

CAMPUS RECRUITMENT GATEWAY

¹Ms. A. Anitha, ²Mani Charan Reddy Gade, ³Vijay Kanth Reddy Chada, ⁴Uttam Kumar Sevarthy

¹Assistant Professor, Department of Information Technology, Anurag University, India.

^{2,3,4}B.Tech Students, Department of Information Technology, Anurag University, India.

ABSTRACT:

Traditional campus recruitment processes are plagued by inefficiencies, manual workflows, and limited accessibility, resulting in challenges such as the mismanagement of student data, delays in job postings, and ineffective communication between students, faculty, and recruiters. These limitations hinder the ability of institutions to track placement trends, provide career guidance, and conduct successful recruitment drives. This project seeks to address these challenges by developing a Campus Recruitment Gateway using the MERN stack, offering a centralized, user-friendly platform equipped with advanced filtering, role-based access control, automated notifications, real-time updates, and analytics. By streamlining placement activities and fostering seamless collaboration, the system aims to maximize placement outcomes, reduce manual efforts, and enhance the overall recruitment experience.

1. INTRODUCTION

The **Campus Recruitment Gateway** is a state-of-the-art, web-based platform designed to revolutionize the campus recruitment process for college placement departments. Built on the **MERN stack** (MongoDB, Express.js, React.js, Node.js), this application serves as a centralized solution for managing student data, posting job opportunities, and facilitating seamless communication between students and faculty. Unlike traditional recruitment systems, the Campus Recruitment Gateway is designed to exclude direct recruiter access, ensuring that the

placement department retains full control over the recruitment process. This approach safeguards student data, maintains institutional integrity, and ensures that all recruitment activities are conducted in a structured and secure manner.

The platform addresses the inefficiencies of traditional recruitment methods, such as manual data management, lack of transparency, and communication gaps, by providing a digital, automated, and user-friendly solution. With features like automated notifications, event management tools, and analytics, the Campus Recruitment Gateway empowers placement departments to conduct successful recruitment drives while minimizing administrative burdens and maximizing placement outcomes.

2. REQUIREMENTS

The Campus Recruitment Gateway is a web-based platform designed to streamline the campus recruitment process by connecting students, colleges, and companies on a single platform. The system aims to provide a centralized solution for managing job postings, applications, and recruitment activities efficiently. It targets three primary user groups: students, college/university administrators, and system administrators. The platform will be accessible via desktop and mobile devices, ensuring ease of use for all users. It should integrate with third-party tools like Google Calendar for scheduling and LinkedIn for profile verification. Additionally, the system must comply with data protection regulations such as GDPR and FERPA to ensure the security and privacy of user

data.

2.1. Functional Requirements

Functional requirements define what the system should do.

1. Student Module

1. **User Registration and Login:** Students can register and log in using their credentials.
2. **Profile Management:** Students can create, update, and view their profiles (personal details, academic records, skills, and resumes).
3. **Job Search:** Students can search for job postings based on filters like job title, location, and skills.
4. **Job Application:** Students can apply for jobs and upload required documents.
5. **Application Tracking:** Students can track the status of their applications (e.g., Applied, Shortlisted, Rejected).
6. **Notifications:** Students receive notifications for new job postings, application updates, and interview schedules.

2. Faculty Module

1. **Filtration:** Filter students by CGPA, skills, placement status, and department; filter placements by company, role, and eligibility.
2. **Analysis:** Analyze student performance (CGPA, placement rates) and placement trends using visualizations and reports.

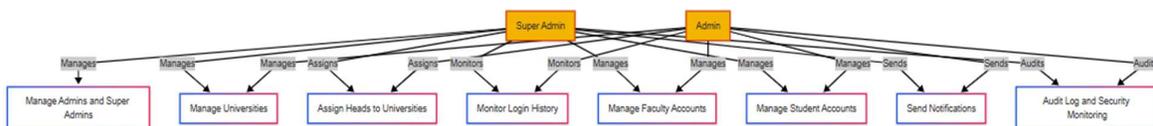
3. **Personalized Dashboard:** Display student counts, active placements, deadlines, and quick actions (add companies, assign placements).
4. **Notifications:** Real-time alerts for profile updates, new placements, and deadlines; customizable notification settings.
5. **System Activities:** Change password, deactivate/delete account, and update profile securely.
6. **Report Generation:** Admins can generate reports on recruitment activities and student placements.

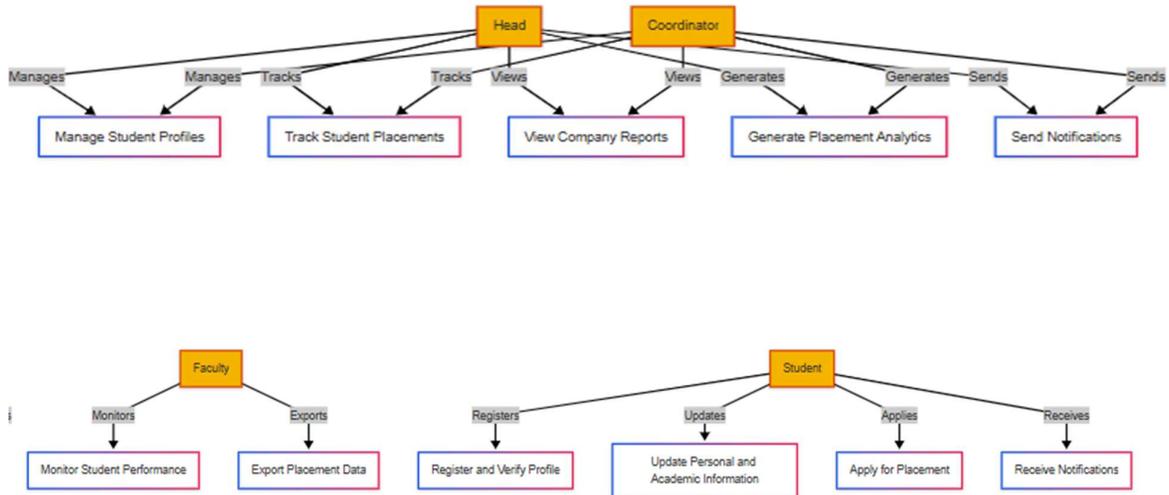
3. Admin Module

1. **College Management:** Add/remove colleges, manage college details (name, address, courses), and track registered colleges.
2. **Faculty Management:** Add/remove faculty, assign faculty to colleges, and manage faculty profiles (name, email, department).
3. **Student Management:** Add/remove students, manage student profiles (CGPA, skills, resume), and track student enrollment and placements.
4. **System Analytics:** Admin can view system usage statistics (e.g., number of job postings, applications, and registered users).
5. **System Configuration:** Admin can configure system settings, such as notification templates and user access levels.

