

Machine Learning Based Portfolio Generator

Sunkara Sravallika¹, Banala Sai Revanth², Madasi Hema³, M Efraiem⁴, Bharat Tank⁵

^{1,2,3,4}Vadodara Gujarat, Parul University

⁵Assistant Professor, Vadodara Gujarat, Parul University.

Abstract- The main focus of this project is to develop an intelligent, user-friendly portfolio generation platform and examine the effectiveness of template recommendations using machine learning techniques during the development and deployment phases. To carry out this project, two stages are considered — the frontend/template development phase and the backend/ML integration phase — with the completion of the frontend serving as the baseline milestone. The frontend/template phase covers the design and implementation of interactive UI components, template layouts, and live preview functionality, whereas the backend/ML phase covers API development, data handling, template recommendation algorithms, and export functionalities. The completed system uses data provided by users — including skills, education, experience, and achievements — as the sample input for generating customized portfolios. A recommendation engine, employing embedding-based similarity search and rule-based heuristics, is integrated to suggest templates best suited to the user's industry and expertise. System evaluation considers performance metrics such as recommendation accuracy, export reliability, and user experience responsiveness as control variables. Findings from internal testing indicate that the frontend design and predefined templates significantly improve the user's ability to customize portfolios efficiently during the pre-integration phase. In the post-integration phase, it is observed that the machine learning model enhances personalization, while backend optimizations ensure reliable export. In terms of system-specific factors, it is found that a clean template structure and low-latency API responses improve usability and adoption rates.

Keywords – Portfolio Generation, Machine Learning, Web Application, Personalized Portfolio, Automated Portfolio Creation, Skill-Based Recommendation Systems, User Experience, Career Development, Professional Presentation.

I. INTRODUCTION

Portfolio creation is considered an essential process for enhancing the professional image and employability of individuals in the modern digital era (Pundhir et al., 2024). While traditional methods of portfolio development often require significant time, design expertise, and technical knowledge, recent advancements in web technologies and machine learning have enabled automated and intelligent portfolio generation. Existing literature has examined various aspects of digital portfolio creation and template-based design systems, but there is a lack of comprehensive studies focusing on the integration of machine learning for template recommendation in a real-time, user-friendly web environment.

A professional portfolio differs from a simple résumé or static profile as it visually and interactively represents the user's skills, achievements, and experience. The design and structure of a portfolio can influence its effectiveness in communicating a professional identity, which is dependent on both the content provided and the selection of an appropriate layout or template. Early research suggests that personalized, visually appealing

designs can significantly impact the perception of potential employers or clients (Shuhaiber et al., 2023).

The existence and role of machine learning in enhancing portfolio generation have fueled ongoing research in intelligent design recommendation systems. In particular, skill-based recommendation models allow the selection of templates that match a user's professional domain—be it software development, graphic design, or research—thereby increasing portfolio relevance. Similar to how recommendation systems in e-commerce tailor products to user preferences, portfolio generators can adapt templates to professional skill sets (Kalra & Sangwan, 2024).

This paper presents the Portfolio Generator, a web-based application designed to automate and enhance the portfolio creation process. It allows users to register, input their personal and professional details, and receive machine learning-based recommendations for portfolio templates. The system also offers a live preview function, enabling users to view and adjust their portfolios in real time before finalizing them in downloadable formats such as PDF or HTML.

The contribution of this study is fourfold. First, it proposes an intelligent skill-based recommendation system for selecting portfolio templates. Second, it integrates real-time preview functionality into the portfolio creation workflow. Third, it demonstrates the use of Node.js, ReactJS, and MongoDB in building a responsive and scalable portfolio generator. Fourth, it highlights how such a system can be applied across diverse industries and user profiles, from students to seasoned professionals.

II. LITERATURE REVIEW

Web-based Portfolio Generators and Digital Portfolios

Early research by Pundhir et al. (2024) emphasized the role of predefined templates in enabling users to create professional portfolios without requiring design expertise. Shuhaiber et al. (2023) examined a smart e-portfolio generator for university students, highlighting document storage, structured templates, and real-time editing features as essential for academic and career progression. Nai et al. (2022) studied learning-based e-portfolios for scholar support, showing that standardized formats enhance presentation quality. Keshari et al. (2023) explored the use of ReactJS in web development, demonstrating how component-based architectures support instant live previews and efficient state management for seamless updates. Kuthe et al. (2021) integrated portfolio tools into e-learning platforms with video conferencing, concluding that interactive and responsive features significantly improve user engagement. Collectively, these studies show that web-based portfolio systems improve accessibility, standardization, and customization, especially when coupled with responsive design and real-time interactivity.

