

JOB FINDER: AI JOB SEARCH ASSISTANT FOR THE UNEMPLOYED

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

DEVARAPALLI BHASKAR SATYANARAYANA (Reg.No - 40110310)

T DHANUSH (Reg.No - 40110314)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Category-1 University by UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

APRIL - 2024





SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **DEVARAPALLI BHASKAR SATYANARAYANA**(Reg No-40110310), **T DHANUSH**(Reg No-40110314) who carried out the Project entitled "**JOBFINDER: AI JOB SEARCH ASSISTANT FOR THE UNEMPLOYED**" under my supervision from November 2023 to April 2024.


15/4/24

Internal Guide

Dr. K. VEENA, M.E., Ph.D.



Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Reg. Gandhi Salai,
Chennai - 600 119.

Submitted for Viva voce Examination held on 15-04-2024


15/4/24

Internal Examiner


15/4/24

External Examiner

DECLARATION

I, **D Bhaskar Satyanarayana** (Reg.No-40110310), hereby declare that the Project Report entitled **JOB FINDER: AI JOB SEARCH ASSISTANT FOR THE UNEMPLOYED** done by me under the guidance of **Dr.K.Veena, M.E.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 15-04-2024

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board Of Management** of **Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.K.Veena, M.E.,Ph.D.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

An industry or a business or a firm requires quality personnel for accomplishment of objectives framed by them in order to survive in this competitive era. They all are in the beginning of fourth industrial revolution. To remain competitive in this digital world all search for bright, potential and dynamic employees. Organizations with an effective recruitment strategy will be able to employ suitable individual in order to manage the digital world and developing business environment. So the recruitment strategy is the prime factor for every organization in hiring skilled employees who could be more efficient and effective in accomplishing the job objectives. The recruitment strategy as it is a major function of organization apparently takes help of data analysis for decision making process. The data analysis is known as “Artificial Intelligence” which plays a crucial role in recruitment decision. Artificial intelligence in a most basic terminology and is a human develop intelligent machines. AI will work and react like human and its ultimate goal is to facilitate computers to carry out the work as normally done by people. AI leads with an incredible speed and accuracy. The major objective of this project is to study how Artificial Intelligence influences the recruitment strategy.

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		viii
1	INTRODUCTION		1
2	LITERATURE SURVEY		3
	2.1	Proposed System	16
	2.2	Inferences from Literature Survey	16
	2.3	Open Problems in Existing System	16
3	REQUIREMENT ANALYSIS		17
	3.1	Scope of the investigation	17
	3.2	Feasibility studies / risk analysis of the project	18
	3.3	Software requirements specification document	19
4	DESCRIPTION OF PROPOSED SYSTEM		20
	4.1	Materials and methods	20
	4.2	Selected methodology or process model	22
5	IMPLEMENTATION DETAILS		28
	5.1	System study	28
	5.2	System testing	29
	5.3	Unit Testing	31
	5.4	Integration testing	31
	5.5	Acceptance testing	32
	5.6	Architecture / overall design of proposed system	32
	5.7	Description of software for implementation and testing plan of the proposed model / system	47
	5.8	Project plan	49
6	RESULTS AND DISCUSSIONS		56
7	CONCLUSIONS		59
	REFERENCES		61
	APPENDIX		64
	A. SOURCE CODE		64

	B. SNAPSHOTS	78
	C. RESEARCH PAPER	80
	D. CERTIFICATE	86

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
5.6.1	System Architecture for Job finder	32
5.7.1	Decision Tree	48
5.8.1	Use Case Diagram for Job finder	53
5.8.2	Class diagram for Job finder	53
5.8.3	Sequence diagram for Job finder	54
5.8.4	Activity diagram for job finder	55
6.1	Enter the details	57
6.2	Fill user details	57
6.3	Recruitment status mail	58
6.4	Accuracy report	58
B.1	Enter the details	78
B.2	Fill user details	78
B.3	Recruitment status mail	79
B.4	Accuracy report	79

CHAPTER 1

INTRODUCTION

Organizations develop and sustain by innovating new ideas to compete in the digital era. Innovation leads to decrease in manpower and increase in machine handling. Organizations will also need to train its employees to handle machines, software or any equipment for that matter. Now organizations are trying to implement recruitment techniques which lead to acquire talented employees. Artificial Intelligence term was coined by John McCarthy (1950) soon after when he published a paper entitled "Computing Machinery and intelligence" and this paper lead to open the doors to the field termed as AI. Now, HR managers implement Artificial Intelligence technology to recruit, retain and inspire the proficient manpower which leads to success and growth for both the employer and employee. AI in recruiting has a major role in talent acquisition (Madeline Laura no Co-Founder & Chief Research Office). Nearly 30% of companies implementing AI for recruiting activity which can lesser down time, cost reduction and can place right talent in the right position. Artificial Intelligence is defined as "An area of study in the field of computer science. Artificial intelligence is concerned with the development of computers able to engage in human-like thought processes such as learning, reasoning, and self-correction". Artificial intelligence is all about fast thinking with lot of knowledge, think as human, logical reasoning, etc., as it is a part of computer science leading to efficiency in problem solving by giving solutions.

Artificial intelligence is technologically superior and superior to create robots. A PC or product is designed for human imitation and ways to find conclusions. Artificial intelligence is one of the most famous enterprise regions in finance, era and technology these days. With recent technological advances, the age of gift giving may be a completely glad one. Artificial Intelligence and Machine Learning. Central offerings that had been once executed bodily are actually finished through software and automated structures [1]. John McCarthy. some other well-known American laptop scientist, artificially describes artificial intelligence (AI) and intelligent vehicles.

Machines paintings tough and do their task beneath tough conditions. Excellent use of perceptual intelligence, that's manipulated, solves issues, masters common sense and recognizes complicated ideas, learns fast from revel in, and develops.

In today's dynamic and competitive job market, unemployment is a pressing concern, with individuals often encountering obstacles and uncertainties as they navigate the process of finding employment. The AI Job Search Assistant project emerges as a response to these challenges, aiming to revolutionize the job search experience through the integration of artificial intelligence technologies. By leveraging sophisticated algorithms and natural language processing capabilities, this project endeavors to offer tailored support and guidance to those who are currently unemployed, thereby enhancing their prospects of securing meaningful employment opportunities.

Unemployment not only impacts individuals' financial stability but also takes a toll on their morale and sense of purpose. Many unemployed individuals face difficulty in accessing the resources and guidance necessary to effectively navigate the job market landscape. Traditional job search methods can be time-consuming and often yield limited results, leaving individuals feeling discouraged and disheartened. Recognizing these challenges, the AI Job Search Assistant project seeks to empower unemployed individuals by providing them with a personalized and efficient tool to streamline their job search efforts.

By harnessing the power of artificial intelligence, the project aims to offer a comprehensive solution to the complexities of job hunting. Through advanced algorithms and machine learning techniques, the AI assistant will analyze the unique skills, qualifications, and preferences of each user, enabling it to generate tailored job recommendations from a myriad of sources, including online job boards, company websites, and professional networks. This personalized approach not only saves users time and effort but also increases the likelihood of them discovering relevant employment opportunities that align with their interests and expertise.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations

2.1 PROPOSED SYSTEM

The Work of Dr. Shejin Cherian et al [1] Through qualitative research, this paper gives a descriptive overview of Artificial Intelligence, its effect on our day-to-day life on how it would transform the way we think, perceive and most importantly, live. Drawing on from sources including prominent article writers, professors and websites on the same, this paper sheds light on why Artificial Intelligence (AI) is essential in the field of education humankind to progress, especially for the Indian Education System, also expressing the limitless possibilities of AI and its relevance in the in the years to come. This article then explores the sheer magnitude of changes AI has brought forward, in the Indian education sector. Since the year 2000, there has been an exponential increase in IT companies investing in the education sector so as to give the students a customized and

personalized learning experience, which has fared well for the students looking to plug the skill gap between a fresher and an experienced working individual.

A review on software requirements validation and consistency management

Requirements need to be validated at the early phase of the software development to avoid errors such as inconsistency, incompleteness and incorrectness. Drawn from this argument, a requirements validation process needs to consider Consistency, Completeness and Correctness ("3 Cs") for the production of a quality software specifications. This paper provides a review of requirements validation and consistency management based on the existing literature in order to identify the gaps in the existing knowledge on the process of software requirements specifications. This paper begins with a review of the definitions of the 3Cs, upon which the understanding of the 3Cs is derived. Next comprehensive review of related works on the identified consistency management techniques: traceability and analysis approaches are then presented. This is supported with a heat map representations of the related research on the types of contributions, techniques, specifications and semantics used in consistency management. Since semi-formal specifications were found as the most common representation of the requirements, the types of models used as semi-formal specifications to represent the requirements were also discussed. Overall, this paper identifies the various gaps existing within the process of validating and managing the consistency of requirements to avoid re-inventing the wheels in the diverse and comprehensive knowledge of requirements engineering.

Niketan Bothe¹, Priya Sudewad², Vaishnavi kasar et al [2] Increasing demand for university education is putting pressure on campuses to make better use of their real estate resources. The current higher education landscape was facing multifaceted disruption due to a new digital student body. Then COVID-19 happened, making this transition even faster. These new students are expecting a more personalized and intuitive smart campus experience, just like they find on the web. With the growing concern about meeting enrolment goals, universities need to reshape how they move forward. They need to adopt technologies other industries have long grown accustomed to. This paper outlines our efforts to address classroom under- utilization in real university. So by instrumenting classrooms with IOT sensors to measure realtime usage, using AI to predict

attendance in classrooms, Social distance monitoring, Crowd scanning and performing optimal allocation of rooms to courses so as to minimize space wastage.

Dr. Renuka Ekanath Walunj et al [3] In today's pandemic situation Technology is playing a very significant role in the economic and commercial life of different countries globally. Changing technology, evaluation, assessment and critical thinking as a developing talent can take the county forward. It is urged by society to build the best leadership and talent amongst the students as they are pillars for the successive future of the county. In the present corporate world a lot of attention is given to the practical knowledge and experience which is lacking to some extent in the education system. In this new changing scenario it is needed to fill the gap between theoretical knowledge and practical knowledge which is required in the corporate world as well in the entrepreneurship world. Now it is a challenge in front of students and teachers to accept and bend towards new technology in teaching and learning pedagogy.