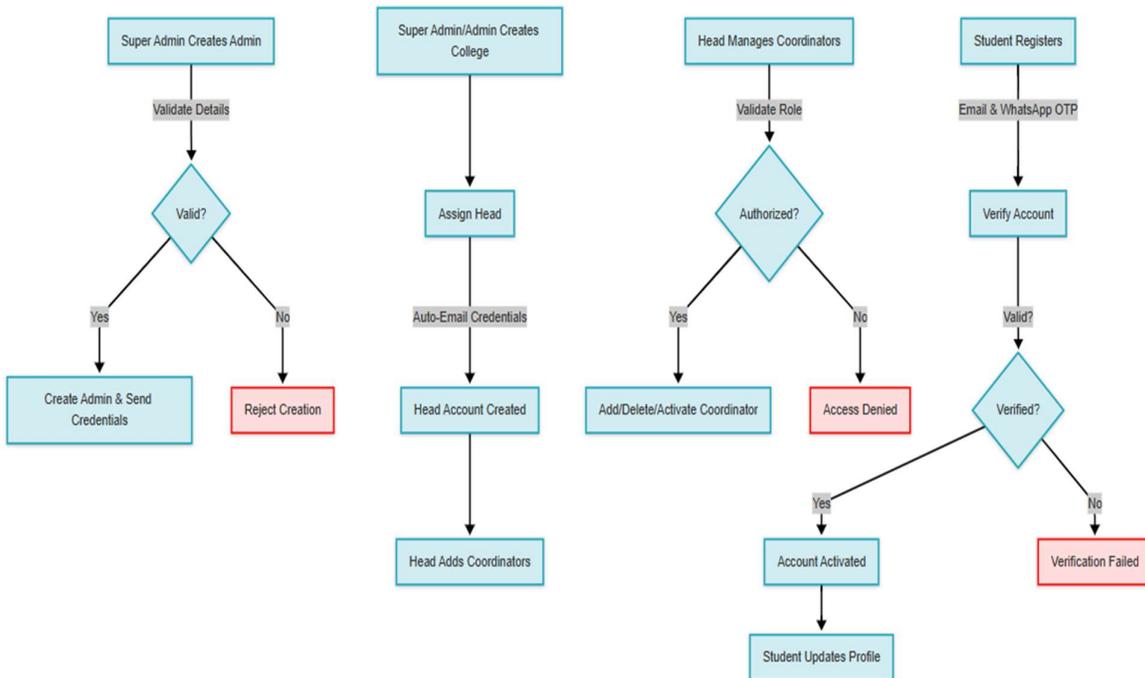
3. ANALYSIS

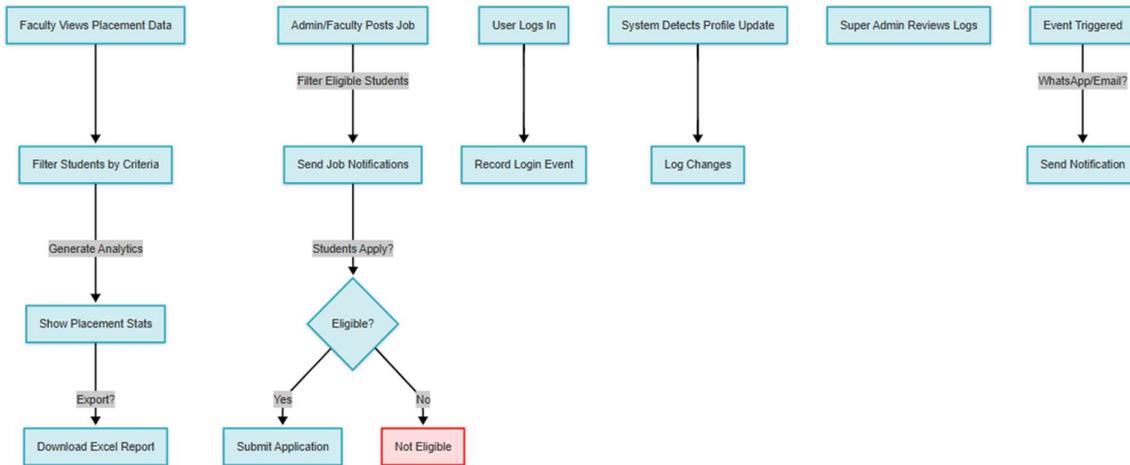
3.1. Use Case Diagram





3.2. Flow of Events





4. IMPLEMENTATION

The Campus Recruitment Gateway project leverages a variety of modern libraries and frameworks to ensure a robust, scalable, and user-friendly application. These libraries are carefully chosen to streamline development, enhance performance, and provide a seamless experience for both students and recruiters. Below is a comprehensive breakdown of the libraries used in the project:

Frontend Libraries

1. **@reduxjs/toolkit**: Simplifies Redux setup with built-in tools for state management.
2. **@tanstack/react-query**: Handles server-state management, caching, and data fetching.
3. **@tanstack/react-query-devtools**: DevTools for debugging React Query.
4. **axios**: Promise-based HTTP client for making API requests.
5. **chart.js**: Library for creating interactive charts and graphs.
6. **class-variance-authority**: Utility for managing dynamic CSS class names.
7. **clsx**: Utility for conditionally joining class names.
8. **cra-template**: Template for Create React App.

9. **framer-motion**: Library for animations and transitions in React.
10. **html2canvas**: Converts HTML elements to canvas for image generation.
11. **js-cookie**: Utility for handling browser cookies.
12. **jspdf**: Generates PDF files from JavaScript.
13. **jwt-decode**: Decodes JWT tokens for authentication.
14. **lucide-react**: Icon library for React applications.
15. **react**: Core library for building user interfaces.
16. **react-chartjs-2**: React wrapper for Chart.js.
17. **react-datepicker**: Date picker component for forms.
18. **react-dom**: DOM-specific methods for React.
19. **react-dotenv**: Load environment variables in React.
20. **react-dropzone**: Handles file uploads with drag-and-drop functionality.
21. **react-hook-form**: Simplifies form management and validation.
22. **react-icons**: Provides a collection of icons for React.
23. **react-redux**: Official React bindings for Redux.
24. **react-router-dom**: Routing library for React applications.

