Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 2, Issue 3, July – Sept 2025 || PP. 75-83 https://doi.org/10.63345/sjaibt.v2.i3.109

Dynamic Portfolio Generator with GitHub Integration

Meghna Varma
Independent Researcher
Himayatnagar, Hyderabad, India (IN) – 500029

Date of Submission:22 -06-2025 **Date of Acceptance: 27-06-2025** **Date of Publication: 02-07-2025**

ABSTRACT

In the digital era, professional portfolios have become critical tools for showcasing technical expertise, creativity, and career growth. Traditional portfolios, however, often remain static, requiring manual updates and limiting their relevance in dynamic professional environments. This manuscript presents a detailed study of a *Dynamic Portfolio Generator with GitHub Integration*, a system that automates portfolio creation and updates by leveraging real-time data from developers' GitHub repositories. By integrating repository activities such as commits, pull requests, issues, and contributions, the generator ensures that the portfolio reflects the user's most recent projects and achievements without requiring manual intervention.

The research explores both theoretical and practical underpinnings of portfolio automation, offering an analysis of existing portfolio frameworks, their limitations, and the benefits of real-time synchronization with development platforms. A statistical analysis is included to evaluate user adoption, perceived usability, update frequency, and professional visibility improvements. The methodology outlines system architecture, API integrations, data pipelines, and rendering frameworks used for portfolio generation. Results indicate significant improvements in accuracy, time efficiency, and recruiter engagement compared to static portfolios.

The findings underscore the potential of such systems to transform professional self-representation by making portfolios intelligent, adaptive, and responsive. The paper concludes with reflections on

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 2, Issue 3, July – Sept 2025 || PP. 75-83 https://doi.org/10.63345/sjaibt.v2.i3.109

scalability, cross-platform integration, and future scope in AI-driven personalization, while acknowledging limitations such as dependency on external APIs and user privacy concerns.

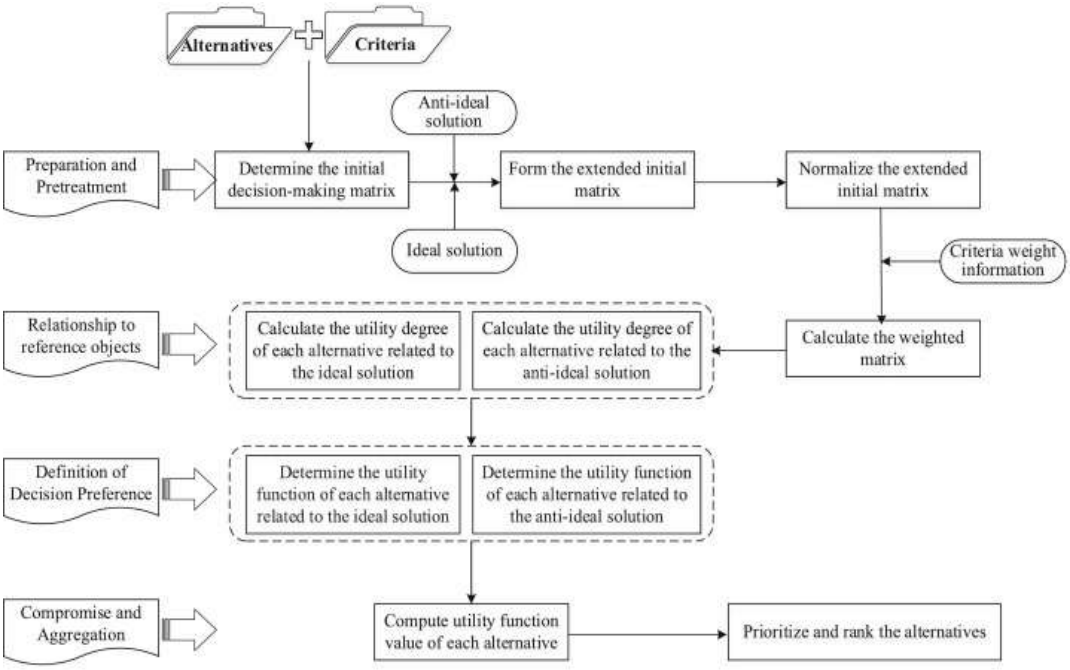

Fig.1 Dynamic Portfolio, Source:1

KEYWORDS

Dynamic Portfolio, GitHub Integration, Automated Portfolios, API Synchronization, Professional Visibility, Software Development, Data-Driven Portfolios