S.Surenderkumar¹, K.M.Priya et al [4] This paper proposes strategies to improve the quality of students pursuing engineering in India. The effective action required for improving technical education to make it effective not only to creamy layer students but also to the slow learning students pursuing education in engineering institutions are recommended in the paper. The current scenario of engineering colleges and learning trend of students is briefed. The idea of the paper to improvise technical education in student perspective is discussed which involves, activity based teaching-learning pedagogy. The idea of increasing to five years degree, which improves learning opportunities for slow learning students, is also proposed.

T. Kiran, S. Farhan, H. Tauseef[5] and M. A. Fahiem, "Requirement validation model for virtual distributed system", *Int. J. Multidiscip. Sci. Eng.*, vol. 4, no. 8, pp. 34-42, 2013. In the beginning of the fourth industrial revolution, organizations are increasingly relying on Artificial Intelligence (AI) to enhance their recruitment strategies. AI brings efficiency by automating repetitive tasks like resume screening and candidate sourcing, improving decision-making through

data analysis, and enhancing the candidate experience with chatbots for real-time interaction. Despite these advantages, challenges such as the lack of human touch, technical complexities, and potential for algorithmic bias must be addressed. The methodology involves data collection from various sources, preprocessing for quality assurance, model selection and training, evaluation and validation, and deployment and integration into existing recruitment systems, with ongoing monitoring for effectiveness and fairness. AI's role in recruitment signifies a shift towards leveraging technology to meet the demands of the digital world and the evolving business landscape.

Dhirendra Pandey[6], Waris Khan and Vandana Pandey, *ROLE OF REQUIREMENT VALIDATION IN REQUIREMENT DEVELOPMENT*, 2012. Effective requirement elicitation is indeed crucial for developing quality software products that meet user needs. However, it is equally important to validate these requirements to ensure their correctness and completeness. Requirement validation involves checking the gathered information against predefined criteria to confirm its accuracy and adequacy for the intended purpose. This process plays a vital role in the specification development process, as it helps identify and rectify any inconsistencies, ambiguities, or missing details in the requirements documentation. By emphasizing the importance of requirement validation over elicitation, this paper highlights the significance of ensuring that the final set of requirements accurately represents the stakeholders' needs and can serve as a reliable foundation for the software development process. Additionally, the paper delves into the scope of requirement validation within the broader context of requirement development, emphasizing its role in mitigating risks and ensuring the successful implementation of proposed requirements, ultimately leading to the production of high-quality software products.

U.A. Raja[7], "Empirical studies of requirements validation techniques", *2009 2nd International Conference on Computer Control and Communication*, pp. 1-9, 2009, February. Requirements validation techniques (RVTs) are essential for detecting potential defects.

.X. Zhu,[8], and A. B. Goldberg, "Introduction to semi-supervised learning", Synthesis lectures on artificial intelligence and machine learning, vol. 3, no. 1, pp. 1-130, 2009. Prototyping serves as an effective RVT by providing stakeholders with a tangible representation of the system, allowing them to interact with it and identify any issues early in the development process. The advantage of a prototyping-based methodology lies in its ability to quickly deliver a system for user interaction, even if it's not fully functional for organizational use initially. In the case of the proposed web application, usability testing was conducted to assess system quality, which directly influences users' intentions to use the application in the future. The analysis highlights that factors such as the attractiveness of the user interface, performance, and functionality significantly impact the acceptance of the application. The mean scores for intention to use and ease of use indicate a positive reception of the web application, with users generally finding it acceptable. These findings underscore the importance of considering user experience and system quality in ensuring the success and acceptance of software applications like TIAS (the proposed web application).

SHEJIN CHERIAN,[9], ROLE OF ARTIFICIAL INTELLIGENCE IN THE PRESENT DAY EDUCATION. Student, Dept. of Mechanical Engineering, Dronacharya College of Engineering, Haryana, India, 2020. The abstract outlines a qualitative research paper that delves into the transformative impact of Artificial Intelligence (AI) on day-to-day life, particularly focusing on its implications for the Indian Education System. Drawing upon various sources including articles, academic insights, and websites, the paper highlights the significance of AI in driving progress and innovation in education.

Niketan Bothe, Priya Sudewad, Vaishnavi kasarlt[10] discusses how AI technologies have revolutionized the learning experience by offering personalized and customized educational opportunities, thereby bridging the skill gap between fresh graduates and experienced professionals. The paper also explores the use of machine learning and image recognition in enhancing efficiency and providing diverse opportunities for students to excel in their chosen industries. Overall, the paper aims to provide a descriptive overview of AI's role in education, with a

specific focus on its applications within the Indian context, showcasing its potential to reshape learning paradigms and foster growth in the years to come.

Surenderkumar¹, K.M.Priya²[11], ¹Assistant Professor, Department of EEE, Vidya Vikas Institute of Engineering and Technology, Mysuru, Karnataka, India, ²BE Graduate, Department of Biomedical Engineering, Dhanalakshmi Srinivasan Engineering College, Trichy, TamilNadu, India .The abstract outlines a paper proposing strategies to enhance the quality of engineering education for students in India, focusing on both high-achieving and slower-learning students.

Dr. Renuka Ekanath Walunj [12]The paper discusses the need for effective actions to improve technical education, considering the current state of engineering colleges and learning trends among students. It advocates for a shift towards activity-based teaching-learning pedagogy to better engage students and enhance their understanding of technical concepts. Additionally, the paper suggests extending the duration of engineering degrees to five years to provide more learning opportunities for students who may require additional time to grasp complex topics. Overall, the paper aims to address the challenges in engineering education and offer practical solutions to improve the learning experience for all students.

Katarina Sperling, Carl-Johan Stenberg, Cormac McGrath, Anna Åkerfeldt, Fredrik Heintz, Linnéa Stenliden,[13] In search of artificial intelligence (AI) literacy in teacher education: A scoping review, Computers and Education Open, Volume 6, 2024, 100169, ISSN 2666-5573.The scoping review examines the evolving concept of AI literacy within the context of Teacher Education (TE) and explores its implications for teachers' professional knowledge. Conducting a comprehensive search from 2000 to 2023, the review analyzes 34 papers focusing on the intersection of AI literacy and TE. Employing Aristotelian concepts such as *episteme* (theoretical-scientific knowledge), *techne* (practical-productive knowledge), and *phronesis* (professional judgment), the review elucidates implicit and explicit dimensions of teachers' professional knowledge in relation to AI literacy. The findings highlight the emergence of AI literacy as a globally

recognized research area in education, albeit with limited attention within TE contexts. While the literature encompasses diverse topics and methodological approaches, the influence of computer science and exploratory teaching methods is notable in shaping epistemic, practical, and ethical knowledge related to AI. However, the review reveals a gap in addressing teachers' professional knowledge comprehensively, with ethics primarily explored in terms of understanding technical aspects of data-driven AI technologies. Teachers' practical knowledge often manifests in the adoption of digital resources for teaching about AI or integrating AI educational technology into their teaching practices. By identifying research gaps, particularly concerning practical and ethical dimensions of teachers' knowledge, the review contributes to a deeper understanding of AI literacy in teaching and underscores the need for more informed AI literacy education in TE. Additionally, it lays the groundwork for future research aimed at enhancing teachers' professional knowledge in the rapidly evolving landscape of AI education.

Sridevi G. M. , S. Kamala Suganthi, [14] AI based suitability measurement and prediction between job description and job seeker profiles, International Journal of Information Management Data Insights, Volume 2, Issue 2, 2022, 100109. Advantages of using an AI-based system for candidate suitability assessment include increased efficiency and accuracy in the hiring process. By automatically analyzing candidate resumes against job descriptions, the system saves time and resources while ensuring a thorough evaluation. Additionally, machine learning algorithms like XGBoost can accurately predict candidate suitability based on predefined criteria, reducing the potential for human error. The scalability of such systems allows them to handle large volumes of resumes and job descriptions, making them suitable for organizations with high hiring demands. However, there are also disadvantages to consider. One potential drawback is the presence of biases in the data used to train the AI model. If the training data contains biases, such as gender or racial biases, the system may inadvertently perpetuate these biases in candidate selection. Additionally, AI systems may struggle to fully understand the nuances of language and context, leading to inaccurate assessments of candidate suitability. Overreliance on technology may also overlook important intangible qualities that are crucial for job performance but

not easily quantifiable, potentially resulting in suboptimal hiring decisions. In terms of methodologies, the use of cluster analysis helps to group job descriptions and candidate resumes into clusters based on common themes and patterns, facilitating the comparison of their similarities using Jaccard similarity. This quantitative approach forms the basis for measuring candidate suitability. Employing machine learning classifiers such as linear regression, decision tree, Adaboost, and XGBoost enables the prediction of candidate suitability based on features extracted using the bag of words technique. Feature engineering plays a crucial role in representing text data in a format suitable for machine learning algorithms to process. Evaluation metrics, such as accuracy, help assess the performance of different classifiers, with XGBoost achieving the highest average accuracy of 95.14% in this scenario.

Jenny S. Wesche,[15], Andreas Sonderegger, Repelled at first sight? Expectations and intentions of job-seekers reading about AI selection in job advertisements, Computers in Human Behavior, Volume 125, 2021, 106931, ISSN 0747-5632, Artificial intelligence (AI) is increasingly employed in personnel selection, promising advantages such as efficiency, standardization, and cost-effectiveness. However, research indicates potential disadvantages, particularly negative applicant experiences stemming from perceptions of impersonality and unfairness. To investigate this further, three experiments were conducted, systematically manipulating job advertisement information regarding the automation of different selection stages. Results revealed small negative effects when screening was conducted by AI versus humans, with stronger negative impacts observed when both screening and interviews were AI-driven. A qualitative approach was used to explore potential reasons for these effects, highlighting the significance of different stages in the selection process. This research underscores the importance of considering the implications of AI adoption on job-seekers' perceptions and intentions, informing future recruitment strategies.