25. **react-scripts**: Scripts and configuration for Create React App.
26. **react-select**: Dropdown and multi-select component for React.
27. **react-toastify**: Displays toast notifications.
28. **recharts**: Composable charting library for React.
29. **tailwind-merge**: Utility for merging Tailwind CSS classes.
30. **tailwindcss**: Utility-first CSS framework for styling.
31. **validator**: Library for string validation and sanitization.
32. **web-vitals**: Measures and reports web performance metrics.
33. **xlsx**: Library for reading and writing Excel files.

5. TESTING METHODOLOGY

The testing methodology for the Campus Recruitment Gateway ensures that the application is robust, reliable, and meets user requirements. It includes **Verification**, **Validation**, and the **Fundamentals of Software Testing** (Black Box Testing and White Box Testing). The testing process is divided into **Unit Testing**, **Integration Testing**, **System Testing**, and **Security Testing**.

Verification:

Verification ensures that the Campus Recruitment Gateway adheres to the defined conditions and design specifications set during the development phase. It involves checking whether the product is being built correctly.

- **Activities:**
 - Reviewing design documents and requirements.
 - Conducting code reviews and walkthroughs.
 - Ensuring that each module or component meets its design specifications.
 - Example: Verify that the student login functionality is implemented as per the design

document.

Validation:

Validation ensures that the Campus Recruitment Gateway fulfills the specified requirements and meets the end-users' needs. It focuses on whether the right product is being built.

- **Activities:**

- Testing the application against user requirements.
- Conducting user acceptance testing (UAT) to ensure the system meets user expectations.
- Example: Validate that students can successfully update their profiles and access placement opportunities.

Fundamentals of Software Testing:

Software testing relies on two primary approaches: **Black Box Testing** and **White Box Testing**.

6. RESULTS

The Campus Recruitment Gateway project has achieved significant success in streamlining the interaction between placement coordinators and students, addressing the challenges of campus recruitment through a robust and user-friendly platform built on the MERN (MongoDB, Express.js, React.js, Node.js) stack. The platform has simplified the recruitment process, enabling placement coordinators to manage job postings, track student applications, and schedule interviews efficiently. Students, on the other hand, can create profiles, upload resumes, and apply for jobs seamlessly, reducing the time and effort required to participate in campus recruitment. Real-time notifications and updates have further enhanced communication, ensuring timely information about job openings, application statuses, and interview schedules. This centralized approach has eliminated the need for manual coordination, reducing errors and improving overall efficiency.