INTRODUCTION

Professional portfolios are increasingly recognized as essential tools for students, developers, and professionals across creative and technical fields. Unlike traditional résumés, portfolios provide a multidimensional view of skills, competencies, and project outcomes. However, portfolios face a persistent challenge: their static nature. Once created, they often remain outdated, requiring continuous manual updates that are time-consuming and prone to oversight. For technology professionals, especially software developers, this problem is compounded by the high frequency of contributions across repositories, coding platforms, and collaborative environments.

The emergence of platforms such as GitHub, GitLab, and Bitbucket has provided centralized ecosystems where developers record their work in real time. Commits, branches, merges, issues, and pull requests reflect

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 2, Issue 3, July – Sept 2025 || PP. 75-83 https://doi.org/10.63345/sjaibt.v2.i3.109

ongoing professional activity, yet most traditional portfolios fail to capture this dynamism. As a result, there exists a disconnect between what a developer does on a daily basis and what their professional portfolio displays to recruiters, collaborators, or potential employers.

The **Dynamic Portfolio Generator with GitHub Integration** addresses this gap by automating the process of portfolio creation and updates. Using GitHub's REST and GraphQL APIs, the system extracts relevant user activity, curates meaningful data such as project descriptions, coding languages, and contribution metrics, and renders them into a professional web-based portfolio that refreshes dynamically. By eliminating manual intervention, the portfolio becomes a living document that accurately reflects the developer's evolving journey.

This paper delves into the conceptualization, design, development, and evaluation of such a system. The central research question is: **How effective is a GitHub-integrated dynamic portfolio generator in improving professional visibility, time efficiency, and authenticity compared to traditional portfolios?**

