



Transactions viewer: A web application to perform transaction functionalities based on filters.

Vishal Reddy¹, Dr. Priya D²

Undergraduate Student, Department of Information Science and Engineering,

RV College of Engineering, Bengaluru, India¹

Assistant professor, Department of Information Science and Engineering,

RV College of Engineering, Bengaluru, India²

Abstract: The project is mainly intended to manage the asynchronous requests, responses and provide the filter queries for the transactions of various tenders. The UI used here is the transactions viewer framework. The transactions viewer UI is used to render the data based on the filters provided along with authentication and authorization for the user. The transactions viewer server performs CRUD operations on the transactions using graphql. GraphQL has a query operation that manages reading the data and mutations for updating, creating and deleting the data fields. The project offers huge benefits by handling the majority of client requests asynchronously for all the tenders. This saves a lot of time by not requesting the appropriate queries in an order. The platform also offers additional features to improve user-experience, like that of multiple requests management to queries.

Keywords: React, Material Design, User Interface, User Experience, Web Application.

I. INTRODUCTION

The project focuses on creating an upgraded version of the transactions viewer framework which is used to fetch all the transactions of various tenders based on the filters provided by the user. First, there are two types of classifications namely, tenderA and tenderB (all the other tenders in Payments Platform). The current project is Transactions viewer framework which also manages the transactions of tenderB apart from one tenderA.

The user can apply to get all the transactions based on the filters provided and can also export the data as csv or excel if need be. The framework provides various functionalities such as exporting, customizing columns, importing, search query wherein users can specifically search for a particular transaction. The UI passes on the data to the transactions viewer 2.0 server plugin which will manage all the asynchronous requests for all the tenders.

The plugins for tenders will then fetch the required data from the database where that particular data is being stored. For example, the transaction data for tenderA is stored in Google BigQuery whereas the database used for tenderB is Azure Cloud SQL. The UI testing is done using react-testing library and jest to import the test functionalities. Whereas for the plugins for the backend services to manage other tenders and the data asynchronously, Junit and Mockito are used for testing.

II. RELATED WORK

A literature survey was carried out to understand the existing transaction systems. Similar work has been done in the past in this field. Some earlier studies have performed certain transactional queries wherein the whole data or redundant data is being fetched. Functionalities such as Customizable columns, dynamic change of columns on the UI, Export/download button have not been implemented.

The authors of [1] describe a comparative study depicting the difference between the model of a donation website and the websites that are currently functioning in the field. Firstly, the technologies used in the model along with their benefits and shortcomings are introduced. The donation site model created uses Bootstrap, MaterialUI, Font-awesome for its styling. They can be used by including their import link from their website in their code. Javascript, JSON HTML for creating its skeletal base. Firebase and its API for databases. NodeJS and Npm to manage its packages and dependencies.



[2] describe an advanced version of online education platforms. Celestial Learning is a one-stop solution for authors and subscribers to enhance their learning. To make a lecture effective, proper involvement of both learners and authors is crucial. To reinforce the teaching and learning process, the proposed system provides a web application platform to the content creator to deliver their courses to the audience irrespective of their physical location. Even the students/subscribers could access the course at a flexible pace. The system revolves around two parts. The first part and main distinguishing factor of the system is the security which is lacking in the existing solutions. Due to less security, the courses get uploaded in an unauthorized manner which reduces the credibility of authors. The second part of the system aims at providing a hassle-free experience of learning with some key features such as responsive and attractive User Interface (UI), Live sessions, full-fledged analytical performance manager. An open discussion forum for authors and subscribers which facilitates to resolve their query. The front end of the web application is in React, Material UI whereas the Node-Express server handles the backend and for a user authentication method used the JSON Web Token (JWT). In terms of storage, the system uses MongoDB and cloud platforms for deploying the videos.

[3] describes React JS which is JavaScript library that is developed to design rich UI's. It is one of the famous frontend libraries among the others. It incorporates the 'View' concept from the MVC model(Model-View-Controller). React JS is backed by Facebook, Instagram and community of frontend developers and organization. React JS is emerging as one of the fastest and easiest frontend libraries to be used in web applications as it has fully component based architecture. The purpose behind the paper is to showcase how and why React JS is emerging as one of the finest open source JavaScript libraries.