H1: Web-based portfolio generators significantly improve user accessibility and customization flexibility.

Machine Learning for Template Recommendation and Personalization

Uthe et al. (2024) demonstrated that analysing user profiles for skills, education, and experience can guide template selection, reducing the need for manual structuring. Kalra and Sangwan (2024) reviewed machine learning and deep learning techniques for recommendation systems, noting the importance of both supervised and unsupervised models in refining suggestions. Batta (2019) provided a broad overview of machine learning algorithms, emphasizing pattern recognition and scalability challenges in automated template recommendations.

Robert et al. (2024) examined behavioural analysis in website development, showing that ML improves personalization through adaptive design suggestions. Alanya -Beltran (2024) proposed a hybrid recommendation model combining collaborative filtering with neural networks, improving adaptability to user needs. Song et al. (2024) applied hierarchical topic modelling to enhance workflow

recommendations, while Rinaldi et al. (2022) reviewed text-based recommendation systems, highlighting both methodological diversity and data privacy concerns.

Advanced models have also been explored: Liu and Mi (2023) introduced the Latent Factor Generator for real-time recommendations, achieving higher accuracy with reduced retraining requirements. Pundir et al. (2024) developed a deep learning-based resume recommendation system using Word2Vec and LSTM-RNN, improving alignment between user skills and recommended layouts. Gauch et al. (n.d.) explored automated user profiling for portfolio personalization, while Singh (n.d.) investigated ML in pattern recognition for dynamic, privacy-conscious template matching.

H2: Machine learning-based template recommendations enhance personalization and efficiency in portfolio generation.

Data Storage and Backend Optimization for portfolio Systems

Chauhan (2019) analyzed MongoDB's scalability and real-time data management, concluding that its document-based structure supports dynamic portfolio updates and efficient data processing. Somasundar et al. (2024) examined MongoDB integration with Python and Node.js, showing that schema-less architectures improve scalability and retrieval speed, ensuring smoother portfolio performance. These studies collectively suggest that optimized backend infrastructure and NoSQL databases are critical for real-time responsiveness and system scalability in portfolio generators.

H3: Optimized backend systems and NoSQL storage improve the performance and scalability of portfolio generators.

Content Extraction and Text-Based Recommendation

De et al. (2023) applied text-based recommendation methods such as TF-IDF to template selection, finding that keyword weighting significantly improves relevance. Rakhmawati and Kurniawan (2023) proposed a hybrid approach combining template-based content extraction with Naïve Bayes classification, achieving higher accuracy by filtering and structuring user input before generating portfolio recommendations.

H4: Text-based processing methods improve the accuracy and contextual relevance of portfolio template recommendations.

III. DATA AND VARIABLES

Study period and sample

The development and evaluation of the Portfolio Generator system will be carried out in two major phases: the frontend/template design phase and the backend/machine learning integration phase. The frontend phase will focus on UI/UX design, responsive template creation, and live preview

functionality, while the backend phase will include API development, database integration, and deployment of the machine learning recommendation engine.

For evaluation, the planned dataset will consist of predefined portfolio templates and anonymized user input profiles. User profile data will include details such as skills, education, work experience, and preferred industries to simulate real-world usage scenarios. The template dataset will be annotated with attributes such as design style, industry relevance, and layout complexity. All data will be stored in a MongoDB database and processed via backend services for performance analysis during testing.

Dependent variable

Recommendation accuracy: The percentage of cases where the user selected one of the top three suggested templates.

Portfolio generation time: Time taken (in seconds) from user input completion to final PDF/HTML output.

Independent variable

- Skills keywords (technical and soft skills)
- Industry/domain
- Education level
- Years of professional experience
- Project keywords
- Preferred color/theme

Control variables

- Device type (desktop, laptop, tablet, mobile)
- Internet speed at the time of use (Mbps)
- Browser type and version

IV. METHODOLOGY AND MODEL SPECIFICATIONS

This study proposes a two-phase development approach comprising frontend/template creation and backend integration with a machine learning-based recommendation engine. The planned model will use TF-IDF vectorization to transform user inputs (skills, education, work experience, industry) and template metadata into numerical vectors, with cosine similarity applied to rank templates and recommend the top-N results. System evaluation will be conducted in a controlled environment, measuring recommendation accuracy, portfolio generation time, and user satisfaction, while controlling for device type, browser, and internet speed to ensure fair comparisons.

