JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

KTU RESULT AGGREGATOR USING CI/CD ACTIONS

Mohammed Sabah K T[1*], Muhammed Fabis N V [2*], Muhammed Rithas K[3*], Ummu Habeeba K M[4*] Ajumol P A[5*]
*Department of Computer Science and Engineering
Mar Athanasius College of Engineering, Kothamangalam, Kerala
1

***Abstract*—In the realm of academic progression, students often grapple with the challenge of accessing their results promptly and efficiently, particularly during peak hours when the KTU website experiences frequent crashes. This issue is further exacerbated for students with back papers, as the lack of an integrated system for calculating CGPA adds to the complexity. Our project, the KTU Result Aggregator, addresses these challenges head- on by providing a streamlined solution tailored specifically for KTU students. By leveraging only the registration number, our platform offers easy access to semester-wise results without the need for extensive personal information, ensuring efficiency and privacy. Key to our project is its robustness in handling various academic scenarios, including cases with back papers, enabling accurate CGPA computation and providing students with a clear overview of their academic progress. Furthermore, our system incorporates load balancing mechanisms to ensure stability and accessibility even during peak hours, preventing down times and ensuring uninterrupted service. More over Our system offers a graphical representation of student performance trends over time, facilitating easy analysis of academic progress and achievements.**

***Index Terms*—— Academic progression, Promptly, Integrated system, Robustness, Accessibility, Academic progress**

I. Introduction

This project addresses the challenge lies the inability of students to access their results promptly during peak hours due to the KTU website's frequent crashes. Furthermore, the lack of an integrated system for calculating CGPA exacerbates the problem, particularly for students with back papers. The cur- rent solution, albeit available, poses its own set of challenges. It requires students to manually input grades for each subject, leading to a time-consuming process. Moreover, accessing semester-wise results scattered across different pages further compounds the inefficiency. Recognizing these challenges, our project endeavors to provide a comprehensive solution. By creating a centralized website, we aim to address both issues simultaneously. Students can access all their semester- wise results on a single page, streamlining the process and saving valuable time. Additionally, our system enables CGPA calculation even in the presence of back papers, eliminating the need for manual grade input.

BACKGROUND

A. React

React is a JavaScript library famend for its efficiency In constructing dynamic person interfaces for internet applica- tions. With React, builders can create reusable UI components, facilitating modular and maintainable code bases. One pop- ular integration with React is Chart.js, a versatile charting library That permits the introduction of visually attractive and interactive charts, graphs, and data visualizations. By leveraging Chart.js within React components, developers can seamlessly integrate dynamic data representation into their applications, enhancing user engagement and comprehension. Form.js, another integral component in React applications, streamlines the process of capturing user input and han- dling form submissions. With Form.js, developers can create forms with custom validation logic, error handling, and data processing functionalities, ensuring a smooth and intuitive user experience. By encapsulating form-related logic within reusable Form.js components, developers can maintain code consistency and promote code reusability across their ap- plications. ResultTable is a React component designed to display structured data in tabular format, providing users with a comprehensive overview of information. Leveraging React's component-based architecture, ResultTable enables develop- ers to organize and present data dynamically, with features such as sorting, filtering, and pagination for enhanced user interaction. By encapsulating table-related functionality within ResultTable components, developers can create versatile and customizable data presentation solutions tailored to their appli- cation's specific requirements. Table is a fundamental building block in React applications, serving as the foundation for orga- nizing and presenting data in a structured format. With Table components, developers can define the structure and layout of tables, specifying attributes such as columns, rows, and cell contents. By encapsulating table-related logic within reusable Table components, developers can promote code modularity and scalability, enabling the seamless integration of tables across various parts of their application. In summary, React's component-based architecture Empowers builders to create modular, reusable, and scalable UI components, facilitating the development of dynamic web applications. Integrating libraries such as Chart.js and implementing custom compo- nents like Form.js, ResultTable, and Table further enhances the functionality and versatility of React applications, enabling developers to build intuitive, interactive, and visually appealing user interfaces.

B. *MongoDB*

MongoDB stands out as a premier NoSQL database man- agement system, prized for its flexibility, scalability, and performance. Adopting a document-oriented data model, Mon- goDB stores data in JSON-like documents, allowing for agile data modeling and dynamic schema evolution. Its scalability features, including sharding and replica sets, make it ideal for handling large volumes of data and high-throughput work- loads. MongoDB's flexibility in data modeling enables the rep- resentation of complex relationships with ease. Its optimized query execution and comprehensive ecosystem of tools further enhance its appeal, providing developers with a powerful solution for building and managing modern, data-driven appli- cations across a diverse range of use cases. Whether for small-scale projects or enterprise-level deployments, MongoDB offers unparalleled flexibility, scalability, And performance, making it a favored preference for developers worldwide.

