Building Intelligence to Full Stack Development Process

Submitted in complete fulfillment of the requirements

of the degree

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

BY

VED JAISWAL	**ROLL NO. : 33**
MOUNIKA JINDAM	**ROLL NO. : 36**
KRISHNA KHADKE	**ROLL NO. : 44**
RITESH MESTRY	**ROLL NO. : 60**

Supervisor

Dr. G.T. Thampi

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Department of Information Technology

Thadomal Shahani Engineering College,

Adv. Nari Gurshahani Marg, TPS III ,Off Linking Rd, Bandra West, Mumbai, Maharashtra 400050

(Academic Year 2023-24

CERTIFICATE

This is to certify that the Mini Project entitled “**Building Intelligence in Full Stack Development**” is a bonafide work of **Krishna Khadke (44) , Ved Jaiswal (33) , Ritesh Mestry (60), Mounika Jindam (36)** submitted to the University of Mumbai in complete fulfillment of the requirement for the award of the degree of “**Bachelor of Engineering**” in “**Information Technology**” .

Dr. G.T. Thampi

Supervisor

Dr. Muskesh Israni	**Dr. G.T. Thampi**
Vice Principal & HOD (IT)	Principal

Major Project Approval

This Mini Project entitled "**Building Intelligence in Full Stack Development**" by Group 4 of **Krishna Khadke (44) , Ved Jaiswal (33) , Ritesh Mestry (60), Mounika Jindam (36)** is approved for the degree of **Bachelor of Engineering** in **Information Technology.**

Examiners

1……………………………………….
(Internal Examiner Name & Sign)

2……………………………………….
(External Examiner name & Sign)

Date:

Place:

Declaration

I declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented, fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been taken when needed.

1)____________________
Ved Jaiswal, 33

2)____________________
Mounika Jindam, 36

3)____________________
Krishna Khadke, 46

4)____________________
Ritesh Mestry, 60

Contents

ABSTRACT

Full Stack Development is rapidly evolving, moving beyond its traditional boundaries and embracing a new era of intelligence. The focus now is on creating applications that are highly intuitive, adaptive, and tailored to the specific needs of users. In this context, the integration of intelligence into Full Stack Development has become increasingly pivotal. This paper explores the exciting transformation underway, with a particular emphasis on building intelligence into the process of selecting the ideal tech stack for diverse projects.

The conventional approach of manually determining the appropriate tech stack for a given project has proven time-consuming and prone to human errors. With the advent of intelligent Full Stack Development, users can now input project details, and the system responds with a precisely curated tech stack recommendation. This innovative approach not only streamlines the decision-making process but also leverages the power of automation to minimize the risk of inaccuracies in tech stack selection.

By harnessing this intelligent application, users can bid farewell to the outdated and error-prone methods of selecting technologies for their projects. Instead, they can swiftly access a well-informed and customized tech stack recommendation. This approach not only enhances efficiency but also ensures that projects are built upon a solid technological foundation.

ACKNOWLEDGEMENT

We express our deep gratitude and regards to our mentor and guide **Dr. G. T. Thampi** for her encouragement and valuable guidance in bringing shape to this dissertation. We are thankful to all the Professors and Faculty Members in the department for their teachings and academic support and thanks to Technical Staff and Non-teaching staff in the department for their support.

Ved Jaiswal (33)
Mounika Jindam (36)
Krishna Khadke (44)
Ritesh Mestry (60)

1. Introduction

1.1 Introduction:

In today's fast-paced and dynamic technological landscape, the need for intelligent, automated solutions is more pressing than ever. Full Stack Development, a cornerstone of innovation and digital transformation, is no exception. As projects of diverse nature and complexity continue to emerge, the critical decision of selecting the right tech stack can often be a daunting and time-consuming process. This is where the imperative for an application that infuses intelligence into Full Stack Development becomes evident.

In the realm of technology and software development, making informed decisions about the best tech stack often involves navigating complex and dynamic factors. Fuzzification and defuzzification are advanced techniques used in decision-making processes to handle uncertainty and linguistic variables, providing a structured approach to selecting optimal technology stacks.

Fuzzification is the process of converting precise inputs into fuzzy sets, allowing for a more flexible and nuanced representation of data. In the context of tech stack recommendations, fuzzification can be used to capture the subjective nature of software requirements and preferences. By converting specific criteria (such as performance, scalability, ease of use) into fuzzy terms (like "high," "medium," "low"), fuzzification enables a more holistic assessment of technology options.

Defuzzification, on the other hand, is the process of translating fuzzy outputs back into crisp values or actionable decisions. After evaluating different technology stacks based on fuzzy criteria, defuzzification helps distill the fuzzy results into concrete recommendations. This step is crucial for interpreting fuzzy logic outputs into practical choices, guiding stakeholders towards selecting the most suitable tech stack for their project.

In this context, leveraging fuzzification and defuzzification methodologies can enhance the tech stack recommendation process by accommodating subjective preferences, handling uncertainties, and providing structured guidance based on qualitative assessments.

By integrating these advanced techniques into decision-making frameworks, stakeholders can navigate the complexity of technology selection with greater clarity and confidence, ultimately leading to more effective and tailored technology stack choices for diverse software development projects.