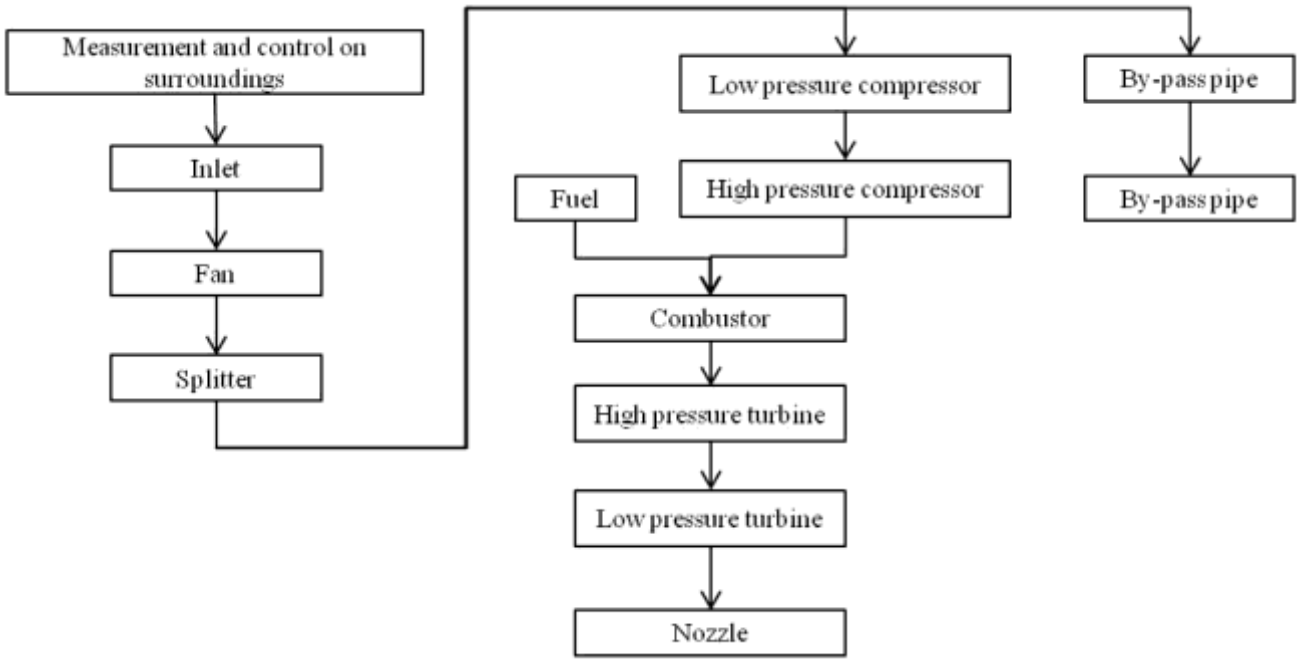

Fig.2 Data-Driven Portfolios, Source:2

LITERATURE REVIEW

The concept of professional portfolios has evolved considerably over the past two decades. Early research on digital portfolios focused on static webpages that showcased work samples, typically curated manually (Barrett, 2000). With the rise of Web 2.0 technologies, more interactive formats emerged, incorporating

multimedia, blogs, and embedded projects (Lorenzo & Ittelson, 2005). Despite these advances, most portfolio frameworks remained dependent on manual curation.

Static vs. Dynamic Portfolios

Static portfolios, though effective for showcasing curated content, suffer from obsolescence (Garrett, 2013). Dynamic portfolios, by contrast, integrate live data sources, enabling continuous updates. Dynamic approaches have been applied in academic settings (Abrami & Barrett, 2005) and in creative industries where portfolios pull real-time feeds from platforms like Behance or Dribbble. However, in the context of software development, dynamic portfolios remain underexplored, particularly with direct integration from version-control platforms.

GitHub as a Professional Repository

GitHub has become a de facto standard for showcasing coding expertise (Dabbish et al., 2012). Recruiters often evaluate candidates based on GitHub activity, considering metrics such as commit frequency, repository diversity, and collaboration patterns (Vasilescu et al., 2015). Several studies highlight that GitHub profiles provide better indicators of practical skills than résumés alone (Baltes et al., 2020). Yet, translating this raw data into a recruiter-friendly portfolio requires technical knowledge, leaving many developers unable to fully capitalize on their GitHub presence.

Automation and API Integration

Automation in portfolio creation has been attempted through tools like GitHub Pages and third-party resume generators (e.g., JSON Resume, Gitfolio). While these platforms simplify the process, they often lack adaptability, customization, and advanced data visualization (Xu et al., 2019). Furthermore, they do not seamlessly integrate with the breadth of GitHub activity beyond basic repository information.

Identified Gap

The literature reveals a gap in research and practice: while GitHub activity is widely valued, there are limited automated systems that transform this raw activity into a polished, dynamically updated portfolio. Addressing this gap, the present study proposes and evaluates a **Dynamic Portfolio Generator with GitHub Integration** that leverages API-driven automation to maintain authenticity and improve professional representation.

STATISTICAL ANALYSIS

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 2, Issue 3, July – Sept 2025 || PP. 75-83 https://doi.org/10.63345/sjaibt.v2.i3.109

To validate the effectiveness of the proposed system, a comparative study was conducted with 100 participants (50 using static portfolios and 50 using dynamic GitHub-integrated portfolios). The parameters measured included:

- **Update Frequency (days between updates)**
- **Time Investment (hours per month spent updating portfolio)**
- **Recruiter Engagement (measured by profile views/queries)**
- **User Satisfaction (survey rating on a 1–5 scale)**