C. *GitHub Actions*

GitHub Actions is a famous CI/CD platform for automating your build, test, and deployment pipeline. Docker provides a hard and fast of authentic GitHub Actions so that it will use in your workflows. These official actions are reusable, easy-to-use components for building, annotating, and pushing images. It ns allows to configure your workflow to be triggered once a specific event occurs in the repository. For example, when a trouble is created or while a pull request is opened. A workflow contains one or several jobs running in parallel or sequential order. Each job runs inside a container or in a separate virtual machine (VM) runner. Additionally, each job includes one or several steps that run a predefined script or an action, a reusable extension that simplifies your workflow.

D. *Docker Container*

Docker is a fixed of platform as a service (PaaS) products That use OS-degree virtualization to supply software program in packages called containers. The provider has each unfas- tened and top rate tiers. The software program that hosts the bins is known as Docker Engine. It turned into first launched in 2013 and is advanced through Docker, Inc. Docker is a device this is used to automate the deployment Of programs in light-weight packing containers in order that programs can work efficiently in different environments in isolation. It Can bundle an utility and its dependencies in a virtual container which could run on any Linux, Windows, or macOS computer. Computer. This permits the software to run in a selection of Locations, including on-premises, in public (see decentralized computing, distributed computing, and cloud computing) or private cloud. When running on Linux, Docker uses the resource isolation features of the Linux kernel (such as c groups and kernel name spaces) and a union-succesful document system (such as Overlay FS) to allow containers to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines. Docker on macOS uses a Linux virtual machine to run the containers. Because Docker containers are lightweight, a single server or virtual machine can run several containers simultaneously. A 2018 analysis found that a typical Docker use case involves running eight containers per host, and that a quarter of analyzed organizations run 18 or more per host. It can also be installed on a single board computer like the Raspberry Pi.

E. *Load Balancing*

Load balancing is a technique used in computer networking and distributed systems to distribute incoming network traffic or workload across multiple servers, CPUs, disks, or other resources. The main purpose of load balancing is to optimize resource utilization, maximize throughput, minimize response time, and ensure high availability and reliability of applications and services.
.

II. PROPOSED MODEL

The methodology for our KTU Result aggregator project encompasses various stages, each crucial for the successful development and implementation of the forecasting models. Here's a breakdown of the methodology:

A. Application Flow

Our project begins by extracting the batch year from user input and utilizing it to retrieve the exam IDs for all batches from the KTU database. This initial step ensures that we have the necessary information to access the relevant examination data. Subsequently, we leverage this data to retrieve the required information from the KTU database via API calls, carefully cleaning the API responses to ensure coherence and accuracy. Once we have obtained the pertinent academic data, we proceed to calculate the Cumulative Grade Point Average (CGPA) for each student. Finally, we present this calculated CGPA data through a user interface, providing students with a clear and intuitive platform to assess their academic performance effectively. Through this streamlined process, our project aims to empower students with the tools they need to track their progress and strive for academic excellence.