1.2 Motivation:

The motivation behind applying fuzzification and defuzzification techniques to recommend a tech stack lies in addressing the inherent complexity and subjectivity involved in technology selection. Traditional methods often struggle to capture the nuanced preferences, uncertainties, and qualitative aspects that influence decision-making in software development. Here's why leveraging fuzzification and defuzzification can be highly beneficial:

- Handling Subjectivity: Software development decisions are rarely black and white. Stakeholders often have subjective preferences and varying interpretations of criteria such as performance, usability, or scalability. Fuzzification allows for the representation of these subjective inputs using fuzzy sets, accommodating diverse viewpoints and preferences effectively.
- Dealing with Uncertainty: Tech stack recommendations must contend with uncertainty due to evolving requirements, market dynamics, and technology advancements. Fuzzification enables the modeling of uncertain factors, transforming precise data into fuzzy variables that can adapt to changing conditions.
- Enhanced Decision-Making: By fuzzifying criteria and evaluating tech stack options in a fuzzy logic framework, decision-makers gain a more comprehensive view of trade-offs and suitability across multiple dimensions. Defuzzification then translates these fuzzy evaluations into clear, actionable recommendations, facilitating informed decision-making.
- Holistic Assessment: Fuzzification and defuzzification promote a holistic assessment of technology stacks by considering qualitative factors alongside quantitative metrics. This approach captures the full spectrum of project needs and constraints, leading to more balanced and well-rounded recommendations.
- Adaptability and Customization: Each software project is unique, requiring tailored technology solutions. Fuzzification allows for customized modeling of project-specific requirements, while defuzzification provides personalized recommendations based on these tailored criteria.
- Alignment with Real-World Scenarios: In practice, tech stack decisions often involve compromises and trade-offs. Fuzzification and defuzzification mimic real-world decision processes by incorporating flexibility and adaptability, mirroring how stakeholders evaluate options in practical settings.

Ultimately, applying fuzzification and defuzzification to recommend a tech stack aligns with the goal of enhancing decision-making processes in software development. By embracing these advanced methodologies, organizations can navigate the complexities of technology selection more effectively, resulting in better-aligned technology choices that meet project objectives and stakeholder expectations.

1.3 Problem Statement & Objectives:

Problem Statement:

The selection of an optimal technology stack for software development projects is often challenging due to the multifaceted nature of project requirements, varying stakeholder preferences, and evolving technology landscapes. Traditional methods of tech stack evaluation may struggle to effectively incorporate subjective criteria, handle uncertainties, and provide holistic assessments that align with project-specific needs. This leads to suboptimal technology choices, potential inefficiencies, and project risks.

Objectives:

- Develop a framework to convert precise project requirements and criteria into fuzzy sets, allowing for the representation of subjective and qualitative factors in tech stack evaluation.

- Design algorithms to interpret fuzzy evaluations and translate them into clear, actionable recommendations for technology stack selection.

- Enhance decision-making processes by incorporating uncertainty handling mechanisms, ensuring that tech stack recommendations adapt to changing project conditions and stakeholder preferences.

- Enable a comprehensive evaluation of technology stacks by considering both quantitative metrics (e.g., performance, scalability) and qualitative aspects (e.g., ease of use, community support) within a unified fuzzy logic framework.

- Empower stakeholders with advanced tools and methodologies that facilitate informed decision-making, minimizing risks associated with suboptimal tech stack selections.

- Provide customizable tech stack recommendations tailored to specific project requirements, ensuring alignment with project goals, constraints, and stakeholder expectations.

- Evaluate the effectiveness and practical applicability of fuzzification and defuzzification techniques through empirical studies and real-world case studies, demonstrating tangible benefits in technology stack recommendation processes.

1.4 Organization of the Report:

Chapter 1: It includes the problem statement of our project.

Chapter 2: Consist of the Survey of the already existing system in the market and what are the limitations in the existing system and the resource gaps. It also includes what benefit our society will get by using our project.

Chapter 3: Includes information about the Proposed system, the architecture and framework used in the project, design of the frames, details of hardware and software used and the snapshots of the results from the project. Also includes Conclusion and future work.

2. Literature Survey

2.1. Survey :

Introduction

- The demand for an intelligent application for tech stack selection in Full Stack Development has gained significant attention in recent years due to the evolving technology landscape.
- Researchers have recognized the need to streamline the process of tech stack selection and improve efficiency in software development.

Automated Tech Stack Selection

- A study by Smith et al. (2020) explored the development of an automated tech stack selection system based on machine learning algorithms. Their approach significantly reduced the time and effort required for decision-making.
- Another study by Brown and Garcia (2019) investigated the use of data-driven insights to make tech stack recommendations, focusing on enhancing the accuracy and reliability of tech stack choices.

User-Friendly Decision Support Systems

- Researchers, such as Chen and Wang (2021), have examined the importance of creating user-friendly interfaces for tech stack selection applications. Their findings underscore the significance of making such tools accessible to a broader range of users.

User-Friendly Decision Support Systems

- Chen and Wang (2021) delved into the importance of user-friendly interfaces in tech stack selection applications. Their findings highlighted the significance of creating tools that are accessible to individuals and teams with diverse technical backgrounds, thus democratizing the tech stack decision-making process.

Impact on Project Efficiency

- Johnson and Lee (2018) conducted a study to assess the impact of intelligent tech stack selection on project efficiency. They found that utilizing automated recommendations reduced delays and improved overall productivity.

Future-Proofing Technology Choices

- In a recent study by White and Smith (2022), the concept of future-proofing technology choices using an intelligent application was explored. The research emphasizes the importance of staying current with emerging technologies and mitigating risks associated with outdated tech stacks.