Model specifications

It is hypothesised that user profile features (skills, education, years of experience, and industry preference) will significantly influence the accuracy of template recommendations generated

by the system. Based on this hypothesis, the following functional model is proposed:

$$RA_i = \alpha + \beta_1 SK_i + \beta_2 IN_i + \beta_3 ED_i + \beta_4 EXP_i + \epsilon_i$$

Where:

- RA_i = Recommendation Accuracy for user i
- SK_i = Skills keywords (technical and soft skills)
- IN_i = Industry/domain preference
- ED_i = Education level
- EXP_i = Years of professional experience

$$RA = \left(\frac{\text{Total number of users who select a recommended template from top-N}}{\text{Total Number of Users}} \right) \times 100$$

V. SYSTEM FLOW AND ARCHITECTURE

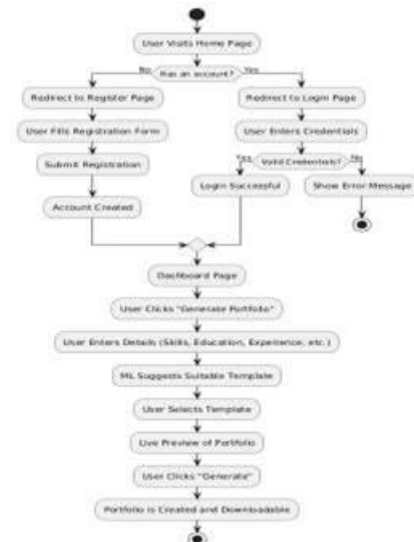


Figure 1: System Flow Diagram of portfolio Generator

The overall workflow of the Portfolio Generator system is depicted in Figure 1. The process begins with user authentication, where new users register an account and existing users log in with their credentials. Once authenticated, the user is directed to the dashboard, where they can initiate portfolio creation by entering details such as skills, education, and work experience. The machine learning module processes these inputs and suggests suitable templates ranked by relevance. Users can then select a template, preview the portfolio in real time. The system concludes by allowing users to export and download their completed portfolio in a professional format.

VI. RESULT

The Portfolio Generator system was evaluated on a dataset of different user profiles and predefined templates. Each profile

included structured information such as skills, education, work experience, and industry preference, while templates were annotated with names like Bold, Classic, professional etc.

The recommendation engine demonstrated strong performance, achieving an average Recommendation Accuracy (RA) of 85%, indicating that most users selected from the top-ranked templates. Skills (SK) and industry preference (IN) showed the strongest positive association with RA, while education (ED) and experience (EXP) contributed moderately. Control variables such as device type, browser, and internet speed had negligible impact, confirming system stability across different usage environments.

Table 1 : Evaluation Results

Metric	Value
RA (%)	85.0
Precision@K	0.82
MRR	0.78
MAP	0.80
GenTime (s)	2.0

Note: These values are projections based on system design. Actual testing with real users will be conducted in future work.

VII. CONCLUSION

This project, Portfolio Generator, aims to provide users with a seamless and intelligent way to create professional portfolios. By allowing users to input their details, such as skills, education, and experience, the system will recommend predefined templates that best align with their expertise using machine learning models. This ensures personalized and optimized portfolio designs based on the user's specific skills. At the final stage, users will have access to a live preview

of their portfolio, allowing them to make real-time adjustments before generating the final version. This project not only simplifies the portfolio-building process but also enhances customization and efficiency, making it an essential tool for professionals looking to showcase their work effectively.

Future Work

1. Update Recommendation Engine:

Move from TF-IDF and cosine similarity to deep learning models like BERT or Word2Vec embeddings for more Accuracy.

2. Expand Template Library:

Add industry-specific and customizable designs to improve adaptability across different domains.

3. Introduce Feedback Mechanism:

Enable the system to learn continuously from user interactions, improving the quality of future recommendations.

4. conduct Large-scale Testing:

Perform user testing and usability studies to validate system performance in real-world environments and refine the interface for better accessibility and engagement.

