

TEST DATA REPORTING AND VISUALIZATION TOOL

Dr S.Mani¹, Kaushik Sarkar², Goutam Mahana³, Rishika Kumari⁴, Kunwar Divas Singh⁵

¹Professor Department of Computer Science and Engineering T John Institute of Technology
Bengaluru, India

^{2,3,4,5}Department of Computer Science and Engineering T John Institute of Technology Bengaluru, India
manicovai@gmail.com, goutammahana@gmail.com, kaushiksarkar6190@gmail.com, kumaririshika@gmail.com,
kunwardivassingh@gmail.com

DOI: <https://www.doi.org/10.58257/IJPREMS38374>

ABSTRACT

The project Test Data Reporting and Visualization Tool is designed to address the challenges of managing, analyzing, and presenting test data efficiently. Built entirely in Python, the tool utilizes Dash and Plotly to deliver a powerful yet intuitive platform for data reporting and visualization. This solution caters to teams and organizations seeking to transform raw test data into meaningful insights without relying on complex or resource-intensive software. This tool enables users to seamlessly import data from multiple sources and formats, ensuring compatibility with various testing environments. It features automated data processing capabilities that reduce the need for manual intervention, enhancing efficiency and accuracy.

1. INTRODUCTION

In the ever-evolving landscape of software development, testing is a critical process to ensure the quality, functionality, and reliability of applications. With the increasing complexity of software systems, the volume and variety of test data generated have grown exponentially. Managing this data effectively and turning it into actionable insights is a significant challenge for development teams. To address this, tools that can automate data processing, streamline reporting, and deliver impactful visualizations are essential. The Test Data Reporting and Visualization Tool is designed as a robust and user-friendly solution to these challenges. Built entirely in Python, this tool leverages Dash for creating interactive web-based dashboards and Plotly for producing high-quality, dynamic visualizations.

2. LITERATURE REVIEW

Effective test data reporting and visualization are essential for efficient software testing, yet traditional tools often lack automation and scalability. The Test Data Reporting and Visualization Tool leverages Python, Dash, and Plotly to provide an interactive, real-time, and customizable solution. Unlike costly commercial tools, it enables seamless data integration, dynamic filtering, and automated report generation. By reducing manual effort and enhancing decision-making, this project streamlines test data management while addressing challenges like performance and security. Future enhancements may include AI-driven analytics for improved test efficiency, making it a powerful tool for data-driven teams.

3. METHODOLOGY

The Test Data Reporting and Visualization Tool follows a structured approach, starting with requirement analysis to define authentication, dataset upload, filtering, and report generation. It uses Dash, Plotly, Pandas, and fpdf2 for interactive visualizations and PDF exports. Users upload Excel test reports, which are processed with Pandas, enabling dynamic filtering and owner-based reporting. Plotly generates real-time dashboards, updating visualizations based on user inputs. The tool allows PDF export of reports with graphs and summaries. After testing and optimization, it is deployed and refined based on user feedback, ensuring efficiency and scalability.

4. RESEARCH DESIGN

The research design for the Test Data Reporting and Visualization Tool follows a design and development approach, focusing on creating an efficient reporting system using Dash, Plotly, and Pandas. The tool processes Excel-based test reports, enabling data filtering, visualization, and report generation. It follows a modular architecture, integrating user authentication, dataset upload, interactive dashboards, and PDF export functionality. The development follows an iterative process, ensuring real-time updates and seamless user interaction. Functional and performance testing validate its accuracy, scalability, and efficiency in handling large datasets. The outcome is an automated, user-friendly tool that enhances decision-making by transforming raw test data into meaningful insights.

5. TOOLS AND TECHNIQUES

The development of the Test Tool Reporting and Visualization Tool utilized a streamlined technology stack centered around Dash and Plotly to deliver an interactive, user-friendly platform. Dash was used for the frontend, enabling the creation of dynamic, responsive user interfaces for tasks like uploading datasets, applying filters, and displaying visual reports. Plotly was leveraged for data visualization, creating interactive charts to analyze and present test results. Pandas was employed to process and filter the uploaded datasets, ensuring efficient data handling. Dash Auth was integrated for secure user authentication, ensuring only authorized access to the tool. This combination of technologies provided a seamless, interactive experience for test report generation and visualization, along with secure login and efficient data processing.

6. PROCEDURE

Planning and Requirement Analysis:

- Focused on gathering insights from project stakeholders to identify key requirements such as interactive dashboards, data filtering options, and secure user authentication. Defined the scope, technical specifications, and project objectives for the Test Tool Reporting and Visualization Tool.

System Design:

- Created wireframes and UI/UX mockups using tools like Figma to visualize the user flow and layout of the platform.
- Designed the structure for handling data, including the process of uploading and filtering datasets, and visualizing test reports.