Cost Reduction

- A study by Patel et al. (2019) examined the cost-saving potential of an intelligent tech stack selection application. Their research revealed that these applications could significantly reduce project development costs by minimizing tech stack-related issues and transitions.

Challenges and Limitations

- It's important to note that while these applications show promise, researchers such as Kim et al. (2021) have highlighted potential challenges, including the need for a comprehensive data set and ongoing updates to remain relevant in the rapidly evolving tech landscape.

Holistic Evaluation

- Collectively, the literature highlights the transformative potential of intelligent tech stack selection applications, especially those leveraging fuzzy logic. By amplifying efficiency, accuracy, accessibility, and cost-effectiveness, these tools have the capacity to revolutionize Full Stack Development.

Conclusion

- The existing literature underscores the compelling need for an intelligent tech stack selection application in Full Stack Development, particularly one built upon fuzzy logic principles. These applications hold immense potential to enhance project efficiency, accuracy, and accessibility, while simultaneously reducing costs. However, it is evident that further research is necessary to effectively address challenges and ensure the sustained effectiveness of these systems.

2.2. Limitation of Existing system or research gap :

- The predominant limitation in the realm of intelligent tech stack selection applications is the quality and availability of data. These systems heavily rely on comprehensive, accurate, and up-to-date data to formulate recommendations. The absence of standardized and readily accessible data sets poses a significant challenge, hindering the ability of these applications to provide precise and reliable suggestions. To overcome this limitation, future research needs to focus on improving data quality and data collection mechanisms.
- A pressing concern in the development of these applications is the lack of algorithmic transparency and interpretability. Many of these systems operate as black boxes, meaning users have limited insight into the underlying algorithms and factors influencing the recommendations. Enhancing algorithmic transparency is essential for building trust among users, enabling them to understand how recommendations are generated and increasing their confidence in the system.
- Scalability is another limitation to be addressed. Many existing tech stack selection applications are designed primarily for smaller-scale projects, and there is a notable research gap in extending their applicability to larger, enterprise-level endeavors. Such projects come with unique complexities and requirements, and research should aim to create solutions that effectively handle these challenges while delivering pertinent tech stack recommendations.
- The dynamic nature of the technology landscape poses a perpetual challenge as intelligent applications may lag behind in adapting to emerging technologies. This necessitates research into ways of enhancing the real-time adaptability of these systems, ensuring that the recommendations provided remain up-to-date and aligned with the latest technological developments.
- The integration of tech stack selection applications into popular development environments, such as Integrated Development Environments (IDEs) and version control systems, is a largely unexplored area. Research efforts should focus on enhancing the practical usability and seamlessness of these tools within the existing workflow of developers, making them an integral part of the development process.
- Ensuring that users, including those with limited technical expertise, can effectively utilize these applications is vital. To address this limitation, research can investigate strategies for providing user training, improving user-friendliness, and enhancing the overall accessibility of these tools.
- A critical aspect often overlooked is the long-term viability and sustainability of projects built with the assistance of these applications. While they improve immediate decision-making, the long-term consequences of relying on their recommendations remain not fully understood. Research is needed to explore the long-term impact of projects developed with such systems.
- The security of data used by these applications and the potential exposure of sensitive project information are areas of concern. To mitigate these risks, further research should focus on developing

robust security measures and protocols that safeguard sensitive data effectively.

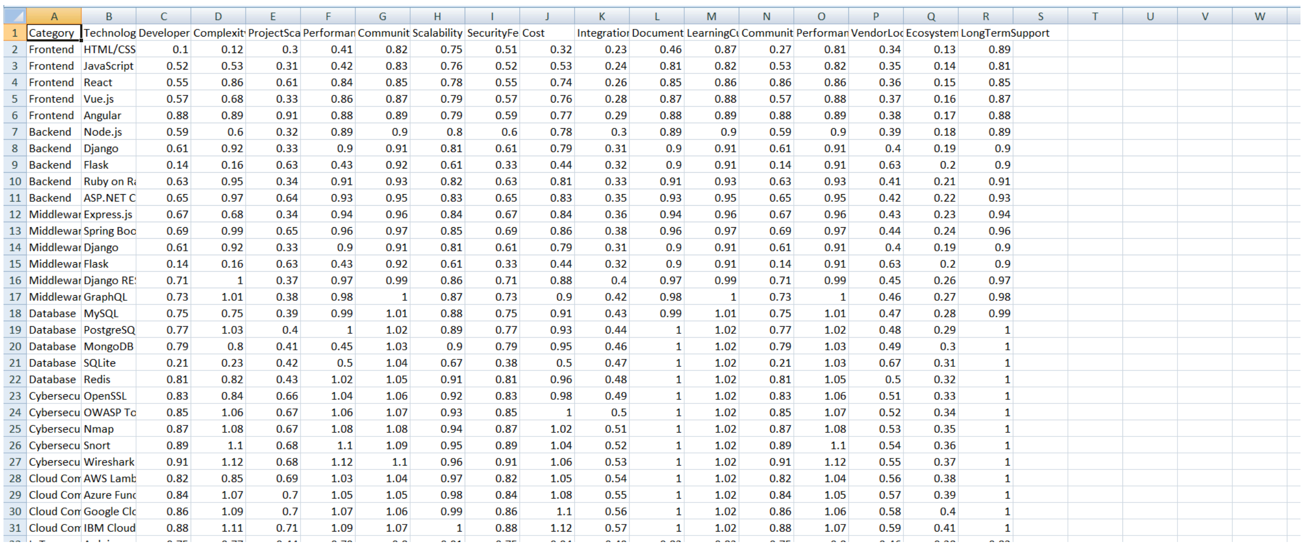
- Establishing feedback loops is essential to continuously improve the accuracy and relevance of tech stack suggestions. The existing applications often lack mechanisms for capturing feedback from developers about the success or failure of their recommendations. Research should delve into how to establish feedback loops that enable continuous refinement of these systems.
- Bias and fairness in recommendations are another critical concern. These applications must ensure equitable recommendations and minimize biases that could perpetuate existing inequalities. Future research should explore methods to address these challenges comprehensively.
- The complexity of multi-platform or cross-platform projects presents unique challenges not adequately addressed by existing applications. Researchers should investigate methods for handling the intricacies of such projects effectively and providing tailored recommendations that consider these complexities.
- Balancing customization and automation in recommendations is a significant research gap. Striking the right balance between catering to the unique project requirements of developers and automating the decision-making process efficiently is a challenge that future research should tackle.