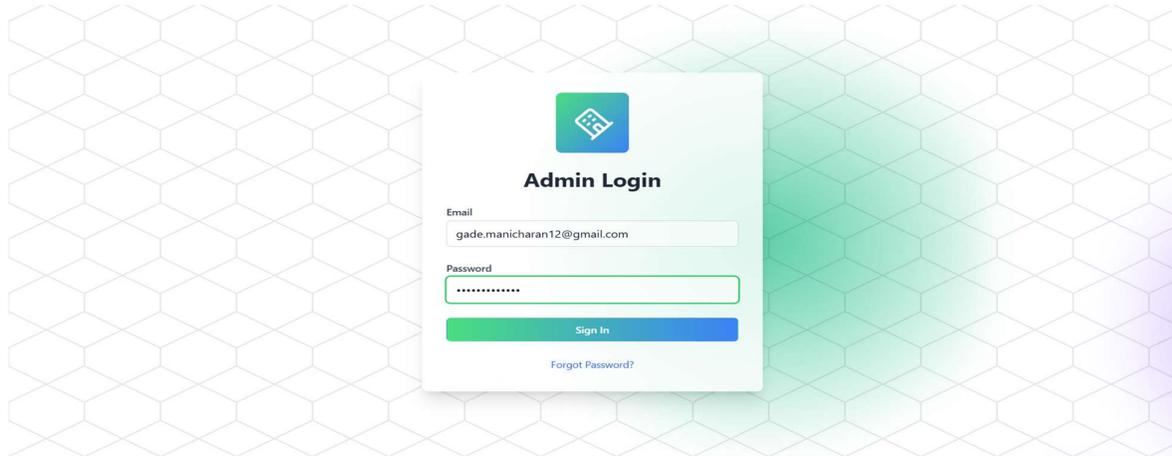


Fig: Admin Login Page

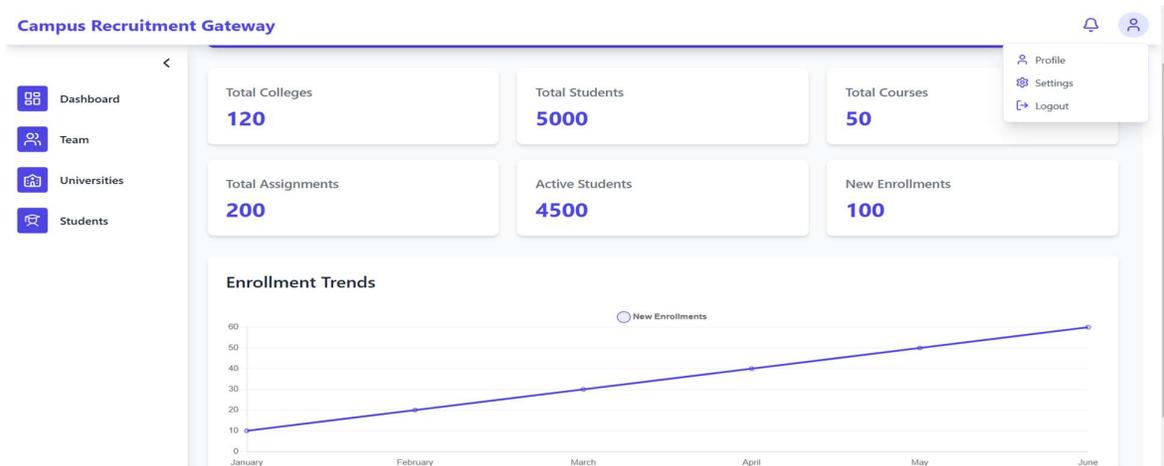


Fig: Dashboard Screen in Admin page

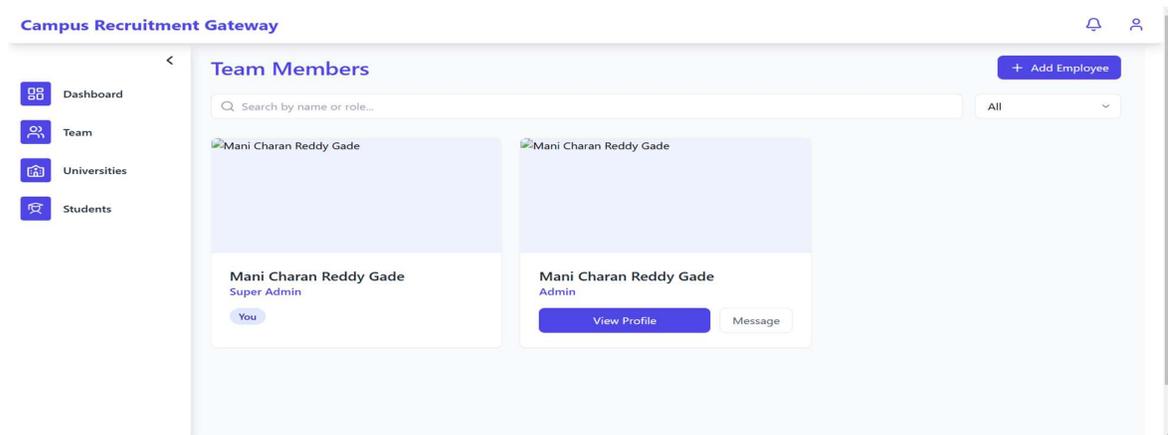


Fig: Team Members Screen in Admin page

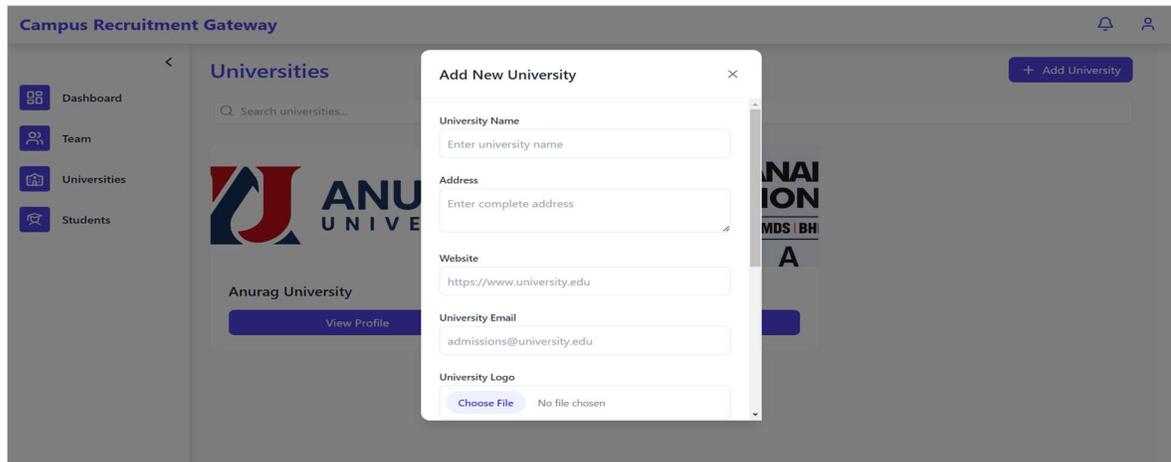


Fig: Adding University Screen in Admin page

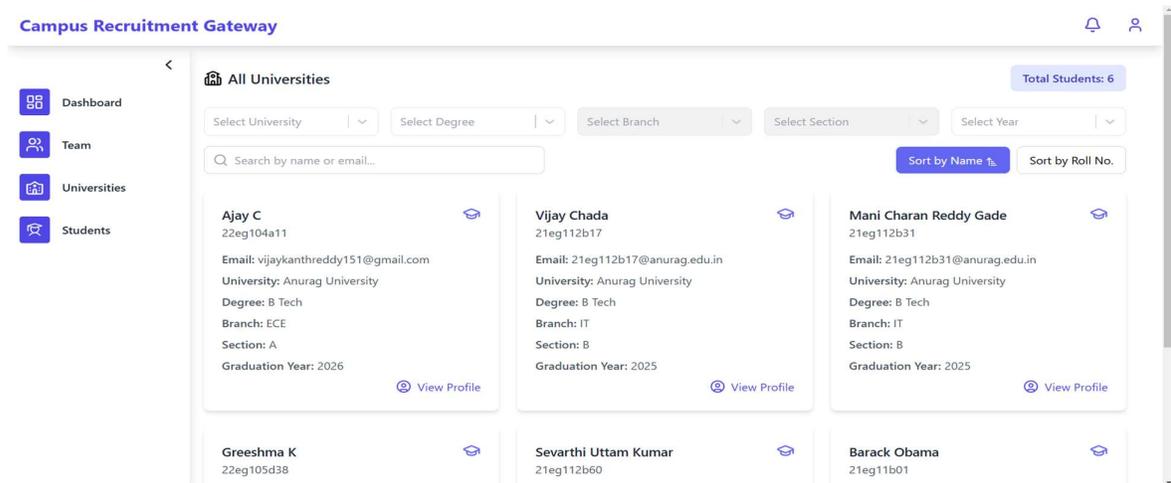


Fig: Students data Screen in Admin page

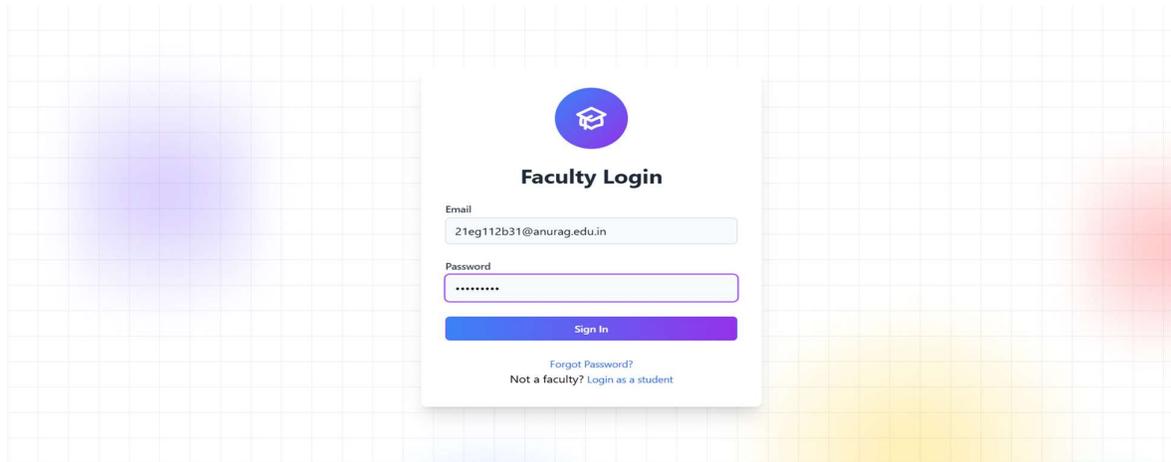


Fig: Faculty Login Page

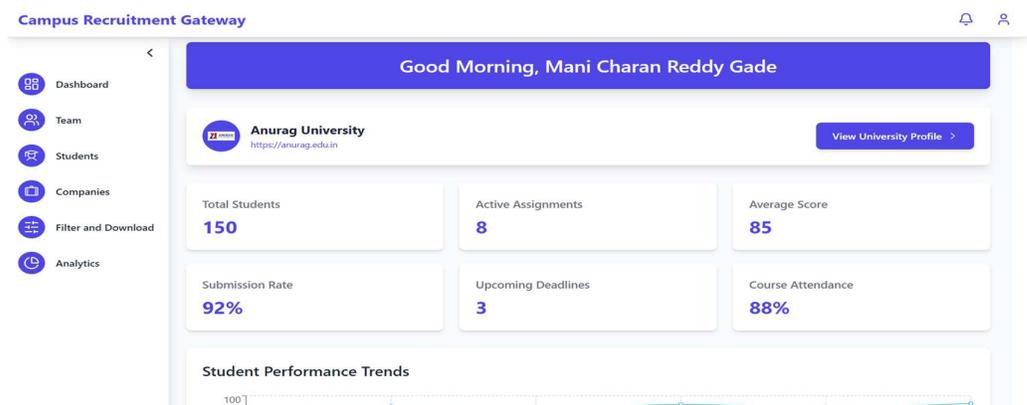


Fig: Dashboard Screen in Faculty page

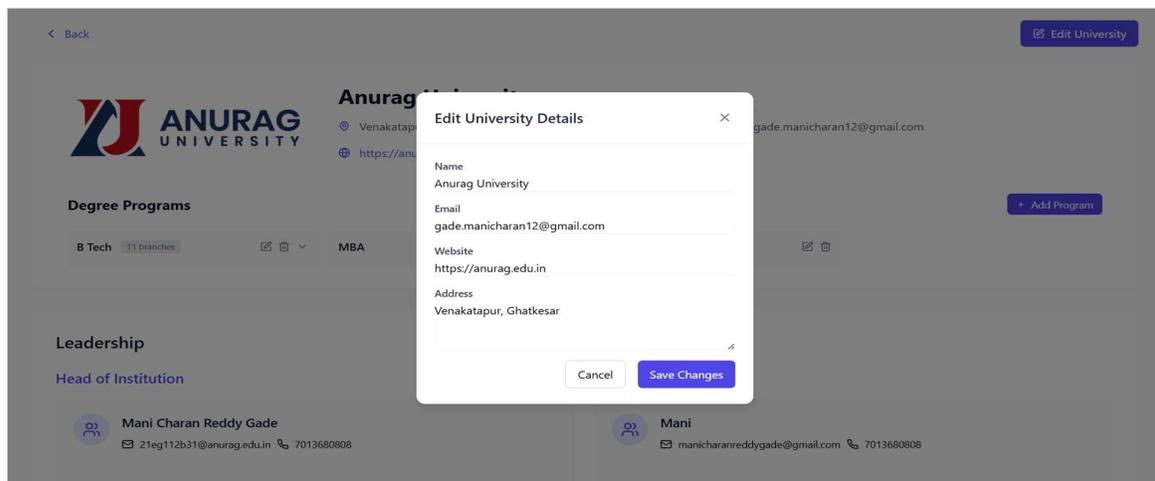


Fig: Editing University Details Screen in Faculty page

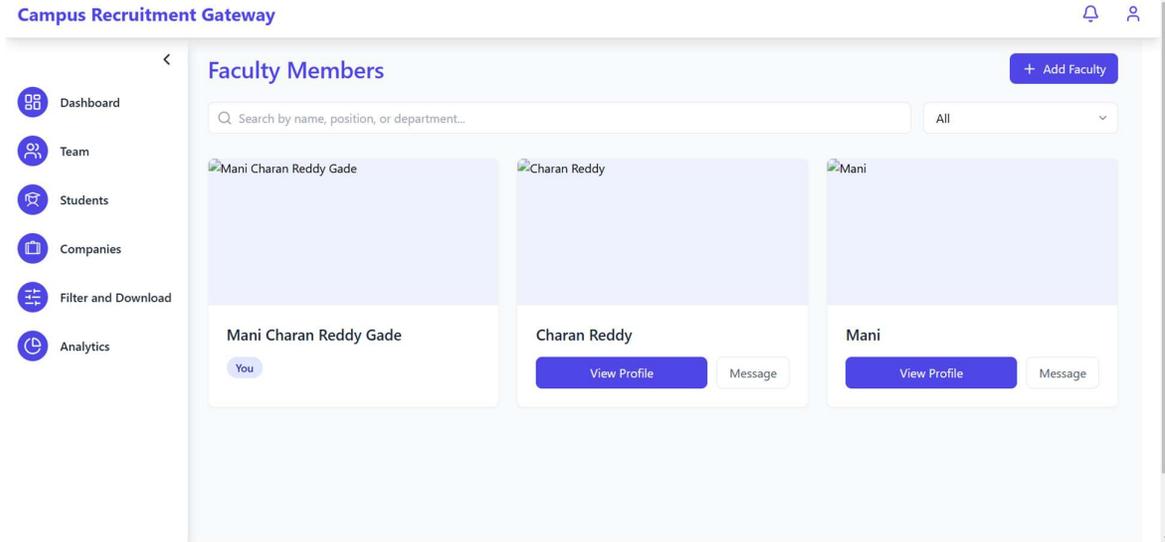


Fig: Team members screen in Faculty page

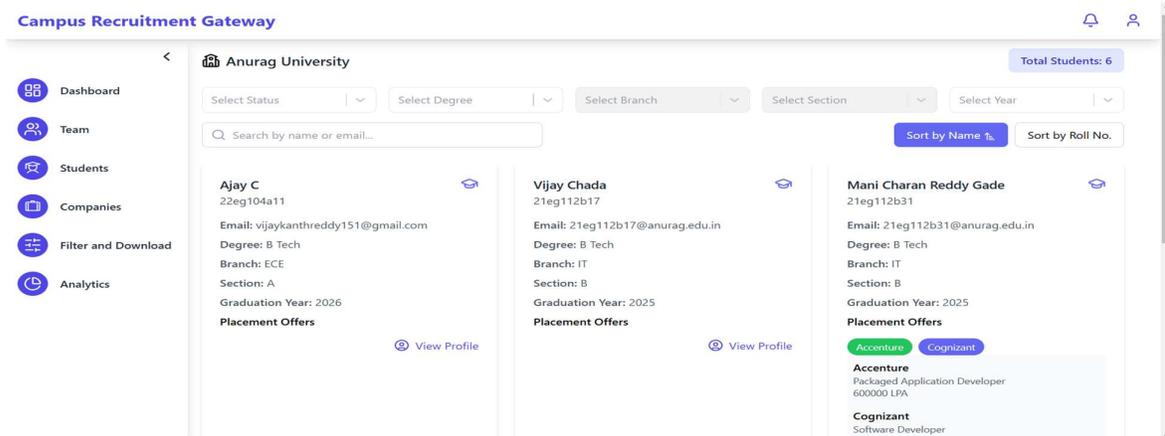


Fig: Handling Students screen in Faculty page

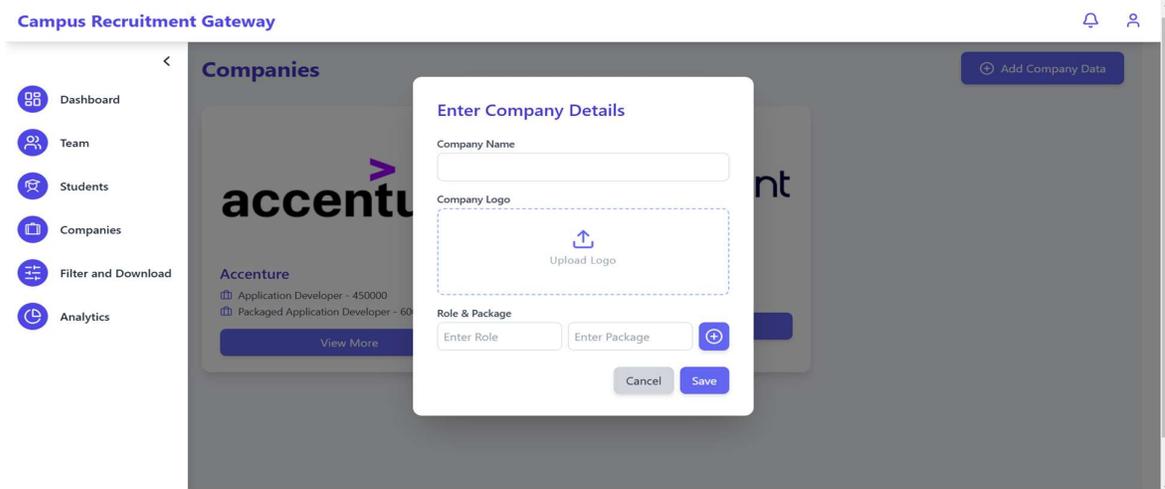