Backend Development:

- Set up Dash for building the backend logic and server-side functionalities.
- Implemented secure user authentication with Dash Auth and integrated data management, including reading, filtering, and processing datasets using Pandas.

Frontend Development:

- Developed responsive UI with Dash components.
- Integrate frontend with backend using Dash callbacks

Testing and Debugging:

- Conducted usability testing with potential users, including stu testers, to ensure the platform met their needs and provided an intuitive experience.
- Performed functional testing to check the accuracy of data processing, filtering, and report generation.

Deployment:

- Ensured the platform was set up for smooth deployment, with a focus on security, scalability, and high performance for handling large datasets and user interactions.

7. EXPERIMENTAL RESULT

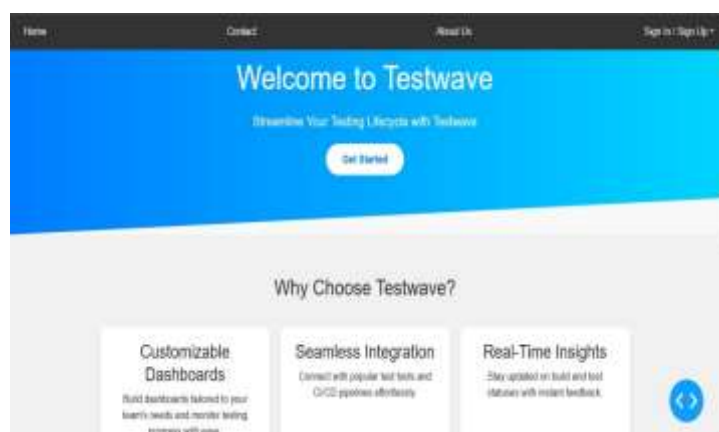


Fig : 7.1 Home Page



Fig: 7.2 Login Page

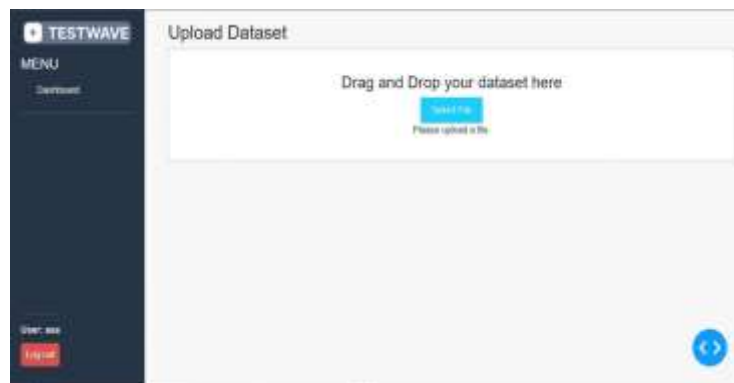


Fig: 7.3 Dataset Upload Page



Fig: 7.4 Dashboard Page



Fig: 7.5 Dashboard after applying filter

8. RESULTS

The development and implementation of the "Test Tool Reporting and Visualization Tool led to valuable outcomes in terms of usability, data management, and user satisfaction. Key findings from this project include:

1. **User Engagement:** The platform effectively enabled users to interact with datasets, apply filters, and visualize test results, leading to high engagement from users.
2. **Data Handling Efficiency:** The integration of Dash and Plotly ensured seamless data processing, filtering, and dynamic report generation, improving workflow efficiency.
3. **Visualization Impact:** Interactive charts and dashboards increased user understanding of test results and trends, enhancing decision-making capabilities.
4. **Usability:** The platform's intuitive design and responsive interface made it easy for users to upload datasets, apply filters, and generate reports without technical barriers.
5. **Overall Impact:** The tool successfully streamlined test data reporting and visualization, empowering users to quickly analyze and export reports, boosting productivity and decision-making.

9. CONCLUSION

In conclusion, the Test Tool Reporting and Visualization Tool successfully achieved its goal of providing an efficient, interactive platform for test data reporting and analysis. By leveraging Dash and Plotly, the tool allowed users to easily upload datasets, apply filters, and visualize test results through dynamic dashboards. The platform's intuitive design and secure user authentication ensured a smooth user experience, while the ability to export reports in PDF format added significant value. Overall, the tool streamlined the test reporting process, improved data analysis capabilities, and enhanced decision-making, making it a valuable resource for users seeking to manage and visualize test data effectively.

10. REFERENCES

- [1] Dash, Plotly. Dash User Guide: Interactive Web Apps for Data Visualization. 2023.
- [2] Plotly. Plotly Python Open-Source Graphing Library. 2023.
- [3] Pandas. Python Data Analysis Library Documentation. 2023.
- [4] Towards Data Science. Using Dash for Data Visualization in Python. 2023.
- [5] Faculty. Dash Bootstrap Components: User Guide. 2023