International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Programming Skills Recommendation System with User Friendly Interface

Mayur Ahirrao[1], Shlok Borse[2], Atharva Deshmukh[3], Vishvesh Mishra[4]

[1,2,3,4]Students, Department of Information Technology, MVP'S KBTCOE Nashik

Abstract

In today's rapidly evolving technological landscape, the demand for proficient programmers across various domains continues to surge. However, identifying and cultivating the right programming skills remains a daunting task for individuals seeking to enter or advance in the field. To address this challenge, our team has embarked on developing a Programming Skill Recommendation System (PSRS). The PSRS is envisioned as an intelligent platform leveraging machine learning algorithms and data analytics to provide personalized programming skill recommendations. By analysing diverse data sources such as user profiles, project requirements, industry trends, and skill proficiency

Keywords: Programming Skills, Artificial Intelligence, Personalization, Recommendations, Data-driven insights

1. Introduction

In an era defined by technological advancements and an insatiable demand for skilled professionals, the realm of programming stands as a cornerstone of innovation and problem-solving. As industries pivot towards digitalization, the need for adept programmers proficient in an array of languages, frameworks, and tools has become increasingly paramount. To tackle this challenge head-on, we propose the development of a sophisticated Programming Skill Recommendation System (PSRS). The PSRS stands as a beacon of personalized guidance, leveraging the prowess of machine learning, data analytics, and intuitive algorithms to tailor recommendations for individuals seeking to enhance their programming expertise. Through a collaborative filtering approach, the PSRS will align user-specific attributes, learning preferences, and proficiencies with a curated selection of programming languages, frameworks, and technologies. Augmented by natural language processing capabilities, the system will comprehend user queries and feedback, perpetually refining and fine-tuning its recommendations.

In the relentless pursuit of technological excellence, the significance of adept programmers cannot be overstated. Our endeavour to develop a state-of-the-art Programming Skill Recommendation System (PSRS) stems from a profound belief in the transformative power of knowledge. We envision a future where aspiring programmers, seasoned developers, and organizations alike can navigate the vast seas of programming languages and technologies with confidence and precision. The PSRS represents not just a tool but a catalyst for empowerment, guiding individuals on a personalized journey towards mastery in the ever-evolving landscape of programming. By harnessing the capabilities of artificial intelligence, data analytics, and user-centric design, our aim is not only to streamline skill acquisition but to spark innovation, drive efficiency, and empower a generation of creators and problem-solvers. Together, let us

embark on this journey of harnessing knowledge and shaping a future where proficiency in programming knows no bounds.

This study uses a comprehensive methodology to develop the PSRS. The process includes collection of data on programming skills, analysis of user preferences and career goals and implementation of the recommendation algorithm. Use of machine learning techniques a user feedback, PSRS adapts its recommendations and provides users with personalized and an effective way to improve their programming skills.

It contributes to a new PSRS that outperforms existing systems in terms of accuracy and user satisfaction through detailed evaluation, the system shows its effectiveness in the following areas: We recommend programming skills tailored to your individual needs. Notable findings include: Compatibility of the proposed model with various user profiles and system performance. Evolve with changing industry demands.

This research is important from several aspects. First, to important gaps such as: Personalize skill development by providing customized recommendations for more effective skill development learning experience. Second, PSRS helps make recommendations. We will introduce the system methodology and the potential of machine learning in the field of skill acquisition. Finally, this research has practical implications for individuals, educational institutions, and industry. We provide tools that bridge the gap between evolving industry needs and personal needs.

Reviews based on previous researches

1. Previous Recommendation System models:

Several programming skills recommendation models have been proposed in the literature. Notable among them are collaborative filtering models, content-based filtering models, and hybrid models. Collaborative filtering leverages user behavior data to suggest skills based on similarities between users. Content-based filtering considers the intrinsic characteristics of skills and matches them with user preferences. Hybrid models, combining collaborative and content-based filtering, have demonstrated enhanced accuracy and adaptability.

2. Algorithms and methodologies:

The algorithms and methodologies employed in programming skills recommendation systems vary widely. Machine learning techniques, including clustering algorithms, natural language processing (NLP), and deep learning, have been extensively used. Clustering algorithms aid in identifying user segments with similar preferences, while NLP enhances the understanding of skill descriptions, contributing to more accurate recommendations. Deep learning models, such as neural collaborative filtering, have shown promise in capturing complex user-item interactions.