Giacomo Damioli,[16], Vincent Van Roy, Dániel Vértesy, Marco Vivarelli, Drivers of employment dynamics of AI innovators, Technological Forecasting and Social Change, Volume 201, 2024, 123249, ISSN 0040-162, This study provides

new insights into the relationship between innovation, particularly in the realm of artificial intelligence (AI) technologies, and employment growth within companies. By analyzing data from 4,184 firms that patented AI inventions between 2000 and 2016, employing the GMM-SYS estimation method, the study finds a positive and statistically significant impact of AI patent families on employment levels. However, this effect is relatively small and predominantly observed in service sectors and among younger firms, which are often at the forefront of the AI revolution. Additionally, the study identifies evidence of increasing returns, indicating that companies more heavily focused on AI technologies tend to experience greater job creation impacts from their innovative efforts. These findings underscore the role of innovation, particularly in AI, as a driver of employment growth, particularly for companies positioned to capitalize on advancements in this field.

Araz Zirar,[17], Syed Imran Ali, Nazrul Islam, Worker and workplace Artificial Intelligence (AI) coexistence: Emerging themes and research agenda, Technovation, Volume 124, 2023, 102747, ISSN 0166-4972, This study delves into the evolving relationship between workers and artificial intelligence (AI) in the workplace, aiming to understand how they can coexist effectively. Through a systematic search protocol across prominent academic databases, four key themes emerged: workers' distrust in AI due to perceived job threats, the potential for AI to augment worker abilities and facilitate interactions, the necessity for workers to possess technical, human, and conceptual skills for successful coexistence, and the importance of ongoing reskilling and upskilling to foster a symbiotic relationship with AI. The study proposes four propositions and outlines relevant research questions to guide future investigations. Notably, it argues that workers' distrust in AI is better explained through an existential lens, identifies essential skills for worker-AI coexistence, emphasizes the significance of human and conceptual skills alongside technical expertise, and offers 20 evidence-informed research questions for scholarly inquiry. This review contributes to the understanding of worker-AI dynamics and provides a roadmap for future research in this burgeoning field.

Herman Aguinis,[18], Jose R. Beltran, Amando Cope, How to use

generative AI as a human resource management assistant, *Organizational Dynamics*, Volume 53, Issue 1, 2024, 101029, ISSN 0090-2616, This study highlights the potential of generative artificial intelligence (AI), specifically ChatGPT, as an assistant for Human Resource Management (HRM) professionals, addressing their overwhelming workload and job complexity, often leading to burnout. By providing specific guidelines for creating high-quality prompts, the study demonstrates how AI can offer valuable recommendations for both strategic and operational HRM tasks. The effectiveness of these prompts is illustrated across various HRM domains, with a focus on areas such as workforce diversity and strategic HRM. Additionally, recommendations for implementing a critical verification process to validate ChatGPT's suggestions are provided. The study concludes by emphasizing the importance of adequately trained HRM professionals utilizing AI tools to enhance efficiency, reduce burnout, and allocate more time to strategic initiatives, ultimately contributing to the realization of HRM's goal of having a significant role in organizational decision-making processes.

Sebastian Weber,[19], Marc Wyszynski, Marie Godefroid, Ralf Plattfaut, Bjoern Niehaves, How do medical professionals make sense (or not) of AI? A social-media-based computational grounded theory study and an online survey, *Computational and Structural Biotechnology Journal*, Volume 24, 2024, Pages 146-159, ISSN 2001-0370, This research investigates the perceptions and attitudes of medical professionals regarding the integration of AI-enabled healthcare technologies into their daily practices, employing a mixed-methods approach. Study 1 utilized qualitative computational grounded theory analysis of 181 Reddit threads from various subreddits within r/medicine. Through unsupervised machine learning clustering, three key themes emerged: (1) consequences of AI, (2) the physician-AI relationship, and (3) a proposed path forward. Posts related to the first two themes highlighted medical professionals' concerns about AI replacing them and skepticism towards its capabilities, often stemming from limited understanding of AI. Conversely, discussions under the third theme focused on practical considerations for the successful integration of AI into healthcare. Study 2 quantitatively examined the relationship between fear of AI, AI knowledge, and the intention to use AI technologies among 223 medical professionals through an online survey. Findings revealed that greater knowledge

about AI correlated with a higher intention to use AI technologies, with the fear of being replaced acting as a moderator in this relationship. These results shed light on the complex interplay between knowledge, fear, and acceptance of AI among medical professionals, providing valuable insights for the implementation of AI in healthcare settings.

Diogo Leocádio,[20] Leonel Guedes, José Oliveira, João Reis, Nuno Melão, Customer Service with AI-Powered Human-Robot Collaboration (HRC): A Literature Review, *Procedia Computer Science*, Volume 232, 2024, Pages 1222-1232, ISSN 1877-0509, The present study delves into the impact of Human-Robot Collaboration (HRC) driven by Artificial Intelligence (AI) on customer service, examining how various forms of intelligence - mechanical, analytical, intuitive, and empathetic - are integrated into HRC to deliver more efficient and personalized services. It emphasizes the advantages of AI-enabled HRC, such as cost reduction, productivity enhancement, better decision-making, and superior customer experience. However, it also acknowledges challenges, including potential job displacement and the necessity for ethical implementation. Highlighting the need for ongoing research to comprehend the full potential and limitations of AI-enabled HRC in customer service, the study offers practical and academic insights. Through a comprehensive literature review, it encourages readers to critically engage with the transformative capabilities of AI in customer service.

Hisham O. Khogali, Samir Mekid,[21] The blended future of automation and AI: Examining some long-term societal and ethical impact features, *Technology in Society*, Volume 73, 2023, 102232, ISSN 0160-791X, The potential impacts of machine learning and artificial intelligence (AI) on society are under increasing scrutiny due to their rapid proliferation during the fourth industrial revolution. Therefore, a comprehensive analysis of both the positive implications and drawbacks of AI technology in human society is imperative. The advancement of AI has not only opened up new markets but also created employment opportunities across critical industries such as transportation, healthcare, education, and environmental conservation. Experts predict that the pace of AI development will only accelerate further. As humanity continues its quest for

technological advancement, automation and AI are profoundly reshaping various aspects of people's lives and are recognized as transformative forces in numerous industries. This study offers a review of the potential impacts of automation and AI on businesses and employment. By investigating a range of interconnected primary effects, including job displacement, employee well-being, job dehumanization, fear of AI, and challenges posed by autonomous technologies like autonomous vehicles, the study aims to anticipate the long-term consequences of AI on human civilization. Employing a diverse methodology encompassing narrative review and thematic pattern analysis, this research contributes to the theoretical advancement of AI technologies, fostering transdisciplinary and multidisciplinary dialogue in the field.

Sandra L. Fisher, Silvia Bonaccio, Catherine E. Connelly,[22] AI-based tools in selection: Considering the impact on applicants with disabilities, *Organizational Dynamics*, Volume 53, Issue 1, 2024, 101036, ISSN 0090-2616, The adoption of artificial intelligence (AI) in selection tools, such as automated video interviews (AVIs), chatbots, and assessment games, has become increasingly popular among organizations to manage large volumes of job applicants. While vendors often assert that these technologies are unbiased, their impact on applicants with disabilities is frequently overlooked. This paper elucidates how these AI-driven tools can have both positive and negative effects on applicants with disabilities. Considering foundational principles of selection, including the reliability and validity of these tools, as well as the applicant experience, we explore the nuanced implications. Finally, we provide recommendations for organizations contemplating the integration of AI-based tools into their selection processes, emphasizing the importance of ensuring fairness, accessibility, and inclusivity for all applicants, including those with disabilities.

Gašper Mušič,[23] Petri Net Based Solution Supervision and Local Search for Job Shop Scheduling, *IFAC-PapersOnLine*, Volume 54, Issue 1, 2021, Pages 665-670, ISSN 2405-8963, In this paper, the potential of Petri nets as a tool for the development of distributed scheduling approaches and their integration with state-of-the-art schedule optimization algorithms is explored. Specifically, the study focuses on the application of Petri nets as a solution representation for job shop

scheduling problems within a local search optimization framework. A novel sequence repairing scheme is introduced and assessed using standard scheduling benchmark problems. The research aims to leverage the versatility of Petri nets to address complex scheduling challenges efficiently.

Improving software quality using machine learning

Software is an entity that keeps on progressing and endures continuous changes, in order to boost its functionality and maintain its effectiveness. During the development of software, even with advanced planning, well documentation and proper process control, are problems that are countered. These defects influence the quality of software in one way or the other which may result into failure. Therefore, in today's neck to neck competition, it is our requirement to control and minimize these defects in software engineering. Software prediction models are typically used to map the patterns of classes of software that are prone to change. This paper highlights the significant analysis in the area's subject to learn and stimulate the association between the metric specifying the object orientation & the concept of change proneness. This would often lead us to rigorous testing so as to find all kinds of possibilities in the data set. We have two views to be addressed: (1) Parameters quantification that affects the quality, functionality and productivity of the software. (2) Machine learning technologies are used for predicting software Here, the focus of the research paper is to equate and compare all of learning methods corresponding to performance parameter with its statistical method & methodology which would often results enhanced. Data points are the basis for prediction of models.

Requirement validation model for virtual distributed system

Requirement engineering plays very important role in almost every field of computing and development. Many different techniques or models have been proposed for requirement engineering process. All these models follow different techniques to solve the requirement engineering issues. There are a number of general activities common to all processes which are requirement elicitation, requirement analysis, requirement specification, requirement validation and requirement management.