Fig: Adding Company Details Screen in Faculty page

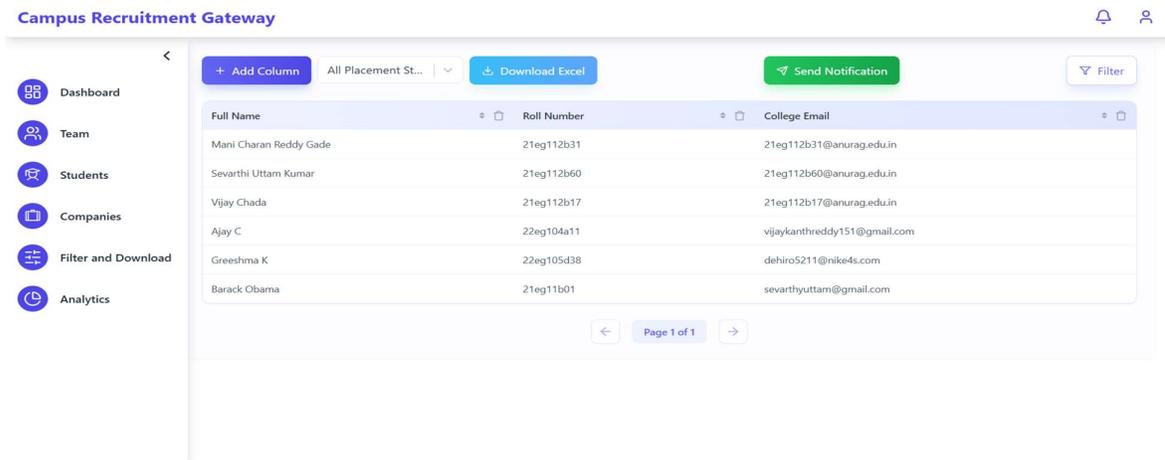


Fig: Filter and Download Screen in Faculty page

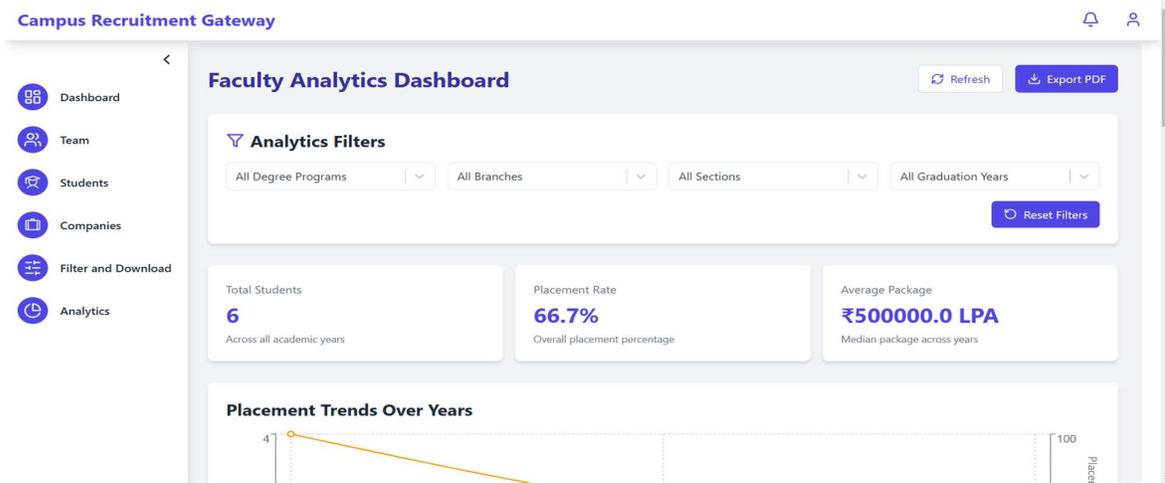


Fig: Analytics Dashboard screen in Faculty page

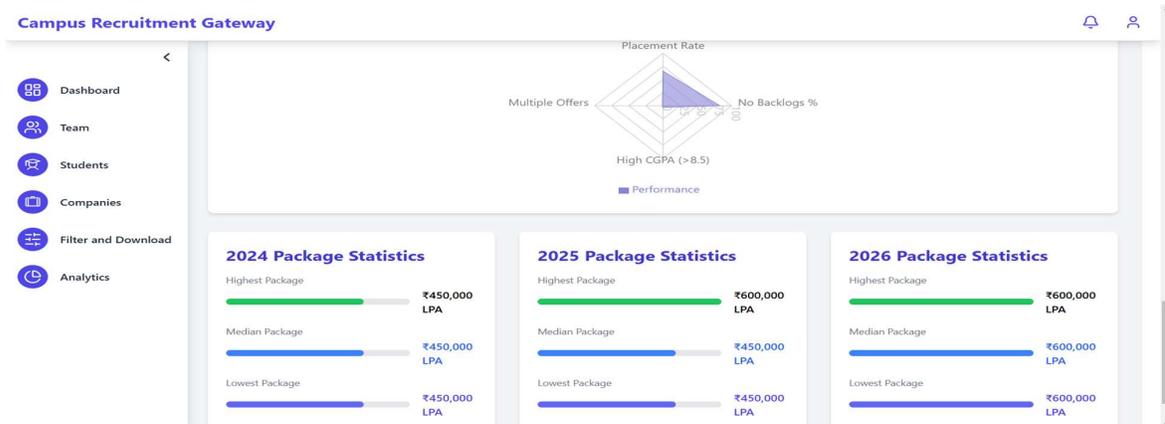


Fig: Analytics Dashboard screen in Faculty page

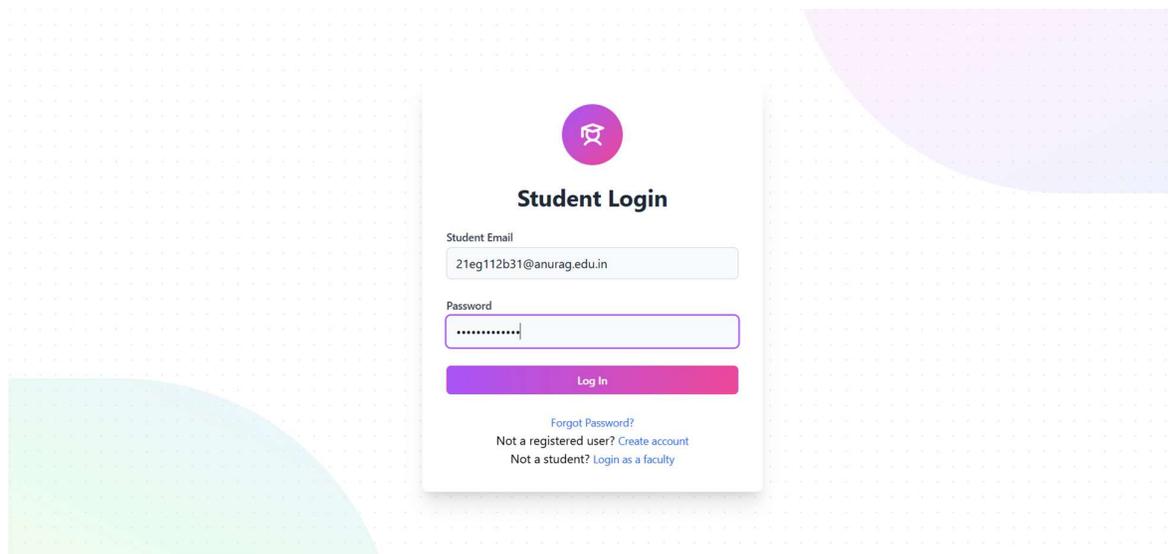


Fig: Student Login Page

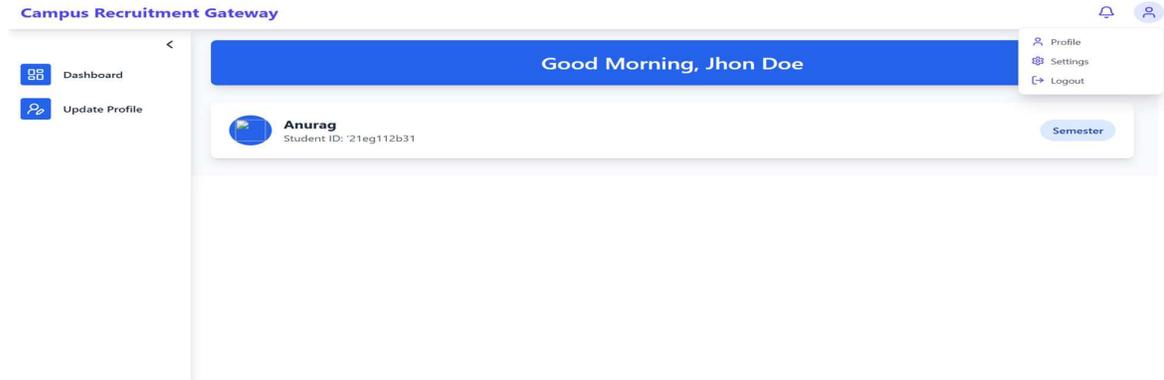


Fig: Dashboard Screen in Student page

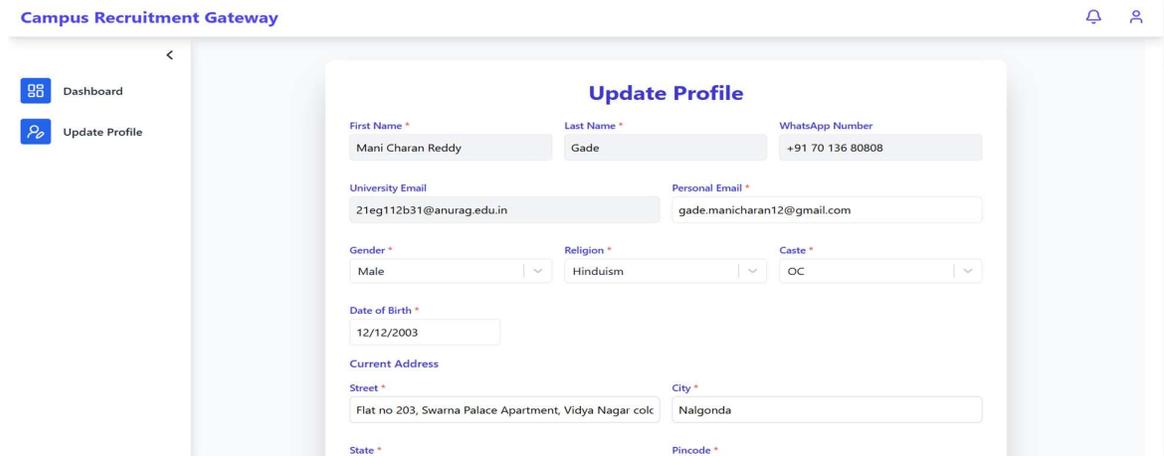


Fig: Profile Update Screen in Student page

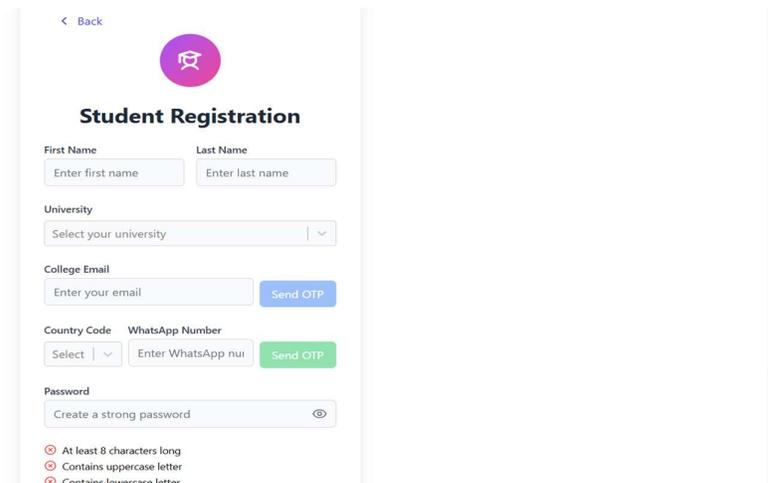


Fig: Student Registration Screen

7-CONCLUSION

The Campus Recruitment Gateway project, designed to facilitate interaction between placement coordinators and students within a campus, serves as an efficient and centralized