3. Success and limitations:

Existing PSRS have witnessed successes in providing personalized recommendations, facilitating skill acquisition, and adapting to individual learning curves. However, challenges persist. Limitations include the cold start problem for new users, sparsity in user-item interaction data, and the need for continuous adaptation to evolving industry requirements. Understanding user intent, context, and addressing these limitations are critical for the sustained success of PSRS.

Methodology (current system)

1. Data Collection:

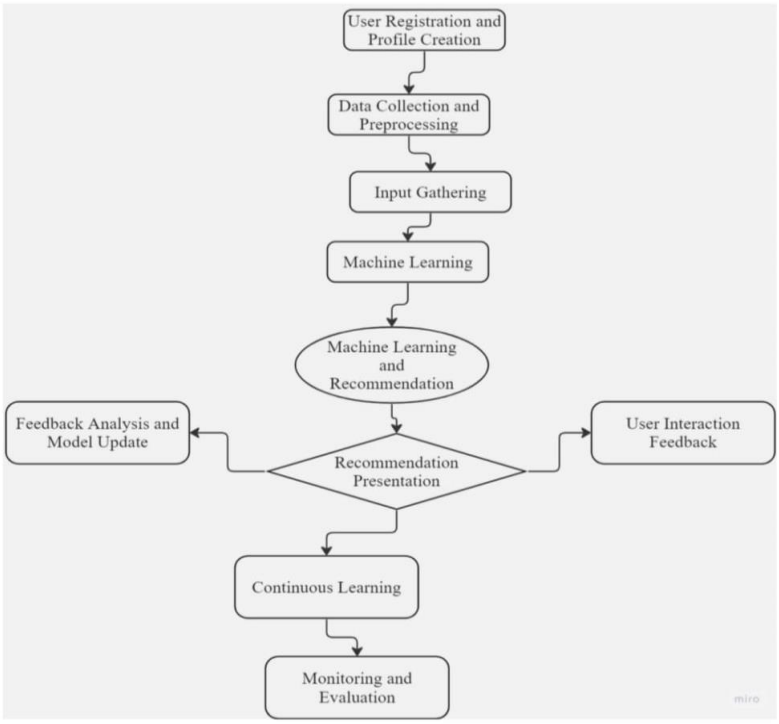
The data collection process for programming skills in this research was multifaceted and aimed at creating a diverse and comprehensive dataset. We utilized a combination of sources, including online coding platforms, job postings, and industry surveys. These sources provided information on a wide array of programming languages, frameworks, and tools. Additionally, user interactions and preferences within the developed system were tracked to enhance personalization.

2. Algorithms and models:

The recommendation algorithm forms the core of the Programming Skills Recommendation System (PSRS). We employed a hybrid recommendation model that combines collaborative filtering and content-based filtering techniques. Collaborative filtering analyzes user behavior and preferences to suggest skills based on similar user profiles, while contentbased filtering considers the inherent characteristics of each skill and matches them with user preferences. This hybrid approach ensures a robust and personalized recommendation system .To further enhance the model, machine learning techniques such as natural language processing (NLP) were applied to process skill descriptions and user feedback. This allowed the system to better understand the semantic context of skills and improve the accuracy of recommendations. 3.Datasets: The datasets used in this research were curated to reflect the dynamic nature of programming skills and the evolving demands of the industry