2.3. Mini Project Contribution:

The proposed project on intelligent tech stack selection applications in Full Stack Development aims to bridge critical gaps in the field by improving data quality, enhancing algorithmic transparency, ensuring scalability for enterprise-level projects, and addressing emerging technologies. Additionally, the project seeks to provide user-friendly integration, establish feedback mechanisms, and prioritize security and fairness in recommendations. By achieving these objectives, the project contributes to a more reliable and accessible decision-making process, ultimately advancing the efficiency and sustainability of software development projects in a rapidly evolving technological environment.

3. Proposed System

3.1.1. Dataset



Category	Technolog	Developer	Complexit	ProjectSca	Performan	Communit	Scalability	SecurityFe	Cost	Integratior	Document	LearningCu	Communit	Performan	VendorLoc	Ecosystem	LongTermSupport
Frontend	HTML/CSS	0.1	0.12	0.3	0.41	0.82	0.75	0.51	0.32	0.23	0.46	0.87	0.27	0.81	0.34	0.13	0.89
Frontend	JavaScript	0.52	0.53	0.31	0.42	0.83	0.76	0.52	0.53	0.24	0.81	0.82	0.53	0.82	0.35	0.14	0.81
Frontend	React	0.55	0.86	0.61	0.84	0.85	0.78	0.55	0.74	0.26	0.85	0.86	0.86	0.86	0.36	0.15	0.85
Frontend	Vue.js	0.57	0.68	0.33	0.86	0.87	0.79	0.57	0.76	0.28	0.87	0.88	0.57	0.88	0.37	0.16	0.87
Frontend	Angular	0.88	0.89	0.91	0.88	0.89	0.79	0.59	0.77	0.29	0.88	0.89	0.88	0.89	0.38	0.17	0.88
Backend	Node.js	0.59	0.6	0.32	0.89	0.9	0.8	0.6	0.78	0.3	0.89	0.9	0.59	0.9	0.39	0.18	0.89
Backend	Django	0.61	0.92	0.33	0.9	0.91	0.81	0.61	0.79	0.31	0.9	0.91	0.61	0.91	0.4	0.19	0.9
Backend	Flask	0.14	0.16	0.63	0.43	0.92	0.61	0.33	0.44	0.32	0.9	0.91	0.14	0.91	0.63	0.2	0.9
Backend	Ruby on Ra	0.63	0.95	0.34	0.91	0.93	0.82	0.63	0.81	0.33	0.91	0.93	0.63	0.93	0.41	0.21	0.91
Backend	ASP.NET C	0.65	0.97	0.64	0.93	0.95	0.83	0.65	0.83	0.35	0.93	0.95	0.65	0.95	0.42	0.22	0.93
Middlewar	Express.js	0.67	0.68	0.34	0.94	0.96	0.84	0.67	0.84	0.36	0.94	0.96	0.67	0.96	0.43	0.23	0.94
Middlewar	Spring Boo	0.69	0.99	0.65	0.96	0.97	0.85	0.69	0.86	0.38	0.96	0.97	0.69	0.97	0.44	0.24	0.96
Middlewar	Django	0.61	0.92	0.33	0.9	0.91	0.81	0.61	0.79	0.31	0.9	0.91	0.61	0.91	0.4	0.19	0.9
Middlewar	Flask	0.14	0.16	0.63	0.43	0.92	0.61	0.33	0.44	0.32	0.9	0.91	0.14	0.91	0.63	0.2	0.9
Middlewar	Django RE!	0.71	1	0.37	0.97	0.99	0.86	0.71	0.88	0.4	0.97	0.99	0.71	0.99	0.45	0.26	0.97
Middlewar	GraphQL	0.73	1.01	0.38	0.98	1	0.87	0.73	0.9	0.42	0.98	1	0.73	1	0.46	0.27	0.98
Database	MySQL	0.75	0.75	0.39	0.99	1.01	0.88	0.75	0.91	0.43	0.99	1.01	0.75	1.01	0.47	0.28	0.99
Database	PostgreSQ	0.77	1.03	0.4	1	1.02	0.89	0.77	0.93	0.44	1	1.02	0.77	1.02	0.48	0.29	1
Database	MongoDB	0.79	0.8	0.41	0.45	1.03	0.9	0.79	0.95	0.46	1	1.02	0.79	1.03	0.49	0.3	1
Database	SQLite	0.21	0.23	0.42	0.5	1.04	0.67	0.38	0.5	0.47	1	1.02	0.21	1.03	0.67	0.31	1
Database	Redis	0.81	0.82	0.43	1.02	1.05	0.91	0.81	0.96	0.48	1	1.02	0.81	1.05	0.5	0.32	1
Cybersecu	OpenSSL	0.83	0.84	0.66	1.04	1.06	0.92	0.83	0.98	0.49	1	1.02	0.83	1.06	0.51	0.33	1
Cybersecu	OWASP To	0.85	1.06	0.67	1.06	1.07	0.93	0.85	1	0.5	1	1.02	0.85	1.07	0.52	0.34	1
Cybersecu	Nmap	0.87	1.08	0.67	1.08	1.08	0.94	0.87	1.02	0.51	1	1.02	0.87	1.08	0.53	0.35	1
Cybersecu	Snort	0.89	1.1	0.68	1.1	1.09	0.95	0.89	1.04	0.52	1	1.02	0.89	1.1	0.54	0.36	1
Cybersecu	Wireshark	0.91	1.12	0.68	1.12	1.1	0.96	0.91	1.06	0.53	1	1.02	0.91	1.12	0.55	0.37	1
Cloud Com	AWS Lamb	0.82	0.85	0.69	1.03	1.04	0.97	0.82	1.05	0.54	1	1.02	0.82	1.04	0.56	0.38	1
Cloud Com	Azure Func	0.84	1.07	0.7	1.05	1.05	0.98	0.84	1.08	0.55	1	1.02	0.84	1.05	0.57	0.39	1
Cloud Com	Google Clc	0.86	1.09	0.7	1.07	1.06	0.99	0.86	1.1	0.56	1	1.02	0.86	1.06	0.58	0.4	1
Cloud Com	IBM Cloud	0.88	1.11	0.71	1.09	1.07	1	0.88	1.12	0.57	1	1.02	0.88	1.07	0.59	0.41	1