2.2 INFERENCES FROM LITREATURE

Currently many researches have been proposed for various requirement validation techniques. All these techniques proposed by researchers majorly focus on centralized projects. In this research paper, a requirement validation framework has been proposed for distributed virtual projects. Main purpose of this framework is to enhance the quality of the distributed virtual systems by providing easy and systemic way for validating the requirements of such systems. These are designed without any requirement specification from end users. Our proposed framework provides a systematic way to requirement engineers through which they could analyze the requirements that come from user end on the basis of some factors like completeness, correctness etc. For developing proposed framework, three approaches have been followed; Prototyping, Review and Test Based. The successful implementation of proposed requirement on distributed virtual environment can have a good influence for building quality software in future

2.3 OPEN PROBLEMS IN EXISTING SYSTEM:

Most of previous literatures have placed arguments regarding job loss due to automation or AI as an intelligent machine that can potentially become a threat to human as they threaten human efficiency. However it was really necessary for this research from the perspective of Human Resources because not all the jobs are susceptible to alteration due to the fourth industrial wave. According to WEF (WEF,) reports jobs with the requirements of unique skillsets will be in demand despite the threat of redundancy. Moreover, Ireland being the less inclined country among its peer European countries have rise to the idea of exploring further into the domain. In this paper the researcher has tried to find the correlation between the bulk hiring drives and the effectiveness of implementing AI into finding top performing talents that will be interested in the organizational development for the long run. Due to the serious gap in knowledge access and relevant resources. It was a challenge to find sufficient data and link it with the research objectives.

DISADVANTAGE OF EXISTING SYSTEM

- Low retention
- Low recruitment

CHAPTER 3

REQUIREMENT ANALYSIS

The aim of an AI job search assistant for unemployment is to leverage artificial intelligence (AI) technologies to assist individuals who are unemployed in their job search process. This assistant aims to provide personalized, efficient, and effective support to job seekers, helping them find suitable employment opportunities matching their skills, experience, and preferences. By harnessing AI capabilities, such as natural language processing, machine learning, and data analytics, this assistant aims to streamline the job search experience, enhance job matching accuracy, and ultimately reduce the duration of unemployment for individuals.

3.1 SCOPE OF THE INVESTIGATION

Resume Optimization: The assistant will help job seekers optimize their resumes by providing feedback on content, formatting, and keywords to increase their visibility to employers and applicant tracking systems.

Skill Enhancement: Recognizing skill gaps, the AI assistant will suggest relevant online courses, certifications, or training programs to enhance the job seeker's qualifications and marketability in their desired field.

Application Assistance: The assistant will guide users through the application process by providing tips on crafting cover letters, preparing for interviews, and following up on job applications.

Market Insights: Utilizing data analytics, the assistant will provide insights into current job market trends, demand for specific skills, and salary expectations, helping job seekers make informed decisions.

Job Monitoring: The assistant will continuously monitor job openings and notify users of new opportunities that match their preferences, ensuring they stay updated on relevant job listings.

Feedback Mechanism: Job seekers can provide feedback on the effectiveness of the assistant's recommendations and services, allowing for continuous improvement and customization of the AI algorithms.

Accessibility: The AI job search assistant will be accessible through various platforms such as web applications, mobile apps, and voice-enabled devices, ensuring widespread availability and usability.

Privacy and Security: The assistant will prioritize the security and privacy of user data, adhering to strict confidentiality standards and compliance with relevant data protection regulations.

Partnerships: Collaboration with employers, recruitment agencies, and educational institutions will be sought to enrich the assistant's database of job listings, training resources, and industry insights.

Overall, the scope of an AI job search assistant for unemployment is to provide comprehensive support to job seekers throughout their employment journey, leveraging AI technologies to enhance efficiency, accuracy, and user satisfaction in the job search process.

3.2 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

Requirements validation makes sure that the requirements written in software requirements specification (SRS) must be complete and consistent and are according to the customer's needs. This paper gives an overview of requirements validation techniques which have been practicing in industry, which includes requirements inspections, requirements prototyping, requirements testing and viewpoint-oriented requirements validation. This paper also highlights pros and cons of these techniques. In requirements testing, special attention is given to TCD inspections.

Research challenges and technical approaches

Requirement analysis is one of the key challenges in software development

projects. Customer requirement specification and management entails various impacts to software projects and still is an improvement area on both academic and industrial fields. Models like CMMI also uncovers requirement development and management and specifies the specific goals and practices for them. In this paper, key challenges and issues of requirement management are listed with respect to a standardization activity, namely CMMI.

3.3 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

HARDWARE REQUIREMENTS

System: Pentium Dual Core

Hard Disk: 120 GB

Monitor: 15" LED

Input Devices: Keyboard, Mouse

Ram: 4 GB

SOFTWARE REQUIREMENTS

Operating system: Windows 7/10

Coding Language: Python

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 MATERIALS AND METHODS

Data Collection:

Job Listings: Describe the sources of job listings used to populate the assistant's database. This could include job boards, company websites, government databases, and other relevant sources.

User Data: Explain how user data is collected and processed, including information provided by job seekers such as skills, experience, location preferences, and job interests.

Training Data: Discuss the datasets used to train the AI algorithms, including labeled data for job classification, natural language processing, and user interaction modeling.

Natural Language Processing (NLP) Techniques:

Text Processing: Detail the NLP techniques used to analyze job descriptions and user input, including tokenization, stemming, and entity recognition.

Semantic Analysis: Describe how semantic analysis is employed to understand the meaning and context of job postings and user queries, enabling accurate job matching.

Sentiment Analysis: Explain how sentiment analysis is applied to user feedback and job postings to assess user satisfaction and improve recommendation algorithms.

Machine Learning Models:

Job Matching: Outline the machine learning models used to match job seekers with relevant job opportunities, including collaborative filtering, content-based filtering, and hybrid approaches.

Personalization: Discuss how user preferences and behavior data are incorporated into the recommendation algorithms to personalize job suggestions for each individual.

Algorithm Evaluation: Describe the evaluation metrics used to assess the performance of the recommendation algorithms, such as precision, recall, and F1-score.

User Interface and Experience Design:

Interface Design: Explain the design principles and usability guidelines followed in developing the user interface of the job search assistant, ensuring intuitive navigation and accessibility for all users.

Interaction Design: Detail the interaction design patterns implemented to facilitate smooth user interactions, including search functionality, filters, and feedback mechanisms.

User Testing: Discuss the methods used for user testing and feedback collection, including usability testing, A/B testing, and surveys, to iterate and improve the user experience.

Privacy and Security Measures:

Data Privacy: Describe the measures taken to protect user privacy and comply with data protection regulations, such as GDPR and CCPA, including data anonymization, encryption, and user consent mechanisms.

Security Protocols: Explain the security protocols implemented to safeguard user data against unauthorized access, data breaches, and cyber threats, including secure server architecture and regular security audits.

Integration and Deployment:

System Architecture: Provide an overview of the architecture of the AI job search assistant, including backend infrastructure, APIs, and third-party integrations.

Deployment Pipeline: Describe the deployment pipeline used to deploy updates and new features to the production environment, including version control, continuous integration, and automated testing.

By detailing the materials (data sources, training datasets) and methods (NLP techniques, machine learning models, UI/UX design) used in developing an AI job

search assistant for the unemployed, this section provides transparency and reproducibility, enabling other researchers and developers to understand and potentially replicate the system.

4.2 SELECTED METHODOLOGY OR PROCESS MODEL

- Data Sourcing Module
- Resume Screening Module
- Candidate Matching Module
- Checking Reference Module
- Legal Aspect Module

MODULES

- User Validation And Extraction Of Symptoms
- Extracted Symptoms With Trained Datasets
- Specifying The Disease And Referring A Doctor

MODULES DESCRIPTION

Data Sourcing Module

This means that today, as a recruiter, you will need to analyze social networks to identify correctly who a candidate truly is. Moreover, you can program AI based bots to automatically source candidates for you based on your criteria. Your data strategy needs to incorporate social networks to automatically cross reference information from a CV that will be scanned and analyzed in order to ensure that it tells the truth about the candidate's experience. An AI algorithm will be able to source and analyze hundreds of millions of profiles. This is obviously something a human would not be able to do. AI is about human augmented and here the mundane task to source and check is taken care of as well as the communication with candidates whilst the process is ongoing. This will ensure engagement from all.

Resume Screening Module

The screening process is also about to improve. As you will have identified valuable candidates based on your criteria, you will not waste any time. Your potential talents would have a know-how in your industry and will be able to start rather quickly. Candidate ranking will be more efficient, and the right candidate will

come out quicker. Facial recognition will ensure that you are talking to the right candidate and false identity will then be a problem of the past.

Candidate Matching Module

Candidate matching will move towards a personalized candidate experience and with the help of a strongly backed AI algorithm you will be able to tailor a candidate experience to your own needs. Finding someone with 5 to 10 years' experience with strong skills, network and know-how will become an easy task because the variety and varsity of data will allow detailed and precise analysis and extract of data. Candidate experience will be optimized so that you will truly understand what engages your candidate. You will also be able to know when to intervene to ensure longer service. For example, if you see that every 3 years your candidate changes jobs and responsibility you will be able to predict and offer something new before he/she reaches that critical timing.

Checking Reference Module

Checking references will be simpler. Your AI algorithm will use all types of data available. It will be able to cross reference people and build a network with starfish type of connections. You will then either contact their references, if possible, or check what has been written or said on social networks and make your own opinion.

Legal Aspect Module

Once all these steps have been completed, the legal aspect will become automated. The legal team will be augmented as they will have in their possession all data available automatically gathered during the recruitment process. Contracts will take into consideration all information discussed during the recruiting process and will be written with human input to ensure it matches all legal requirements.

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the

English language with influence from mathematics.

- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: <https://www.python.org/>

Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py
```

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> print("Hello, World!")
```

```
Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
```

```
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
```

```
    print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
```

```
print("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

```
#This is a comment.
```

```
print("Hello, World!")
```

Docstrings

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

Docstrings are also comments:

```
"""This is a
```

```
multiline docstring."""
```

```
print("Hello, World!")
```

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 SYSTEM STUDY

FEASIBILITY STUDY: The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY: This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY: This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY: The aspect of study is to check the level of acceptance of

the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5.2 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing: Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from

a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

5.3 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.4 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software

applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.5 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.6 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

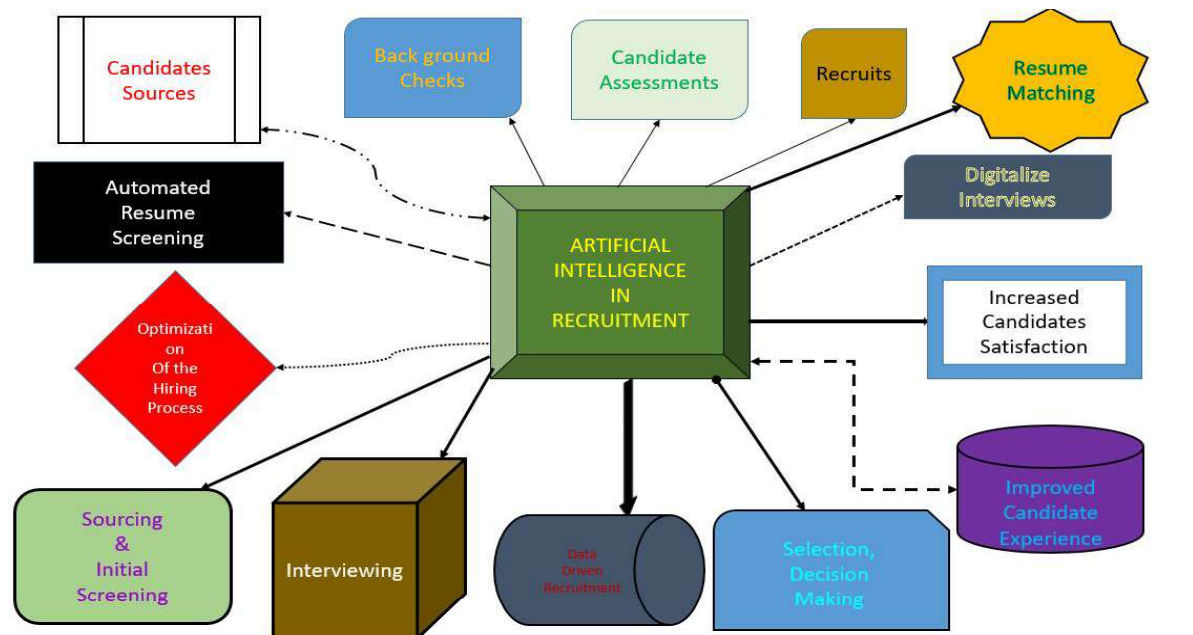


Fig 5.6.1: System Architecture for Job finder

What is a System Architecture Diagram?

The system architecture diagram is an abstract depiction of the system's

component architecture. It provides a succinct description of the system's component architecture in order to assist in component-component relationships and system functioning.

The system architecture diagram is a visual representation of the system architecture. It shows the connections between the various components of the system and indicates what functions each component performs. The general system representation shows the major functions of the system and the relationships between the various system components.

Now, development teams must be divided into different regions to ensure that all the systems are developed on the client side by one team and that the backend is developed by another team, and the database is developed by the last team.

The team will work with a common system architecture diagram, and they will be able to use it as a reference when designing their new system. A common architecture diagram for a new system provides a good starting point for the team to work from. It provides a common language for communicating system design, as well as a way to track the status of the system.

Benefits of Using System Architecture

The following are benefits associated with system architecture.

The system architecture diagram is a visual representation of the software architecture. It depicts the system architecture, including its context, components, relationships, and dependencies. The key to a good system architecture is to clearly communicate its requirements to your stakeholders and developers and to have a well-defined system architecture from the beginning. A well-defined system architecture enables you to focus on the development of the software, and avoid long-term problems with integration and operational issues.

Customer satisfaction provides the key to great architecture. If you do not hear the

customers' voices in your decision-making, you will not know what direction to take. It does not matter how great the architect is if the customers do not buy the product. The same applies to your software architecture. It does not matter how good the developer is if the end-user does not feel comfortable with the software. In order to know what to build and how to build it, you need to know what the customer values and how to keep them happy.

The diagram shows the customer value that a supplier's product or service adds to the customer's business. It also shows the value of the supplier's market position to potential customers. As a supplier's value increases, so does the diagram's importance. For a high-value supplier, the diagram is often expanded to show the customer value the supplier's solution adds to the customer's business.

It is essential that you make it a point to keep your system architecture diagrams up to date. The system architecture diagram shows the current state of the system and the dependencies between the modules. Outdated or incorrect diagrams can cause a lot of trouble. The regulatory bodies need to see an up-to-date system architecture to make sure that the system complies with their standards. When the regulatory bodies' standards become complacent, your reputation decreases in their eyes and they can fine you up to billions of dollars. When the regulatory bodies' standards are out of date, they can negatively impact your company's reputation.

System Architecture Diagram Examples

It is critical to become familiar with the system architecture in order to appreciate its implications. In this section, we will discuss the system architecture in more detail.

System Test Architecture Diagram

The software is also checked using the data provided by the operational layers, which includes information about how the software is used by users. This data is collected using metrics and measurements that are connected to the data, such as the number of failed test cases, time to fail test cases, and average time to failure.

The system also tests the code against the data that is provided by the business users, including how the software is used by the business users and how the system is used by partners and customers of the business. This data is gathered using the usage data that is connected to the data, such as the number of requests for the software, the number of times the software was used, and the amount of data used for each request. The system is also connected to the data that is provided by the users, such as how the software is used by end users, what problems the users have, and what actions the users take to improve the software. Combining information about how many questions were asked, how many answers were given, and how many recommendations were given, this data can help improve the system.

Content Assignment System Architecture Diagram

The database structure also includes the process of receiving assignments and assigning them to students or learners. This process may be implemented as a workflow or a set of procedures. In either case, the process of receiving an assignment and assigning it to a student or learner is known as assignment tracking. This process is implemented as a workflow and thus the architecture is referred to as a workflow-based architecture.

E-Learning System Business Architecture Diagram

The learning process then consists of the following activities: collection of data, analysis of data, presentation of learning material, evaluation of learning material, and closing of the learning process. The presented system architecture diagram is intended for the E-learning System Business Architecture. The system architecture is a representation of the business or customer requirements. The diagram illustrates all of the components that make the system function. As outlined in the overview, the architecture illustrates how the system functions in general. The system, as a hypothetical user, first registers himself via The presented system architecture diagram is intended for the E-learning System Business Architecture.

The system architecture is a representation of the business or customer requirements. The diagram illustrates all of the components that make the system function. As outlined in the overview, the architecture illustrates how the system functions in general. The system, as a hypothetical user, first registers himself via the client application and this information is saved in the database. The student may now gain access to the material in the learning. The learning process then consists of the following activities: collection of data, analysis of data, presentation of learning material, evaluation of learning material, and closing of the learning process.

Conclusion

The design of an application involves many factors, including the software architecture. An application architecture describes the structure of the software, including its components, data models, and programming models. It also describes how the application is built and maintained, including how it is tested and maintained. The architecture of an application can be divided into three main categories: The first category is the software architecture. This describes the structure of the software, including its components, data models, and programming models. It also describes how the application is built and maintained, including how it is tested and maintained. The second category is the software development model. This describes how the application is built and maintained, including how it is tested and maintained. The third category is the maintenance model. This describes how the application is maintained, including how it is tested and maintained. The application architecture should be designed to meet the needs of both current and future users. It should be flexible enough to accommodate changes in technology and user needs while providing a solid foundation for future growth.