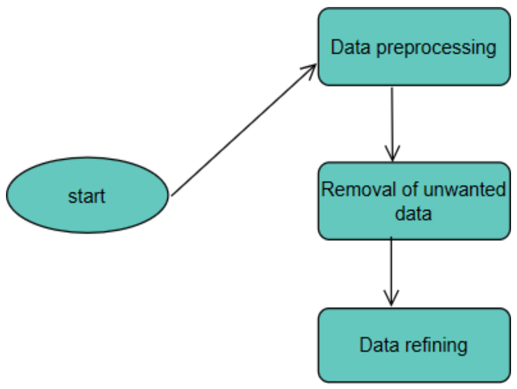


Fig. 1. Application Flow

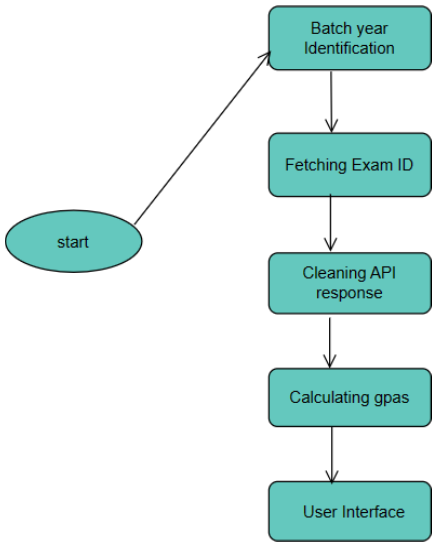


Fig. 2. Data Pre-Processing

B. Data Collection

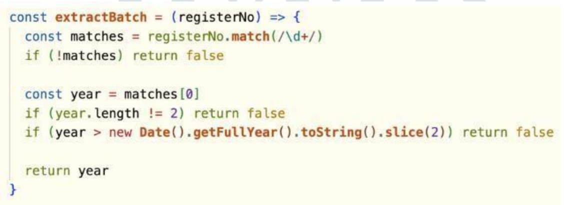
D. Continuous Improvement

To ensure continuous improvement and user satisfaction, we've implemented robust feedback mechanisms within our development process. These mechanisms allow us to gather user input regularly and analyze it systematically. By prioritizing improvements based on user suggestions and addressing pain points, we ensure that our platform evolvesin line with user needs. Embracing an iterative User Input Extraction: Collect data by extracting the batch year from the user's input, serving as a pivotal identifier for accessing relevant academic records. -Exam ID Retrieval: Upon obtaining the batch year, we proceed to retrieve the corresponding Exam ID, a crucial parameter necessary for querying the KTU API. -KTU API Querying: Leveraging the extracted Exam ID, we utilize the KTU API to fetch comprehensive data pertaining to the specified batch year, enabling access to a wealth of academic information. -Data Aggregation: Through a custom function, we combine all the data encompassing various academic parameters, including grades, course credits, and examination results, facilitating a comprehensive analysis of student performance.

C. Data Preprocessing

To enhance the accuracy and reliability of our data, we've undergone a thorough text preprocessing phase. This involved several steps aimed at cleaning and normalizing the collected data. First, we systematically removed null values and performed garbage collection to eliminate any irrelevant or erroneous information. Then, through refining the data, we meticulously standardized formats, corrected spelling errors, and removed inconsistencies. Additionally, we employed techniques such as tokenization, stemming, and lemmatization to ensure uniformity across the dataset. By meticulously refining and standardizing the data, we've significantly bolstered the quality of our output, laying a solid foundation for robust analysis and insights.

Fig. 3.Fetching Result



```
const extractBatch = (registerNo) => {
  const matches = registerNo.match(/\d+/)
  if (!matches) return false

  const year = matches[0]
  if (year.length != 2) return false
  if (year > new Date().getFullYear().toString().slice(2)) return false

  return year
}
```

approach to development, we release small, incremental updates driven by user feedback. This iterative process enables us to continuously refine features, enhancing both usability and effectiveness. By actively listening to our users and adapting our platform accordingly, we strive to deliver an exceptional user experience that meets and exceeds expectations.

III. DESIGN AND IMPLEMENTATION

A. Algorithm for Result Fetching:

1. **collecting user input**: Begin by collecting registration number from user input.
2. **Batch year identification**: extracting the batch year from user input can be done using regex
3. **Fetching exam IDs for that batch** : fetching exam ID for corresponding batch year from the dataset
4. **Fetching exam IDs for that batch** : fetching exam ID for corresponding batch year from the dataset
5. **Fetching results using KTU API** :fetching results using KTU API with registration number and exam IDs
6. **Cleaning API response**: extracting the relevant data and formatting it in JSON

B. Algorithm for CI/CD Actions:

1. **Define Workflow**: Start by creating a github/ workflows directory in your GitHub repository. In this directory, create YAML files to define your workflows. A common filename is 'ci-cd.yaml'.
2. **Trigger**: Specify the events that trigger your workflow. For example, you might want to trigger the workflow on every push to the main branch or when a pull request is opened.
3. **Jobs**: Define one or more jobs within your workflow. Each job represents a set of steps that run sequentially or in parallel.
4. **Steps**: Inside each job, define the steps to be executed. Steps can include actions, commands, or scripts. Actions are reusable units of code defined in separate repositories.
5. **Continuous Integration**: Checking out the code, Installing dependencies ,Running tests.
6. **Continuous Deployment**: Building and packaging application artifacts, and Deploying application to a server or platform.
7. **Error Handling**: Implement error handling in your work- flow to gracefully handle failures and notify relevant parties.