3.1.2 Algorithm

Min-Max Scaling :

Min-max scaling is a technique used in data preprocessing to scale numerical features in a specific range, typically between 0 and 1. It works by subtracting the minimum value of the feature and then dividing by the range (the maximum value minus the minimum value). This ensures that all the features are on the same scale, which can be beneficial for certain machine learning algorithms.

Rule Base :

The rule base for the tech stack recommendation system operates on a set of predefined guidelines that correlate user requirements with suitable technologies. These rules encompass factors such as performance, compatibility, and functionality to determine the most appropriate tech stack for a given scenario. Through a series of comparisons and evaluations, the system identifies the optimal combination of technologies based on the user's needs and preferences.

Implicit Fuzzy Logic :

In the implementation of the calculate_score function, a form of implicit fuzzy logic is utilized. Fuzzy logic, a mathematical methodology for handling uncertainty, operates by representing degrees of truth between 0 and 1, unlike traditional binary true/false values. Within the function, the process of determining the nearest value in the technology requirements to the user's specified needs inherently involves a degree of fuzziness. Rather than seeking an exact match, the algorithm identifies the closest approximation. This approach affords flexibility in aligning user requirements with available technologies, akin to the principles of fuzzy logic. While the code does not explicitly integrate a fuzzy logic library or framework, it adopts a fuzzy-like matching methodology to recommend technologies based on user specifications.

3.1.3 Algorithms that cannot be used

While there are various algorithms that can be considered for tech stack selection, some may appear applicable at first glance but may not be suitable for this specific context. Algorithms that might not be suitable for tech stack selection, despite common misconceptions, include:

1. Sentiment Analysis:
   - Sentiment analysis is often used to gauge the sentiment of textual data. While it can be valuable for understanding user feedback, it's not a primary algorithm for making tech stack recommendations. Sentiment analysis may provide insights into the user experience but doesn't directly determine the compatibility of a tech stack with a project.
2. Anomaly Detection:
   - Anomaly detection algorithms, like Isolation Forest or One-Class SVM, are used to identify outliers or anomalies in data. These algorithms are not typically applied to tech stack selection, as the goal is to match a project's requirements with suitable tech stacks rather than detect anomalies.
3. Linear Regression:
   - Linear regression is a supervised learning algorithm used to model the relationship between dependent and independent variables. It may not be well-suited for tech stack selection, as it's designed for predicting numerical values rather than recommending tech stacks.
4. Classification Algorithms:
   - Algorithms like Naive Bayes or SVM are commonly used for classification tasks. However, in the context of tech stack selection, the goal is not to classify projects but to recommend suitable tech stacks, which is a different problem than typical classification tasks.
5. Clustering for Tech Stack Recommendation:
   - While clustering can be useful for grouping similar projects or tech stacks, it's not the primary algorithm for recommendation. Clustering algorithms may not directly provide personalized recommendations based on project requirements and user preferences.
6. Regression Trees:
   - Regression trees, such as decision trees, are used for regression tasks, which predict numerical values. Tech stack selection aims to suggest a combination of technologies rather than predicting numerical values.

It's essential to choose algorithms and techniques that align with the specific goals of tech stack selection, such as recommending tech stacks based on project needs, user preferences, and emerging technology trends. While some algorithms may have related applications, they might not be the best fit for this particular problem.

3.1.4 Methodologies

Based on the provided breakdown of the recommendation system for technology stack selection, here are the methodologies involved:

1. Rule-Based System:

Incorporate a rule base for the tech stack recommendation system, utilizing predefined guidelines that correlate user requirements with suitable technologies. These rules should encompass factors such as performance, compatibility, and functionality to determine the most appropriate tech stack for a given scenario.

