements, and opportunities, enriching their overall college experience. Following the Model-View-Controller (MVC) architectural pattern, our project emphasizes a visually appealing and interactive user interface crafted using React. This design approach enhances user engagement and ensures a responsive experience.

Our choice of Java as the technology stack for development guarantees reliability and support for various functionalities within the web application. It encourages greater student involvement by facilitating the documentation and review of daily activities that might otherwise be overlooked.

In essence, "Bridging the Gap" serves as a vital link between students and college administration, providing a feature-rich student dashboard within a unified web application framework. This initiative empowers students with essential information for academic success and active participation in college life, while also streamlining administrative processes for the benefit of all stakeholders

REFERENCES

[1] HTTP: The Definitive Guide,
By – David Gourley, Brian Totty, Marjorie sayer.

[2] Programming PHP – ,
By – Kevin Tatroe, Peter Maclntrye & Rasmus Lerdorf
"Foreword By: Michael Bourqure".

[3] Node.js Web Development,
By- David Herron.

[4] Node.js Design Patterns – Third Edition,
By- Luciano Mammino and Mario Casciaro.

[5] JavaScript: The Definitive Guide,
By- David Flanagan.

[6] Advanced Java Programming,
By- Uttam Roy.

[7] Core Java for the Impatient,
By- Cay S.Horstmann.

[8] MongoDB: The Definitive Guide: Powerful and Scalable Data Storage,
By- Shannon Bradshaw, Eoin Brazil, Kristina Chodorow.

[9] Mastering PostgreSQL 15: Advanced techniques to build and manage scalable, reliable, and fault-tolerant databse applications, 5th Edition,
By- Hans-Jurgen Schnig

[10] JavaScript: The Definitive Guide,
By- David Flanagan.