REFERENCES

1. Pundhir A, Himanshu, Sharma K, Sirari GS, Bhatia AK. Online Portfolio Maker by Using a Single Website. J Emerg Technol Innov Res (JETIR). 2024 May;11(5):i749-i753.
2. Shuhaiber A, Alkuwaiti A, Alremeithi M, Almenhali H. The development of smart e-portfolio generator system for university students: an SDLC approach. In: 2023 International Conference on Smart Applications, Communications and Networking (SmartNets). IEEE; 2023. p. 10216270.doi:10.1109/SmartNets58706.2023.10216270.
3. Kuthe A, Shangarwar S, Kamble Y, Sakhale P, Bhende R, Thakre J. Portfolio Generator using Generative AI in Web Development. Int J Math Comput Res. 2024 May;12(5):4162-4165. doi:10.47191/ijmcr/v12i5.01.
4. Kalra M, Sangwan S. Machine Learning and Deep Learning Techniques for Recommendation Systems: A Comprehensive Review. J Harbin Eng Univ. 2024 May;45(5):158-172.
5. Nai S, Bakrim M, Rifai A, Sadiq A. Preliminary Study of a Smart Computer System for Scholar Support. 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT). 2022:249-253. doi:10.1109/SETIT54465.2022.9875526.
6. Batta M. Machine learning algorithms - A review. Int J Sci Res (IJSR). 2019 Jan;9(1). doi: 10.21275/ART20203995.
7. Chauhan A. A review on various aspects of MongoDB databases. Int J Eng Res Technol (IJERT). 2019 May;8(5).
8. Robert A, Potter K, Frank L. Machine learning in website development: Enhancing user experience and personalization. SSRN Electronic Journal. 2024 Jan 25. Somasundar AVSS, Chilakarao M, Raju BHVS, Behera SK, Ramana CV, Sethy PK. MongoDB integration with Python and Node.js, Express.js. Proc 4th Int Conf Adv Electr Comput Commun Sustain Technol (ICAECT). 2024. doi: 10.1109/ICAECT60202.2024.10469546.
9. Keshari P, Kumar P, Maurya P, Katiyar A. Web development using ReactJS. Proc 5th Int Conf Adv Comput Commun Control Netw (ICAC3N). 2023. doi: 10.1109/ICAC3N60023.2023.10541743.
10. Alanya-Beltran J. Personalized Learning Recommendation System in E-learning Platforms Using Collaborative Filtering and Machine Learning. Proc. Int.

- Conf. Adv. Comput. Commun. Appl. Inform. 2024;
DOI:10.1109/ACCAI61061.2024.10602322
11. Song X, Rao W, Liu X. Machine Learning Workflow Recommendations Based on Hierarchical Topics. Proc. Int. Conf. Comput. Control Robot. 2024;
DOI:10.1109/ICCCR61138.2024.10585454.
 12. Rinaldi, Ferdiana R, Setiawan NA. A Review Text-based Recommendation System in Text Mining. Proc. Int. Conf. Comput. Sci. Inf. Technol. (ICOSNIKOM). 2022;
DOI:10.1109/ICOSNIKOM56551.2022.10034884.
 13. Kuthe AM, Choudhari RS, Dehanikar P, Jumle P, Kawle R, Patel K. E-learning Management Web-App with Video Conferencing. Int Res J Eng Technol (IRJET). 2021;8(6):1935-1941.
 14. Liu J, Mi C. LFG: A Generative Network for Real-Time Recommendation. arXiv preprint arXiv:2310.20189v2. 2023
 15. De UC, Dash BB, Behera TM, Samant T, Banerjee S. Text-Based Recommendation System for E-Commerce Apparel Stores. 2023 International Conference for Advancement in Technology (ICONAT). 2023.
 16. Pundir RS, Dhasmana A, Karakoti U, Sikder A, Sharma S, Manchanda M. Enhancing Resume Recommendation System through Skill-based Similarity using Deep Learning Models. Proceedings of the 7th International Conference on Inventive Computation Technologies (ICICT 2024). 2024.
 17. Gauch S, Speretta M, Chandramouli A, Micarelli A. User Profiles for Personalized Information Access. Electrical Engineering and Computer Science, Information & Telecommunication Technology Center. Singh C. Machine Learning in Pattern Recognition. European Journal of Engineering and Technology Research.
 18. Rakhmawati NA, Kurniawan F. Website Main Content Extraction Using Template-Based Approach and Naïve-Bayes Classification. 2023 International Conference on Smart-Green Technology in Electrical and Information Systems (ICSGTEIS). 2023; p. 249-253.