[4] discusses React and its popularity. React is one of the popular web frameworks that has gained importance over other frameworks such as Angular, Vue, etc.. This is because of its implementation of Virtual DOM, whose primary objective is to enhance the overall performance of the application. However, there are certain things that one has to keep in mind before designing the applications. Failing to anticipate the problems that may occur component hierarchy will lead to performance degradation. Some of the commonly faced problems are component re-rendering, application lag due to background computations being run, lag due to processing large data sets in a single stretch, etc. The paper will describe some of the practical ways of overcoming such problems within the application, thus enhancing the performance of the ReactJS App in a production environment. The paper will also describe a time-efficient search algorithm that can be used for searching objects in a large data set.

[5] Describes the recommendations for a responsive user interface (UI) implementation. Google guidelines for UI implementation - Material Design, are briefly presented and compared against Nielsen's design principles. It has been identified limitations in preserving interaction design principles while implementing Material Design Guidelines. With the objective to achieve flexible and responsive layouts respective improvement recommendations are discussed. Sample design cases are presented, illustrated by screens before and after the implementation of the recommendations on Android devices with different resolutions.

[6] Describe the evolution of Web technologies that triggered the appearance of frameworks and libraries aiming to improve the work conducted by those who use them. To maximize customer satisfaction and create solutions under current market standards, companies must adapt their products, with focus on web-based solutions, to enhance the usability and the flow of their use. In this way, the paper presents a real case study, where the company humansoft, revised one of its main solutions, humanportal, using one of the most recent and used web technology, React.js. React.js is a framework specially designed to deal with the visualization layer of web applications and their use in a business context, namely in the human resources management field, which is the focus of the human portal solution. With this process, there was a real customer satisfaction degree, having them highlight the greater flexibility and ease of use of the product, thus allowing optimization and capability of the human resources management procedures.

[7] Describe how code smells impact the UI. Just like code smells in source code, UI design has visual design smells. The 93 don't-do-that guidelines in Material Design, a complex design system created by Google are being studied here. The detection accuracy of UIS-Hunter is high (precision=0.81, recall=0.90) on the 60,756 UIs of 9,286 apps. A guideline gallery with real-world UI design smells that UIS-Hunter detects for developers to learn the best Material Design practices was built. Our user studies show that UIS-Hunter is more effective than manual detection of UI design smells, and the UI design smells that are detected by UIS-Hunter have severely negative impacts on app users.

III. METHODOLOGY

This section gives a brief overview of the different designs and methodologies that can be adopted while working on any web applications or websites. The different designs and their use cases has been explained. The project involves using



Java, spring boot and webflux framework to create plugin services for getting all the details of transactions of various tenders. Firstly, there are 3 plugins for tenderA, tenderB (the latter will come under Payments Platform)

A. Backend Plugins

The queries are fetched by GraphQL which is another form of REST API wherein the user can dynamically choose only those queries that are required. REST API gives you all the data that is related to the query hence GraphQL is preferred here.

The user enters all the required filters onto the UI that is created using react-redux form which stores and manages all the required filter query fields such as transaction type, transaction type, server info, business info, query options (ascending or descending, sort by tender type, country or server, etc).

B. Transactions Viewer UI

The user can then apply to get all the transactions based on the filters provided and can also export the data as csv or excel if need be.

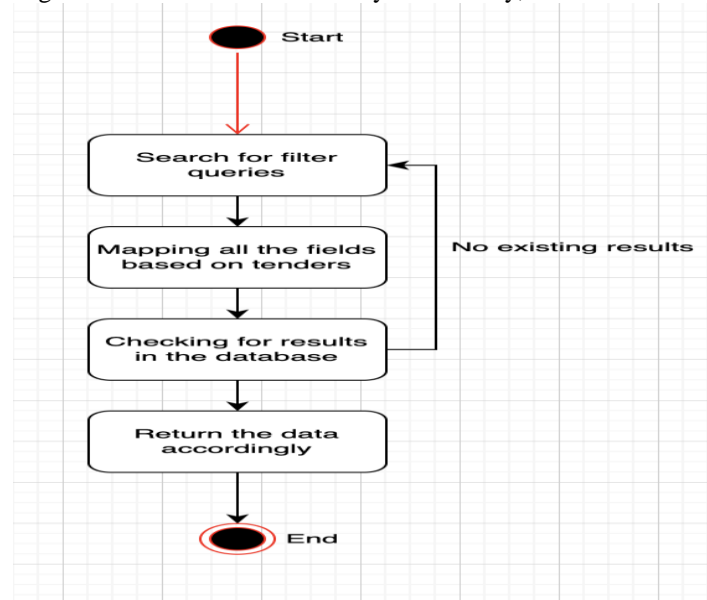
The framework provides various functionalities such as exporting, customizing columns, importing, search query wherein users can specifically search for a particular transaction.