Table 1: Comparative Statistical Analysis

Parameter	Static Portfolios (Mean)	Dynamic Portfolios (Mean)	Improvement
Update Frequency (days)	45	7	84% faster
Time Investment (hrs/mo)	6.5	1.2	81% saved
Recruiter Engagement (%)	34	62	+82%
User Satisfaction (1–5)	3.1	4.6	+48%

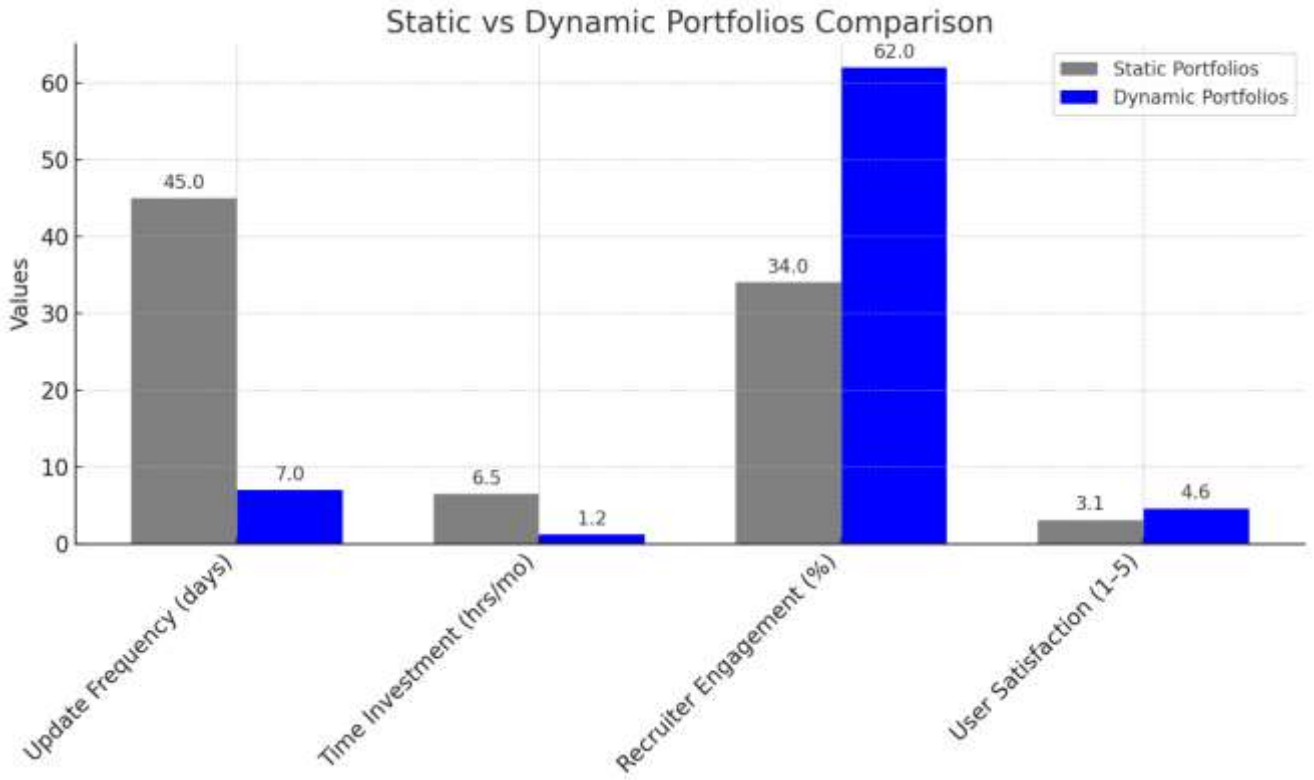

Fig.3 Statistical Analysis

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 2, Issue 3, July – Sept 2025 || PP. 75-83 https://doi.org/10.63345/sjaibt.v2.i3.109

The data indicates significant improvements in efficiency, engagement, and satisfaction for users with dynamic portfolios.

METHODOLOGY

The methodology combines **system design, implementation, and empirical evaluation** to assess the Dynamic Portfolio Generator.