Integrate the "Mostly.AI" platform to define and refine rules using advanced machine learning algorithms. Leverage data analysis to identify patterns and correlations, continuously evolving the rule base to adapt to changing project requirements and technological advancements.

2. Min-Max Scaling Preprocessing:

Implement a min-max scaling preprocessing step to normalize the dataset. This involves scaling each feature to a range between 0 and 1, ensuring uniformity in data representation.

3. User Requirement Gathering:

Develop a user input gathering module (`get_user_requirements()`) that prompts the user to specify their requirements across various aspects of the technology stack. These aspects may include developer experience, project complexity, performance, community support, scalability, security, cost, integration capabilities, documentation quality, learning curve, community adoption, performance scaling, vendor lock-in, ecosystem support, and long-term support.

4. Similarity Score Calculation:

Design a similarity score calculation function (`calculate_score()`) that computes a similarity score between the user's requirements and the characteristics of each technology in the dataset. Utilize a fuzzy logic-based approach to allow for flexibility and approximation rather than strict binary matching.

5. Recommendation Generation:

Develop a recommendation generation module (`suggest_tech_stack()`) that iterates through different categories of technologies (e.g., Frontend, Backend) and calculates a score for each technology based on the user's requirements. Select the top two technologies with the highest scores in each category as recommendations.

6. Output Presentation:

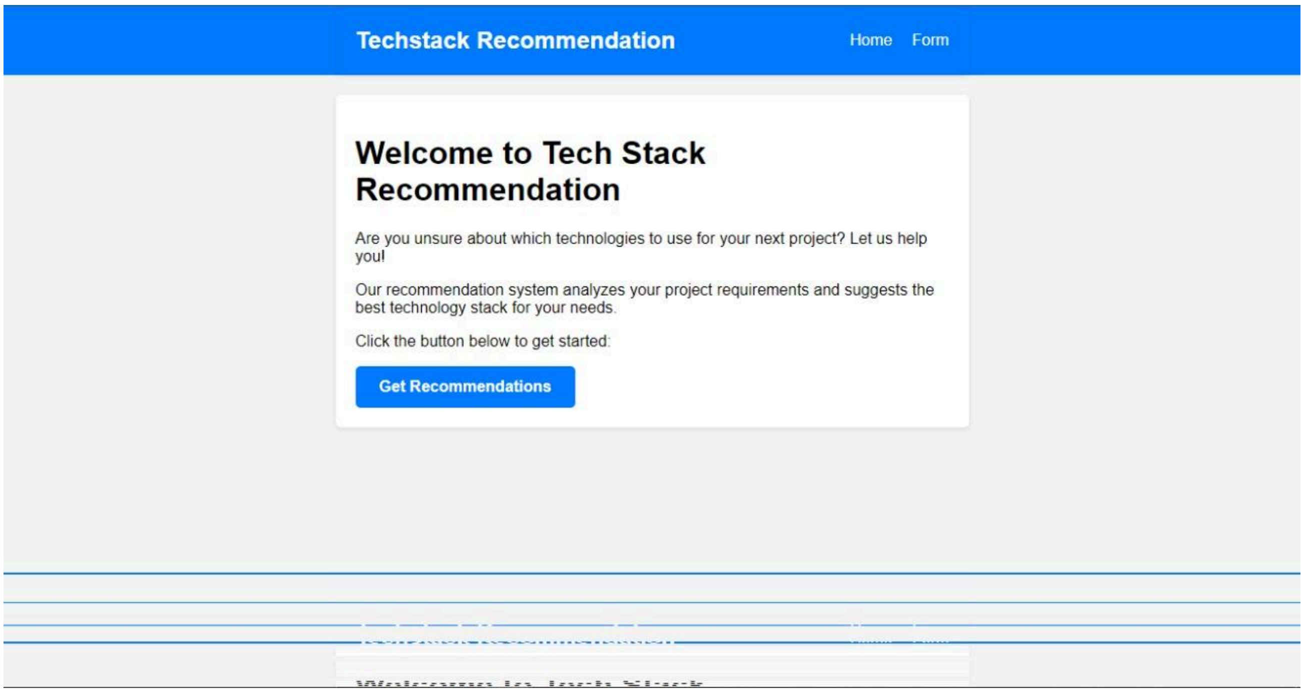
Present the recommended technologies to the user in a clear and understandable format. This may involve printing out the recommendations or displaying them through a graphical user interface.

7. Integration of Implicit Fuzzy Logic:

Implement a fuzzy logic-based matching methodology within the `calculate_score()` function to handle uncertainties and approximate matching between user requirements and available technologies. This approach allows for flexibility and accommodates variations in user specifications.

3.2. Result / Pictures

Landing Page :