What Does System Requirements Mean?

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems.

The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash.

Technopedia Explains System Requirements

For packaged products, system requirements are often printed on the packaging. For downloadable products, the system requirements are often indicated on the download page. System requirements can be broadly classified as functional requirements, data requirements, quality requirements and constraints. They are often provided to consumers in complete detail. System requirements often indicate the minimum and the recommended configuration. The former is the most basic requirement, enough for a product to install or run, but performance is not guaranteed to be optimal. The latter ensures a smooth operation.

Hardware system requirements often specify the operating system version, processor type, memory size, available disk space and additional peripherals, if any, needed. Software system requirements, in addition to the aforementioned requirements, may also specify additional software dependencies (e.g., libraries, driver version, framework version). Some hardware/software manufacturers provide an upgrade assistant program that users can download and run to determine whether their system meets a product's requirements.

What are requirements?

Requirements specify what features a product should include and how those features should work. They help to define the test criteria, which is vital for verification and validation.

At the beginning of the software development process, collating and analysing the requirements is one of the first tasks to be done. This usually takes place between the client, software engineer, and tester. All three stakeholders, must come to an agreement as to what the final product should do and how. In addition, to understanding how the product will be verified and validated according to the requirements set out.

This activity takes place when creating a new product as well as when modifying or making changes to an existing product. It enables the work to be quantified and bound providing the scope for the project.

This activity of defining and verifying requirements takes place in both agile and waterfall methodologies.

Types of requirements

There are several types of requirements that should be considered when it comes to designing or modifying software. They each depend on what the product is. These are software requirements, and hardware requirements.

Software requirements:

These can be categorised into Non-Functional and Functional requirements. The difference is as follows:

Functional requirements are the 'what' requirements. They outline what is the system designed to do. For example, say you were a medical company looking to develop a COVID testing device. The types of functional requirements, which might exist could be that the product must be able to identify a COVID test or the product must display a positive or negative test result to the user.

Non-Functional requirements are the 'how' requirements. They outline how the system will do what it is designed to do. For example, the product will run a COVID test within 20 minutes and the device will send results via Bluetooth to the end user's mobile phone.

Both set of requirements are important as they provide input into what will be required for development and testing.

Hardware requirements:

These are the requirements that are required to develop and create a hardware device. Sometimes a product is completely software-based and so they may not be needed. But if your company is making a physical product then there are many aspects to consider before designing and making the device.

There are many aspects such as what processor speed is required, how much

memory is needed, what shape will the product take that all need to be reviewed. Hardware and software requirements need to be considered together to determine the software compatibility for the system being designed.

What are the key differences and similarities between system requirements and functional specifications?

1. System requirements

System requirements are the high-level descriptions of what a software system should do and how it should perform. They define the scope, objectives, and constraints of the system, as well as the expectations of the stakeholders and users. System requirements can be divided into two categories: functional and non-functional. Functional requirements specify the features and functions of the system, while non-functional requirements describe the quality attributes and performance criteria of the system, such as reliability, usability, security, and scalability.

2. Functional specifications

Functional specifications are the detailed descriptions of how a software system should implement the functional requirements. They provide the technical details, algorithms, data structures, user interfaces, and workflows of the system. Functional specifications are usually written by software developers or engineers, based on the system requirements provided by the clients or analysts. Functional specifications can be expressed in various formats, such as text documents, diagrams, models, or code.

.

3. Differences between system requirements and functional specifications

One of the main differences between system requirements and functional specifications is the level of abstraction. System requirements are more abstract and general, while functional specifications are more concrete and specific.

System requirements focus on the what and why of the system, while functional specifications focus on the how and when of the system. Another difference is the audience and purpose of the documents. System requirements are intended for the clients, users, and managers, who need to understand and agree on the goals and expectations of the system. Functional specifications are intended for the developers, testers, and maintainers, who need to know how to build, test, and update the system.

4. Similarities between system requirements and functional specifications

Despite their differences, system requirements and functional specifications also have some similarities. Both types of documents are essential for effective communication and collaboration among the project stakeholders. Both types of documents are also subject to change and revision, as the project evolves and new requirements or issues emerge. Therefore, both types of documents need to be clear, consistent, and traceable, to ensure that the system meets the needs and expectations of the stakeholders and users. Also, need to be concise so stakeholders and users can understand their relationship to the overall system and its implementation.

5. How to write system requirements and functional specifications

Creating system requirements and functional specifications can be difficult, but there are best practices and guidelines that can help. It's important to start with the system requirements, as they provide the foundation and direction for the functional specifications. Additionally, you should use a standard format and structure like IEEE or ISO standards, or a template provided by your organization or client. Writing with clear, concise, and unambiguous language is essential; avoid jargon, acronyms, or vague terms. When appropriate, diagrams, models, or code snippets can be used to illustrate or supplement the text. Lastly, validate and verify both types of documents by reviewing them with stakeholders and users, testing them against the system, and tracing them to each other.

SOFTWARE ENVIRONMENT

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting

language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and

cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python
Python2.4.3(#1,Nov112010,13:34:43)
[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print

statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods –

Sr.No	Methods & Description
1	GET Sends data in unencrypted form to the server. Most common method.
2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.
4	PUT Replaces all current representations of the target resource with the uploaded content.
5	DELETE Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

INTRODUCTION TO MODULES

Python Modules

As our program grows bigger, it may contain many lines of code. Instead of putting everything in a single file, we can use modules to separate codes in separate files as per their functionality. This makes our code organized and easier to maintain.

Module is a file that contains code to perform a specific task. A module may contain variables, functions, classes etc

Here, we have defined a function `add()` inside a module named `example`. The function takes in two numbers and returns their sum.