C. Algorithm for load balancing:

1. **Start**: Keep track of available servers and their capacities.
2. **Check server loads.**: When a new request comes, Choose a server based on: Round Robin (rotate through servers),Least Connections (choose least busy server),Random (pick any available server)
3. **checking status and restoring**: Periodically check servers' status and restore them if recovered, and Make sure sessions stick to the same server if needed.
4. **Reliability ensuring**: Keep monitoring and adjust bal- ancing as necessary, and Ensure backup load balancers for reliability.

VI. RESULT

Our project entails the development of a Result Management System aimed at providing seamless access to students' aca- demic performance records across multiple semesters. With a simple input of the register number, students gain instant access to their comprehensive semester-wise results. Further- more, the system incorporates features for updating results post-supplementary exams and revaluation requests, ensuring that students have access to the most accurate and up-to- date information regarding their academic progress.it provides Instant Result Access like Upon entering their register number, udents can effortlessly retrieve their semester-wise results in a single page, streamlining the process of accessing academic records. also The system includes an intuitive "Update" but- ton that allows students to receive revised results following supplementary exams or revaluation requests. This ensures that students have access

to the latest information regarding their academic performance. A graphical representation of students' performance over each semester provides valuable insights into their academic journey. This feature enables students to visualize their progress and identify areas for improvement effectively. By consolidating semester-wise results into a single, easily accessible platform, the system facilitates efficient evaluation of academic performance, empowering students to track their progress effectively. also The ability to update results post-supplementary exams and revaluation requests ensures that students receive timely and accurate information, allowing them to make informed decisions re- garding their academic pursuits. The graphical representation of results offers a holistic view of students' performance trends, enabling them to identify strengths and weaknesses across different semesters. This analysis aids in formulating strategies for academic improvement and goal setting .the Result Management System not only simplifies the process of accessing academic records but also empowers students to take ownership of their learning journey. By providing instances to results, facilitating updates, and offering insightful graphical representations, the system equips students with the tools they need to evaluate their performance effectively and strive for academic excellence..

VII. FUTURE SCOPE

In our project's future development road map, we aspire to broaden its utility and inclusivity by introducing several enhancements. One such improvement involves facilitating students to generate a consolidated PDF document containing their semester-wise results. This feature aims to streamline the process of reviewing academic progress by providing a conve- nient and holistic overview in a single document. Furthermore, we plan to address the common challenge of managing back papers or failed subjects by implementing a feature that allows students to adjust their grades. This functionality will enable students to anticipate their performance in upcoming supplementary exams and substitute expected grades for failed subjects, providing a more accurate reflection of their aca- demic standing. Moreover, our project intends to expand its scope beyond the 2019 scheme to include data from other academic schemes, such as the 2015 scheme. Additionally, we aim to incorporate results from various academic programs like B. Arch, B. Des, M. Tech, and others, ensuring that our platform caters to a broader range of students across different disciplines. By integrating these enhancements, our project seeks to evolve into a comprehensive and inclusive platform that empowers students to assess their academic performance effectively and plan their educational journey with confidence.CONCLUSION

In conclusion, our research significantly enhances the ex- perience for BTech students under KTU, providing them with unprecedented convenience and efficiency in evaluating their academic results. Even amidst downtimes of the KTU server, our platform ensures swift access to result data, enabling students to seamlessly assess their performance. Additionally, with the ability to access published results via our website, students can efficiently gauge their progress using intuitive visual representations. By delivering this innovative solution, we aim to empower students with a reliable and accessible tool for monitoring their academic journey, irrespective of external server constraints, thus greatly improving their overall educational experience.

REFERENCES

[1] M. Kuhrmannetal., "What Makes Agile Software Development Agile?," in IEEE Transactions on Software Engineering, vol.48, 2022,IEEE.
[2] M. Bano, D. Zowghi and F. da Rimini, "User Involvement in Software Development: The Good, the Bad, and the Ugly," in IEEE Software, vol. 35, November/December 2018, IEEE.
[3] Y. R. Hoda, N. Salleh and J. Grundy, "The Rise and Evolution of Agile Software Development," in IEEE Software, vol. 35, September/October 2018,IEEE
[4] J. P. Bowen, M. Hinchey, H. Janicke, M. Ward and H. Zedan, "Formality, Agility, Security, and Evolution in Software Development," in Computer, vol. 47, Oct. 2014, IEEE.
[5] M. A. Shahid, N. Islam, M. M. Alam, M. M. Su'ud and S. Musa, "A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach," in IEEE Access, vol. 8, 2020,IEEE.
[6] V. Kumar and C. Khosla, "Data Cleaning-A Thorough Analysis and survey on Unstructured Data," 2018 8th International Conference on Cloud com- puting, Data Science Engineering, 2018,IEEE.