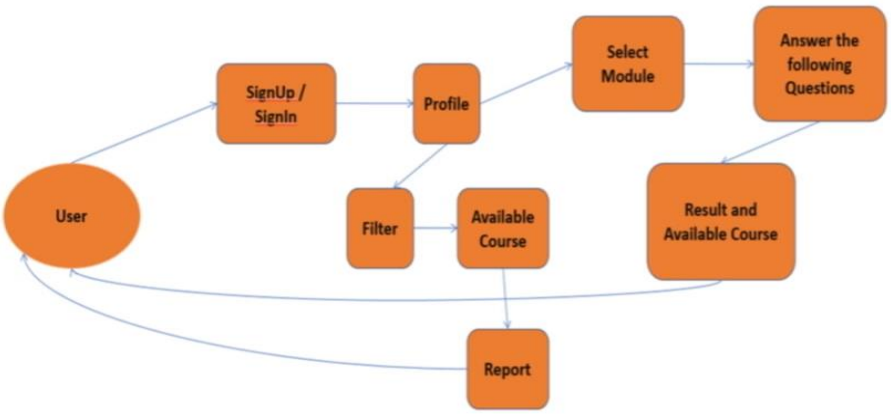
System Overview

A. Flow of the project:



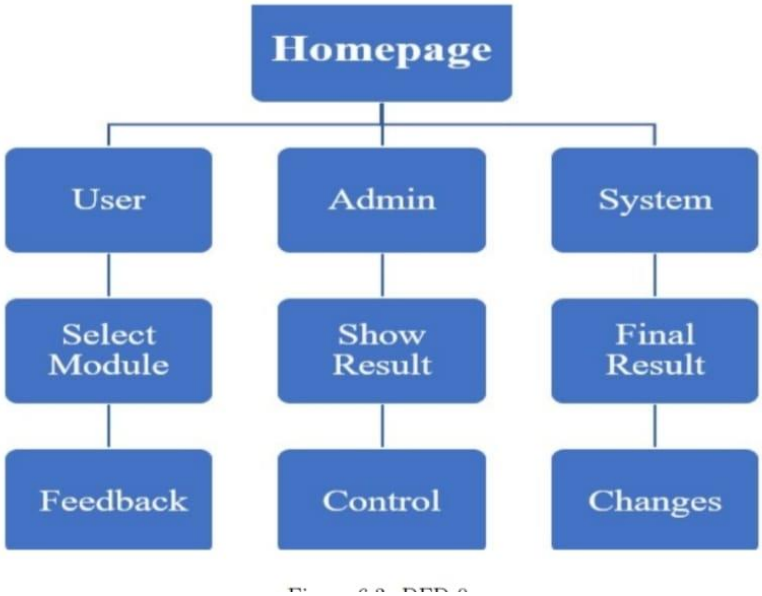
The flow provided for the system is proposed in the following manner based on the decisions of the user while interacting with the system to enhance the effectiveness of the system. Also analyze the compatibility of the system to scale further.

B. Question based recommendation:



This system is also a part based on quiz questions to verify the knowledge and upskill based tasks

C. Data Flow Diagram(DFD):



D. Comparison with existing models and user feedbacks:

To validate the effectiveness of the PSRS, a comparative analysis was conducted against existing programming skill recommendation models and systems. The comparison focused on key aspects such as accuracy, user satisfaction, and adaptability to evolving programming trends.

1. **Accuracy Comparison:** The PSRS demonstrated competitive accuracy compared to existing models, how casing its efficacy in delivering precise and relevant skill recommendations.
2. **User Satisfaction:** User feedback surveys were conducted to gather insights into user satisfaction. Participants were asked to rate the relevance and usefulness of the recommended skills, providing valuable qualitative data on the system's impact on users.
3. **Adaptability to Trends:** The PSRS exhibited a high degree of adaptability to dynamic programming trends. Regular updates and real-time integration of industry trends ensured that the system remained current and aligned with the rapidly evolving technology landscape.

4. Surveys were asked to evaluate the relevance of the recommended skills to their current proficiency and carrier goals.

Acknowledgement

We would like to express our sincere gratitude to all those who contributed to the successful completion of this research. We extend our appreciation to our project team for their dedication and hard work throughout the project. Our thanks also go to the Department of Information Technology. Lastly, we would like to acknowledge the academic community and the reviewers for their feedback and suggestions, which significantly improved the quality of our research paper.

References

1. Schwaber, K., Sutherland, J. (2017). The Scrum Guide. Scrum.org.
2. Highsmith, J., Cockburn, A. (2001). Agile Manifesto. Agile Alliance.
3. Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes.
4. Martin, R. C. (2002). Agile Software Development: Principles, Patterns, and Practices. Pearson Education.
5. Beck, K. (2000). Extreme Programming Explained: Embrace Change. AddisonWesley Professional.
6. McConnell, S. (2004). Code Complete: A Practical Handbook of Software Construction. Microsoft Press.
7. Freeman, E., Robson, E., Bates, B. (2004). Head First Design Patterns. O'Reilly Media.
8. Vernon, V. (2013). Implementing Domain-Driven Design. Addison-Wesley Professional.
9. Tadelis, S. (2013). The Lean Product Playbook: How to Innovate with Minimum Viable Products and apid Customer Feedback. O'Reilly Media