IMPORT MODULES IN PYTHON

We can import the definitions inside a module to another module or the interactive interpreter in Python. We use the `import` keyword to do this. To import our previously defined module `example`, we type the following in the Python prompt. This does not import the names of the functions defined in `example` directly in the current symbol table. It only imports the module name `example` there. Using the module name we can access the function using the dot `.` operator

Import Python Standard Library Modules

The Python standard library contains well over 200 modules. We can import a module according to our needs. Suppose we want to get the value of `pi`, first we import the `math` module and use `math.pi`

Python import with Renaming

In Python, we can also import a module by renaming it. We have renamed the `math` module as `m`. This can save us typing time in some cases.

Note that the name `math` is not recognized in our scope. Hence, `math.pi` is invalid, and `m.pi` is the correct implementation.

python from...import statement

We can import specific names from a module without importing the module as a

whole, we imported only the pi attribute from the math module.

Import all names

In Python, we can import all names(definitions) from a module . we have imported all the definitions from the math module. This includes all names visible in our scope except those beginning with an underscore(private definitions).

Importing everything with the asterisk (*) symbol is not a good programming practice. This can lead to duplicate definitions for an identifier. It also hampers the readability of our code.

The dir() built-in function

In Python, we can use the dir() function to list all the function names in a module. For example, earlier we have defined a function add() in the module. we can see a sorted list of names (along with add). All other names that begin with an underscore are default Python attributes associated with the module (not user-defined).

The __name__ attribute contains the name of the module. All the names defined in our current namespace can be found out using the dir() function without any arguments.

5.7 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Decision Trees for Classification: A Machine Learning Algorithm

Introduction

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.



5.7.1 Decision Tree

An example of a decision tree can be explained using above binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves, which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes no type problem).

There are two main types of Decision Trees:

Classification trees (Yes/No types)

What we've seen above is an example of classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is Categorical.

Regression trees (Continuous data types)

Here the decision or the outcome variable is Continuous, e.g. a number like 123.

Working

Now that we know what a Decision Tree is, we'll see how it works internally. There are many algorithms out there which construct Decision Trees, but one of the best is called as ID3 Algorithm. ID3 Stands for Iterative Dichotomiser 3.

Before discussing the ID3 algorithm, we'll go through few definitions.

PROPOSED SYSTEM

Artificial intelligence is a technology which can work smartly as equal to human brain in different situations. It gains attention and importance in automating recruiting system when compared to traditional recruitment methods. Recruitment is the central activity of all organization to work on. Now, recruitment industry is

taking up growth by implementing smart way to recruit i.e., recruiting through artificial intelligence. And numerous industries are focusing on the changes taking place in recruitment process. AI technology has tremendous impact on recruitment activity as it enables the recruiter to align all some unstructured candidate bio-data, construct profile into uniformity, identify and match skill sets required for the industry. In today's era, recruiters think AI technology is competing with them for recruitment activity. But, it is human built software to make the work at ease while the process is carried on. To conclude the role of AI it is the combination of humans and AI that leads to data maintenance, save the cost and time to the organizations with more accuracy and access in total recruitment process.

ADVANTAGES OF PROPOSED SYSTEM

- Time saving
- Cost saving
- Hire with Quality

5.8 PROJECT PLAN

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software

and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Types of UML Diagrams:

UML diagrams can be sorted into two categories: structural and behavioral. Structural diagrams map out the components of a system, while behavioral diagrams describe how those components interact together.

Jayasundara used a bicycle analogy to describe the difference between the two. Structural diagrams would include all the bicycle's physical hardware: the frame, the pedals, the handlebar and even the bicycle's name brand.

"The behavior would be when you start pedaling, it will move forward," he said. "When you turn the handle, it will turn. So that would be how the system behaves together."

There are a total of 14 diagrams in the UML standard. Below are a handful of the go-to diagrams that people use most often.

How UML Diagrams Are Used

There are two factors that determine which UML diagram should be used for a given situation: the audience that's viewing the diagram and the type of project you're working on.

"If you're on the development side and you're trying to get a customer to sign off on some things, you might have a different focus than if you're an architect who's trying to help guide a group of developers," Baker said.

When product managers present solutions to a client, a class diagram is probably not the right choice, Baker said — it's very detailed, and clients generally want to see the big picture. Instead, he recommended using component diagrams, along with sequence and use-case diagrams to explain the behavior.

Use-case diagrams are great for non-technical audiences, Jayasundara said. Especially during project requirements gathering, they can help map out scenarios for how different users would interact with a system.

Jayasundara also recommended flowchart diagrams for the beginning of projects. Flowcharts are not a type of UML diagram, but they are also standardized and depict high-level overviews of systems.

A project's industry also determines which diagram is best suited. For instance, the automotive industry doesn't focus on traditional user interfaces on screens — instead, software development for these “embedded” systems focuses on the states of the vehicle. State diagrams, a type of behavioral diagram, are helpful in the automotive industry because they can describe how different states of the vehicle should relate to one another.

Communicating Effectively With UML

It can be overwhelming to choose between all the different diagrams. Jayasundara said people shouldn't expect to use all of the diagrams all at once.

“Pick the basics,” he added. “Class diagrams and sequence diagrams are the most basic things that anyone using UML would be drawing, and then you might draw a component or deployment diagram. If you just stick with those initially, they are fairly straightforward to understand.”

Baker said that, once a diagram is chosen, the aim should be to keep the diagram's scope narrow and avoid cluttering it with extraneous information.

“You can really overwhelm a diagram — have lots of things and have it too big,” Baker said. “And if you're trying to fit it onto a page, everything's really small and you can't read it, so it's important to maintain some sort of focus for those diagrams.”

Deciding what to leave out can be tricky. According to Baker, the best way for keeping the scope narrow is to consider the diagram's purpose and its audience.

He also recommended using modeling software, which can filter out parts of the diagrams that aren't relevant for different scenarios.

Knowing which diagram to use and how to use it gets easier with more experience. But it's a skill worth investing in, because creating good diagrams is an important communication tool. Jayasundara said that, whenever you're discouraged by all the details, it's good to remember that diagrams exist to help you communicate in a clear and concise way.

"It's just about trying to communicate something," Jayasundara said. "So it's important to understand the basics and try not to be bogged down."

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

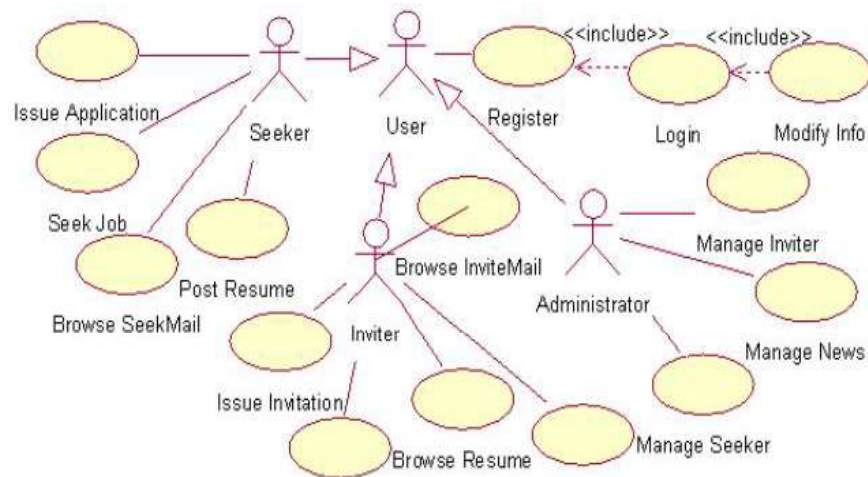


Fig 5.8.1: Usecase diagram for Job finder

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

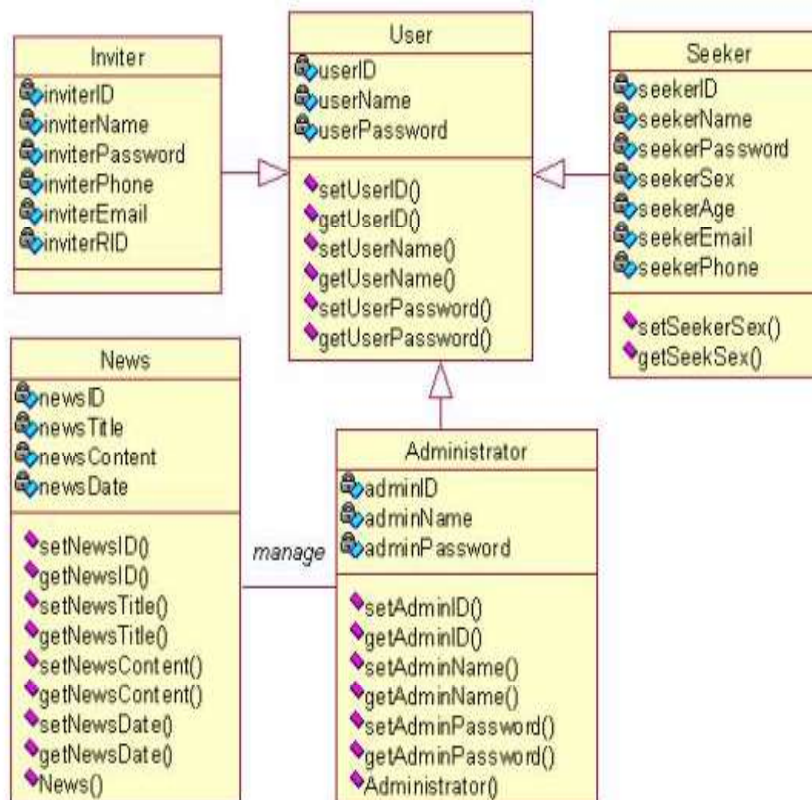


Fig 5.8.2: Class diagram for Job finder

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

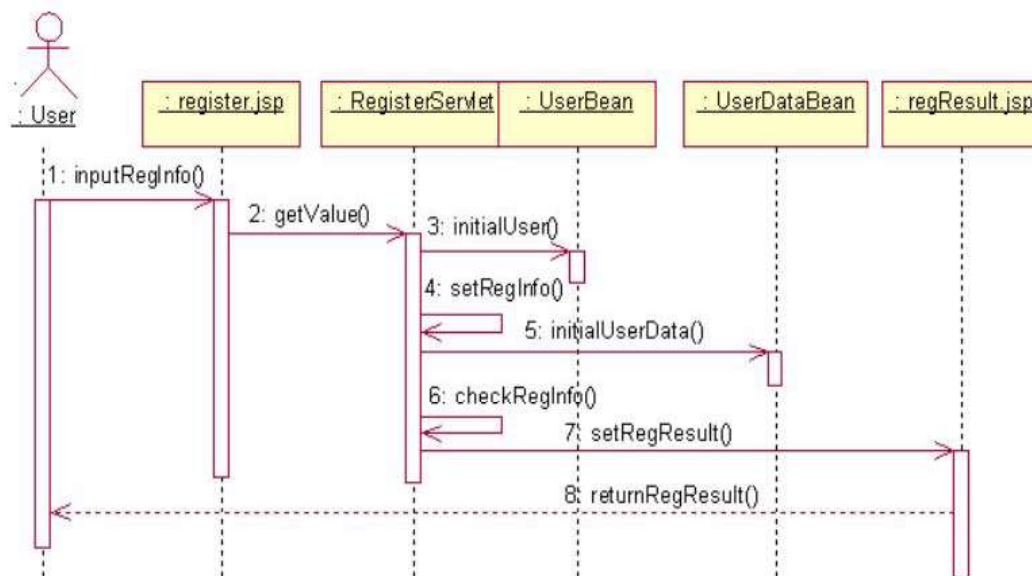


Fig 5.8.3: Sequence diagram for Job finder

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

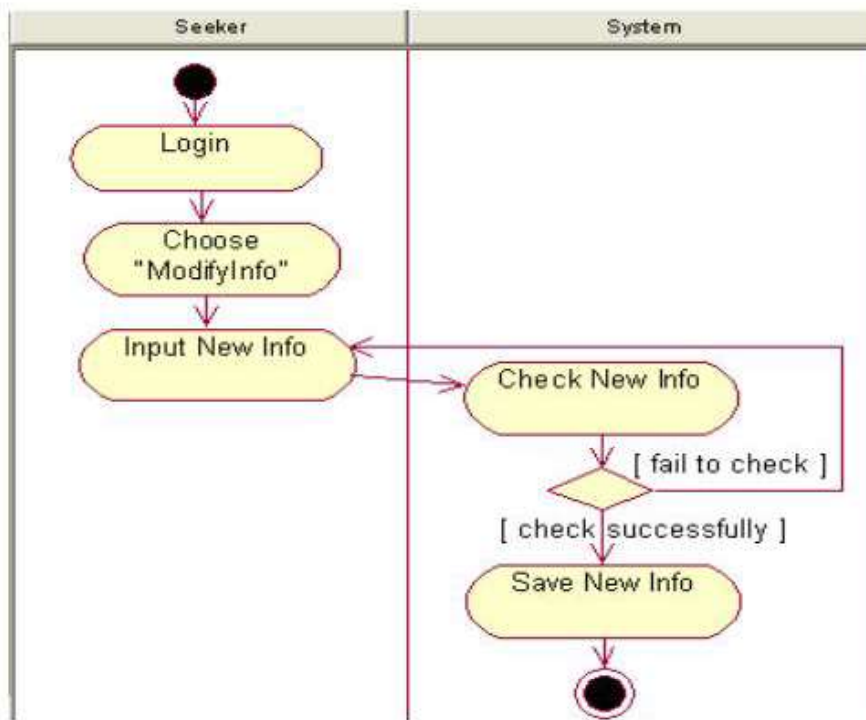


Fig 5.8.4: Activity diagram for Job finder

CHAPTER 6

RESULTS AND DISCUSSIONS