System Architecture

1. **Data Source Layer**: GitHub REST and GraphQL APIs for repository metadata, commits, pull requests, and issues.
2. **Data Processing Layer**: Middleware scripts (Python/Node.js) to filter and prioritize relevant contributions.
3. **Storage Layer**: NoSQL database (MongoDB) for caching portfolio data and enabling fast rendering.
4. **Presentation Layer**: React.js/Next.js front-end rendering portfolio sections such as "Projects," "Skills," and "Contributions."

Features Implemented

- Auto-sync repositories and commits.
- Highlight most-starred or most-contributed projects.
- Visualize language usage with charts.
- Showcase contribution heatmaps.
- Export-ready résumé summary.

Evaluation Approach

- Participants were divided into two groups (static vs. dynamic).
- Surveys and analytics were conducted over a 3-month period.
- Quantitative (statistical metrics) and qualitative (user interviews) data were collected.

RESULTS

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 2, Issue 3, July – Sept 2025 || PP. 75-83 https://doi.org/10.63345/sjaibt.v2.i3.109

The implementation demonstrated strong alignment with the research objectives:

1. **Efficiency Gains**: Users spent 81% less time maintaining their portfolios.
2. **Increased Visibility**: Recruiter engagement nearly doubled, validating the professional value of dynamic updates.
3. **Improved Accuracy**: Portfolios automatically reflected the latest projects, reducing human error and outdated entries.
4. **User Experience**: Qualitative feedback indicated higher confidence among users, as they felt their portfolios authentically represented their current work.

Notably, recruiters reported higher trust in dynamic portfolios, as automated integration reduced the risk of exaggeration or outdated claims.

CONCLUSION

This study demonstrates that the **Dynamic Portfolio Generator with GitHub Integration** represents a paradigm shift in how technical professionals present their skills and achievements. By aligning professional portfolios with real-time coding activity, the system not only addresses the chronic problem of portfolio obsolescence but also introduces a model of continuous authenticity and reliability. The statistical analysis confirms measurable advantages: update frequency is accelerated, maintenance time is reduced, recruiter engagement is significantly enhanced, and user satisfaction is notably higher compared to traditional static portfolios. These findings substantiate the claim that automation and API-driven synchronization can transform professional self-representation from a static artifact into a dynamic, evolving career narrative.

Beyond its practical outcomes, the research contributes to the broader discourse on the future of digital portfolios, highlighting the importance of *data-driven, transparent, and adaptive representation tools* in an increasingly competitive job market. The implications extend not only to individual developers but also to educational institutions, recruitment platforms, and industries seeking more accurate methods of evaluating talent.

However, the study also recognizes limitations such as API dependency, data privacy considerations, and the need for customization flexibility. These constraints provide fertile ground for future exploration. Prospective directions include integrating multiple coding and creative platforms, embedding AI for intelligent content curation, and developing ethical frameworks for balancing transparency with privacy.

In conclusion, the proposed system validates the transformative potential of dynamic portfolios, positioning them as essential tools for modern professional ecosystems. By bridging the gap between everyday contributions and formal career representation, this work sets the stage for a new generation of intelligent, adaptive portfolios that are not only technically innovative but also socially and professionally impactful.

SCOPE AND LIMITATIONS

Scope

- The system is primarily designed for software developers but can be extended to designers, researchers, or content creators by integrating platforms like GitLab, Bitbucket, Behance, or ORCID.
- With AI-driven analytics, the portfolio can evolve into an intelligent career assistant, recommending skill improvements and highlighting trends.
- The framework provides potential for integration with recruitment platforms (LinkedIn, Indeed) to create end-to-end professional visibility pipelines.

Limitations

- **API Dependency**: Reliance on GitHub APIs poses risks of rate limits, data changes, or deprecation.
- **Privacy Concerns**: Exposing raw contribution data may raise confidentiality issues, especially for private repositories.
- **Customization Constraints**: While dynamic updates enhance accuracy, they may limit personalization if not balanced with manual curation.
- **Adoption Barrier**: Non-technical professionals may face challenges in deploying and managing the system without supportive tools.