The UI passes on the data to the transactions viewer 2.0 server plugin which will manage all the asynchronous requests for all the tenders.

The plugins for tenders will then fetch the required data from the database where that particular data is being stored. For example, the transaction data for tenderA is stored in Google BigQuery whereas the database used for external products tender is Azure Cloud SQL.

C. Testing

The UI testing is done using react-testing library and jest to import the test functionalities. Whereas for the plugins for the backend services to manage other tenders and the data asynchronously, Junit and Mockito are used for testing.



IV. CONCLUSION

The project was found to meet the objectives and all the user requirements satisfactorily. The application was showing accurate results after integrating with its associated services. The project is a full fledged application to manage and showcase all the data in a set of tenders.

This research considered a range of aspects that can be used to make a better web application. UX has emerged as a critical component of application design and development, particularly for web-based programmes and websites. They are intended to be simple to use and correlate with business cases. A design that is user friendly and simple while having a rich content always gets more users. The content and design should be user centric. Dynamic addition of fields is allowed so that all the transactions are shown and can be changed as per the user's requirements. React grid components giving different dropdown fields like searchableReduxDropdown, simpleReduxDropdown and simple text fields. The user is able to fetch the transactions based on the filters provided for tender.



ACKNOWLEDGMENT

The authors of this paper express their gratitude to **Mr. Shivam Sehdev** and his colleagues of Walmart Global Tech for their time and guidance.

REFERENCES

- [1] S. P. Tiwari, G. Phonsa and P. Singh, "Study and Comparative analysis of Donation based websites," 2021 International Conference on Computing Sciences (ICCS), 2021.
- [2] S. M. Sinha, V. P. Kale, R. S. Sangle, S. R. Bhoite, V. G. Kottawar and P. B. Deshmukh, "Celestial Learning: Secure eLearning platform using Node-Express," 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2021.
- [3] Pratik Sharad Maratkar , Pratibha Adkar "React JS - An Emerging Frontend JavaScript Library" Iconic Research And Engineering Journals, 4(12), 2020.
- [4] A. Javeed, "Performance Optimization Techniques for ReactJS," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019[5] E. Kalac, N. Borovina and D. Boskovic, "Preserving interaction design principles while implementing Material Design Guidelines," 2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH), 2021.
- [6] M. de Sousa and A. Gonçalves, "humanportal – A React.js case study," 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), 2020.
- [7] B. Yang, Z. Xing, X. Xia, C. Chen, D. Ye and S. Li, "Don't Do That! Hunting Down Visual Design Smells in Complex UIs Against Design Guidelines," 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021.
- [8] S. Parsons, "Current approaches to handling imperfect information in data and knowledge bases", IEEE Transactions on Knowledge and Data Engineering, Volume: 8, Issue: 3, June 2016 .
- [9] S. Sonnentag, C. Niessen and J. Volmer, "Expertise in software design", in The Cambridge handbook of expertise and expert performance, pp. 373-387, 2016.
- [10] N. Oliveira, N. Andrade, and K. Reinecke, "Participation differences in Q&A sites across countries: Opportunities for cultural adaptation," in Proceedings of the 9th Nordic Conference on Human-Computer Interaction - NordiCHI '16, Gothenburg, Sweden, 2016, pp. 1–10
- [11] Hassan, Lobna Hamari, Juho, "Gamification of E-Participation: A Literature Review", The 2015 IEEE International Conference on Systems, Man, and Cybernetics, 12 Aug 2015
- [12] William R. Penuel, Heather C. Hill, "Building a Knowledge Base on Research-Practice Partnerships", 2015 IEEE International Conference on Information and Automation, 2016
- [13] New Vision for Education: Unlocking the Potential for Technology, Geneva:World Economic Forum, 2015.
- [14] Justin Hou, "Does Gamification Work? -- A Literature Review of Empirical Studies on Gamification", 47th Hawaii International Conference on System Sciences, 6-9 Jan. 2014
- [15] ICT for Work: Digital skills in the workplace Brussels: European Commission Directorate-General of Communications Networks Content & Technology, 2014.
- [16] A. Halavais, K.H. Kwon, S. Havener and J. Striker, "Badges of friendship: Social influence and badge acquisition on stack overflow", 47th Hawaii international conference on System Sciences, pp. 1607-1615, 2014.
- [17] A. Robins, J. Rountree and N. Rountree, "Learning and teaching programming: A review and discussion", Computer science education, vol. 13, pp. 137-172, 2013.