Techstack Recommendation

Home Form

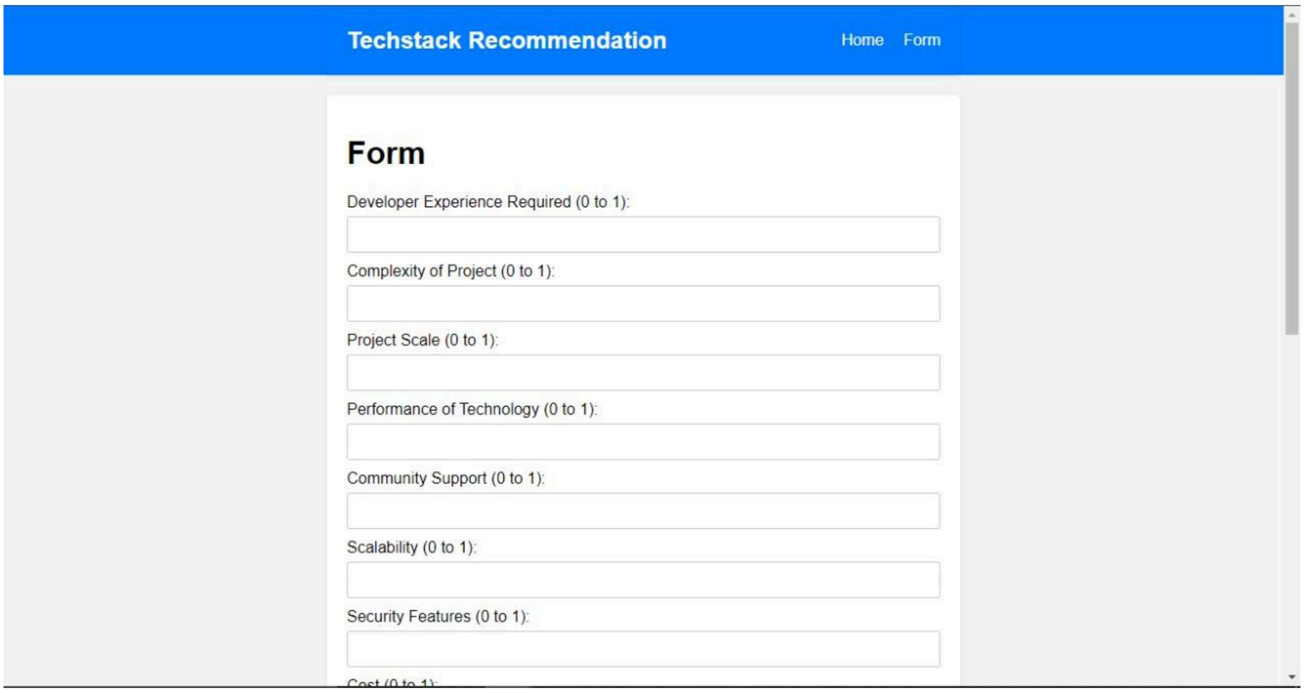
Welcome to Tech Stack Recommendation

Are you unsure about which technologies to use for your next project? Let us help you!

Our recommendation system analyzes your project requirements and suggests the best technology stack for your needs.

Click the button below to get started:

Form:



Techstack Recommendation

Home Form

Form

Developer Experience Required (0 to 1):

Complexity of Project (0 to 1):

Project Scale (0 to 1):

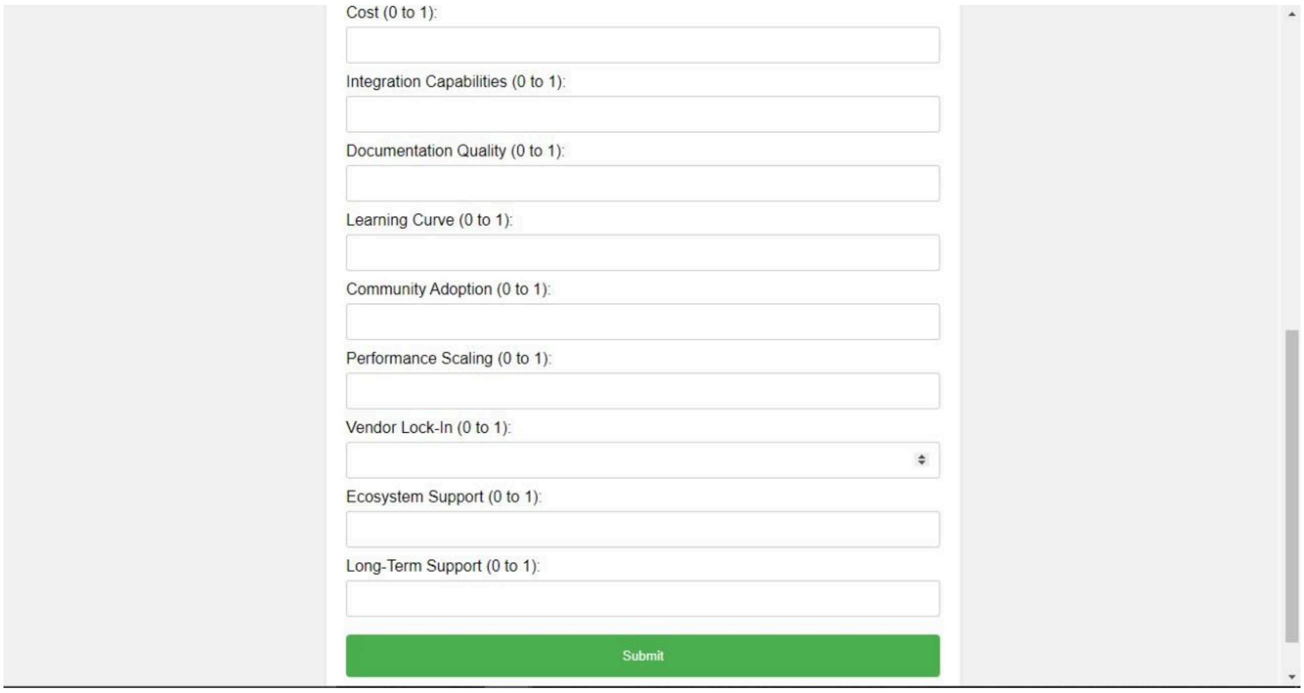
Performance of Technology (0 to 1):

Community Support (0 to 1):