platform to streamline the recruitment process. By leveraging the MERN (MongoDB, Express.js, React.js, Node.js) stack, the platform provides a robust and user-friendly solution tailored to the needs of students and placement coordinators. Students can create profiles, upload resumes, and apply for job openings, while placement coordinators can manage job postings, track applications, and schedule interviews seamlessly. The platform ensures smooth communication and coordination, eliminating traditional inefficiencies in the recruitment process.

One of the key strengths of this project is its focus on simplifying the workflow for placement coordinators, who play a crucial role in bridging the gap between students and recruiters. By providing them with tools to manage job postings, monitor student progress, and organize recruitment activities, the platform enhances their ability to support students effectively. For students, the system offers a convenient way to access job opportunities, submit applications, and receive updates, all within a single platform. Real-time notifications and an intuitive interface further enhance the user experience for both parties.

Despite being tailored for placement coordinators and students, the platform has the potential for future expansion, such as integrating recruiter access or incorporating advanced features like AI-based job matching and analytics dashboards. These enhancements could further improve the efficiency and effectiveness of the campus

recruitment process.

In conclusion, the Campus Recruitment Gateway project successfully addresses the challenges of campus recruitment by providing a dedicated platform for placement coordinators and students. It simplifies the recruitment process, improves communication, and ensures a seamless experience for all users. By leveraging modern web technologies, the project demonstrates the potential to transform traditional recruitment methods and create a more organized and efficient system for campus placements.

REFERENCES

1. React.js

React Beginner Course 2024 (Vite, Tailwind CSS, TypeScript):

[Coding in Flow Channel](#)

2. Node.js and Express.js

MERNStack Tutorials:

[Dave Gray Channel](#)

3. MongoDB

The ULTIMATE MERNStack Complete Guide (MongoDB, Express, React, Node.js):

[MongoDB](#)

4. MERNStack

MERNStack Projects | MongoDB, Express.js, React, Node.js | Website Projects:

[Geeks For Geeks](#)

5. Project Development

Full Stack Web Development Course- ReactJS, NodeJS, Express, MySQL:

[Pedro Tech Channel](#)