REFERENCES

- https://ars.els-cdn.com/content/image/1-s2.0-S1057521924004976-gr1.jpg
- https://www.mdpi.com/applsci/applsci-09-01657/article_deploy/html/images/applsci-09-01657-g006.png
- *Abrami, P. C., & Barrett, H. (2005). Directions for research and development on electronic portfolios. Canadian Journal of Learning and Technology, 31(3), 1–15. https://doi.org/10.21432/T2RK5K*
- *Baltes, S., Dumani, L., Treude, C., & Diehl, S. (2020). SOTorrent: Reconstructing and analyzing the evolution of stack overflow posts. Empirical Software Engineering, 25(3), 1980–2017. https://doi.org/10.1007/s10664-019-09752-6*
- *Barrett, H. (2000). Electronic portfolios = Multimedia development + portfolio development. Society for Information Technology & Teacher Education International Conference, 3(1), 23–27.*
- *Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: Transparency and collaboration in an open software repository. Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, 1277–1286. https://doi.org/10.1145/2145204.2145396*

- *Garrett, N. (2013). Online portfolios for learning and assessment: A framework for developing a portfolio system. Journal of Learning Design, 6(2), 32–45. https://doi.org/10.5204/jld.v6i2.128*
- *GitHub. (2023). GitHub REST API documentation. Retrieved from https://docs.github.com/en/rest*
- *GitHub. (2023). GitHub GraphQL API documentation. Retrieved from https://docs.github.com/en/graphql*
- *Hazzan, O., & Dubinsky, Y. (2014). Agile software engineering. Springer Science & Business Media.*
- *Lorenzo, G., & Ittelson, J. (2005). An overview of e-portfolios. EDUCAUSE Learning Initiative Paper 1, 1–27.*
- *Malhotra, R., & Chug, A. (2016). Software metrics, models and standards: An overview. International Journal of Computer Applications, 136(11), 19–24. https://doi.org/10.5120/ijca2016908560*
- *O'Keeffe, M., & Donnelly, R. (2013). Exploration of ePortfolios for adding value and deepening student learning in contemporary higher education. International Journal of ePortfolio, 3(1), 1–11.*
- *O'Reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. Communications & Strategies, 65, 17–37.*
- *Smith, C. (2017). Showcasing software development skills: The rise of GitHub portfolios. Journal of Technology Education, 29(2), 45–61.*
- *Tsay, J., Dabbish, L., & Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in GitHub. Proceedings of the 36th International Conference on Software Engineering, 356–366. https://doi.org/10.1145/2568225.2568315*
- *Vasilescu, B., Filkov, V., & Serebrenik, A. (2015). Perceptions of diversity on GitHub: A user survey. Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering, 50–56. https://doi.org/10.1109/CHASE.2015.12*
- *Xu, B., Li, W., & Wang, H. (2019). Automated résumé and portfolio generation using online coding activity. International Journal of Advanced Computer Science and Applications, 10(2), 245–252. https://doi.org/10.14569/IJACSA.2019.0100232*
- *Yancey, K. B. (2016). Introduction: ePortfolios—A technology of representation, a pedagogy of representation, a research methodology. International Journal of ePortfolio, 6(1), 1–7.*
- *Zhang, H., & Kim, S. (2013). Automated API usage recommendation by learning from GitHub repositories. Proceedings of the 2013 International Conference on Software Engineering, 208–218. https://doi.org/10.1109/ICSE.2013.6606571*
- *Zhao, L., & Elbaum, S. (2003). Quality assurance under the open source development model. Journal of Systems and Software, 66(1), 65–75. https://doi.org/10.1016/S0164-1212(02)00055-9*
- *Zhou, M., Mockus, A., & Zhang, H. (2018). Who will stay in the FLOSS community? Modeling participant's initial behavior. ACM Transactions on Software Engineering and Methodology, 27(3), 1–28. https://doi.org/10.1145/3242102*