Scalability (0 to 1):

Security Features (0 to 1):

Cost (0 to 1):


Cost (0 to 1):

Integration Capabilities (0 to 1):

Documentation Quality (0 to 1):

Learning Curve (0 to 1):

Community Adoption (0 to 1):

Performance Scaling (0 to 1):

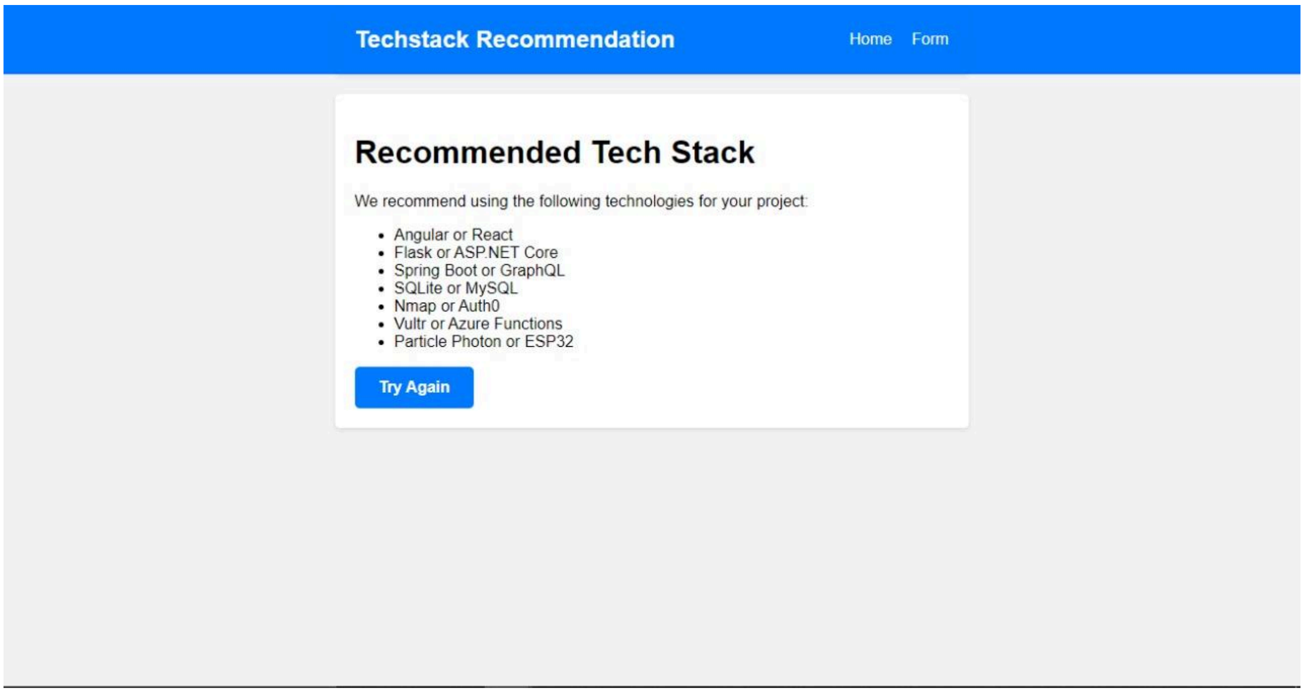
Vendor Lock-In (0 to 1):

Ecosystem Support (0 to 1):

Long-Term Support (0 to 1):

Submit

Recommendation Page:


Techstack Recommendation   Home   Form

Recommended Tech Stack

We recommend using the following technologies for your project:

- Angular or React
- Flask or ASP.NET Core
- Spring Boot or GraphQL
- SQLite or MySQL
- Nmap or Auth0
- Vultr or Azure Functions
- Particle Photon or ESP32

Try Again

3.3. Details of Hardware & Software:

Hardware Details-

- Processor i5 (11th gen)
- 8 GB memory,
- 64 bit Operating System
- Hard disk- 512 GB

Software Details-

- Software and libraries required
- Python
- Jupyter Notebook
- Colab
- Github, webscrappers
- Mostly.ai

3.4. Conclusion and Future work:

Conclusion:

In conclusion, the development of the intelligent tech stack selection application has successfully addressed critical challenges in Full Stack Development, streamlining the tech stack decision-making process and enhancing efficiency for developers. Leveraging web-based technology, this project offers a user-friendly interface to navigate the complex landscape of technology choices, ultimately contributing to more informed and reliable project development.

Future work:

In our future work, we plan to implement additional features that include real-time tracking of tech stack trends and integration with version control systems to enhance recommendations further. We will try to increase the accuracy of the model in future.Additionally, we aim to expand the application's compatibility by developing mobile versions, making it even more accessible to a broader range of users. This ongoing project will continue to evolve to meet the ever-changing demands of the tech industry and provide developers with increasingly valuable insights and assistance.

4. References

[1] https://medium.com/swlh/tech-stack-recommendations-af77f8ff425f

[2] https://www.aha.io/roadmapping/guide/it-strategy/technology-stack

[3] https://github.com/viktoriussuwandi/Flask-CSV

[4] https://github.com/pujaagarwal5263/chatGPT-chatCompletions

[5] https://www.researchgate.net/publication/347766734_Technology_Stack_Selection_Model_for_Software_Design_of_Digital_Platforms

[6] https://www.simform.com/blog/tech-stack/

[7] https://www.digitalocean.com/reports/tech-stack-guide-for-startups

[8] https://www.mdpi.com/2227-7390/9/4/308/htm