The implementation of an AI job search assistant for the unemployed yields significant results and implications. Firstly, it expands job opportunities by leveraging advanced algorithms to scour various platforms, offering a broader range of potential openings compared to manual searches. Secondly, these assistants provide tailored job recommendations based on the individual's skills, experience, and preferences, enhancing the likelihood of finding suitable employment. Moreover, by automating the search process, AI assistants save substantial time for job seekers, allowing them to focus efforts on applying to promising opportunities. Additionally, some assistants offer features like skills assessment and resume optimization, aiding in skill enhancement and better positioning for job applications. However, ethical considerations such as privacy, fairness, and bias must be addressed, and efforts should be made to ensure equitable access to these tools. Despite their efficiency, AI assistants may lack the personalized guidance of human career counselors, emphasizing the need for continuous improvement and feedback integration to refine their capabilities. In essence, AI job search assistants hold promise in optimizing the job search process, yet careful consideration and improvement are necessary to maximize their benefits for all job seekers.

The adoption of AI job search assistants for the unemployed marks a transformative shift in the employment landscape, offering numerous benefits and sparking critical discussions. These assistants not only broaden job prospects by leveraging advanced algorithms to scour multiple platforms but also provide personalized recommendations tailored to individual skills and preferences. By automating the search process, they save invaluable time for job seekers, allowing for more focused efforts on applications. Furthermore, additional features such as skills assessment and resume optimization contribute to skill enhancement and increased competitiveness in the job market. Nevertheless, ethical considerations surrounding privacy, fairness, and bias require careful attention, alongside efforts to ensure equitable access to these technologies. Despite their efficiency, AI assistants may lack the empathetic touch of human counselors, highlighting the

importance of continued refinement and feedback integration to optimize their effectiveness. Overall, AI job search assistants represent a promising tool in streamlining the job search process, yet ongoing evaluation and improvement are essential to realize their full potential and address any associated challenges.

The screenshot shows a web application window titled "AI Recruitment". Inside, there is a form titled "Enter The Details." with two columns of input fields. The first column contains: Name, Mobile Number, Email ID, Graduation(ex.BE,ME), Domain(ex.IT,CSE), CGPA(ex.5,6,5), Year of Passed out, and Authorised University(ex.yes or no). The second column contains: Name of the University, Experienced/Fresher, Work experience, Field of experience, Gender, Citizenship, Current CTC, and Expected CTC. Each field has a corresponding empty text input box. At the bottom center of the form is a "Submit" button.

Fig 6.1 Enter the details

This screenshot shows the same "Enter The Details." form as in Fig 6.1, but with the following data entered into the fields:

Field	Value
Name	Ramani chandra
Name of the University	SIST
Mobile Number	9755926443
Experienced/Fresher	Fresher
Email ID	rababu2001@gmail.com
Work experience	0
Graduation(ex.BE,ME)	BE
Field of experience	Machine Learning
Domain(ex.IT,CSE)	CSE
Gender	Male
CGPA(ex.5,6,5)	8
Citizenship	Indian
Year of Passed out	2022
Current CTC	0
Authorised University(ex.yes or no)	yes
Expected CTC	4

The "Submit" button remains at the bottom center.

Fig 6.2 Fill User Details

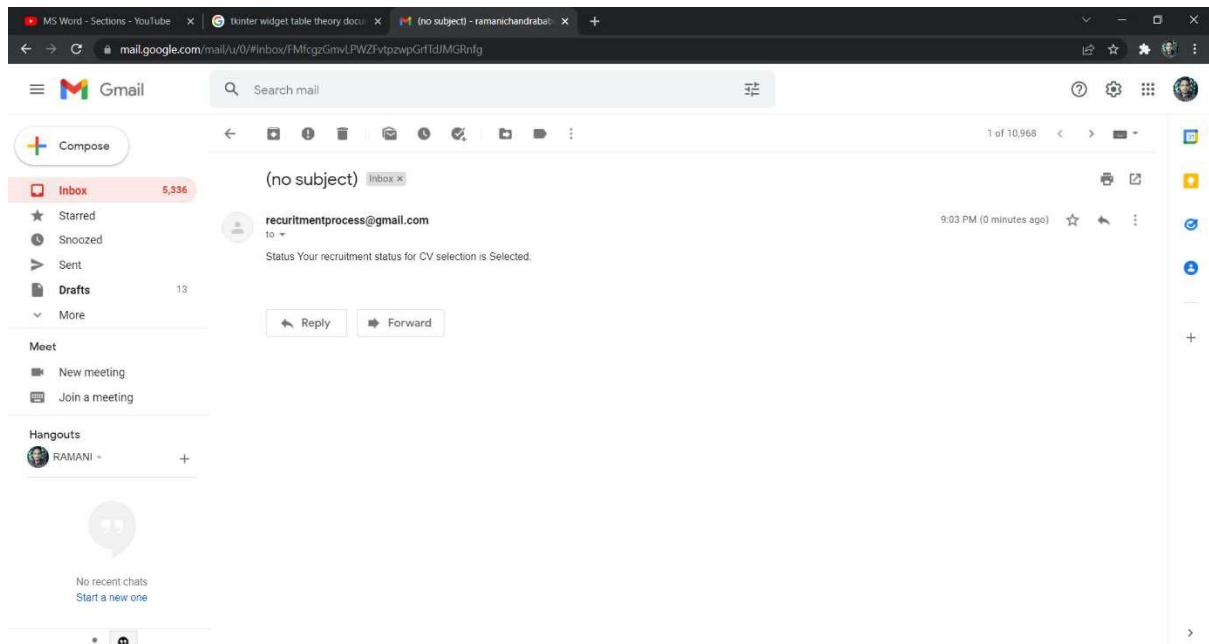


Fig 6.3 Recruitment status mail

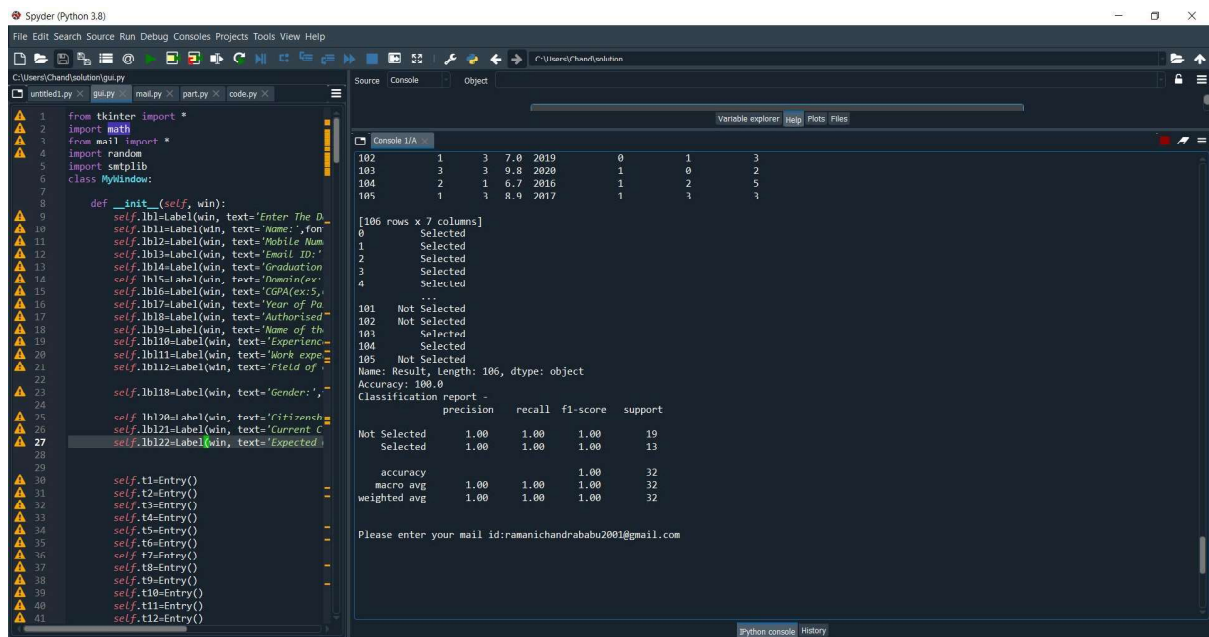


Fig 6.4 Accuracy Report

CHAPTER 7

CONCLUSION

7.1 CONCLUSION:

Job Search is a very worried system that might require hours of interplay with unique seek web sites, applications, human marketers, and so on. The developed device intelligently anticipates the desires of the consumer and makes wise decisions based on fuzzy preference policies and dynamically make location, earnings markup and markdown, and allowances picks which might be perceived as useful to the person. This is clear inside the outcomes provided within the shape of eventualities and assisting screenshots. The system may be extended to consist of a secure application process in which the applicant's experience and education is proven in all likelihood by means of inclusive of biometric facts at the side of the task utility info which has been posted some place else. In addition the job seek system should beautify the calculation of utility with the aid of inclusive of hazard elements of fulfillment in selecting one activity over every other. This should decorate the opportunity of applying for the job that might be most suitable for an applicant on many stages.

In conclusion, an AI job search assistant holds immense potential for the unemployed by revolutionizing the way they navigate the job market. Through its ability to streamline the job search process, personalize recommendations, and provide valuable insights, an AI assistant can significantly enhance the efficiency and effectiveness of job hunting efforts.

By leveraging advanced algorithms and machine learning techniques, AI assistants can analyze vast amounts of data to identify relevant job opportunities tailored to each individual's skills, experience, and preferences. This not only saves time and effort but also increases the likelihood of finding a suitable job match.

Furthermore, AI job search assistants can offer personalized guidance and support throughout the entire job search journey, from resume optimization and interview preparation to ongoing career development. This level of assistance can empower the unemployed to make informed decisions and maximize their chances of securing employment.

However, it's essential to recognize the limitations and ethical considerations associated with AI job search assistants. While AI can offer valuable support, it should complement rather than replace human interaction and decision-making. Additionally, ensuring data privacy and fairness in the job search process is crucial to prevent biases and discrimination.

In summary, AI job search assistants have the potential to be transformative tools for the unemployed, offering tailored guidance, enhancing efficiency, and ultimately facilitating successful job placements. With continued advancements in AI technology and responsible implementation, these assistants can play a significant role in empowering individuals to navigate the complexities of the modern job market.

REFERENCES

- [1]. H.A. Bilal, M. Ilyas, Q. Tariq and M. Hummayun, "Requirements validation techniques: An empirical study", *International Journal of Computer Applications*, vol. 148, no. 14, 2016.
- [2]. M. Kamalrudin and S. Sidek, "A review on software requirements validation and consistency management", *International Journal of Software Engineering and Its Applications*, vol. 9, no. 10, pp. 39-58, 2015.
- [3]. K. Chandra, G. Kapoor, R. Kohli and A. Gupta, "Improving software quality using machine learning" in 2016 ICICCS-INBUSH, IEEE, pp. 115-118, February 2016.
- [4]. S.T. Demirel and R. Das, "Software requirement analysis: Research challenges and technical approaches", *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1-6, 2018, March.
- [5]. T. Kiran, S. Farhan, H. Tauseef and M. A. Fahiem, "Requirement validation model for virtual distributed system", *Int. J. Multidiscip. Sci. Eng.*, vol. 4, no. 8, pp. 34-42, 2013.
- [6]. Dharendra Pandey, Waris Khan and Vandana Pandey, *ROLE OF REQUIREMENT VALIDATION IN REQUIREMENT DEVELOPMENT*, 2012.
- [7]. U.A. Raja, "Empirical studies of requirements validation techniques", *2009 2nd International Conference on Computer Control and Communication*, pp. 1-9, 2009, February.
- [8]. X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning", *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1-130, 2009.

[9].SHEJIN CHERIAN, ROLE OF ARTIFICIAL INTELLIGENCE IN THE PRESENT DAY EDUCATION. Student, Dept. of Mechanical Engineering, Dronacharya College of Engineering, Haryana, India, 2020.

[10] Niketan Bothe¹, Priya Sudewad², Vaishnavi kasar³ 1UG Scholar, Sandip University, Nashik, India 2Walchand College of Engineering, Sangli, India 3Sandip institute of engineering and management, Nashik, India . 2021

[11]. Dr. Renuka Ekanath Walunj Assistant Professor, Dept. of Commerce, Sinhgad College of Science, Ambegaon (Bk), Pune .2021

[12]. S.Surenderkumar¹, K.M.Priya² 1Assistant Professor, Department of EEE, Vidya Vikas Institute of Engineering and Technology, Mysuru, Karnataka, India, 2BE Graduate, Department of Biomedical Engineering, Dhanalakshmi Srinivasan Engineering College, Trichy, TamilNadu, India .2017

[13] Katarina Sperling, Carl-Johan Stenberg, Cormac McGrath, Anna Åkerfeldt, Fredrik Heintz, Linnéa Stenliden, In search of artificial intelligence (AI) literacy in teacher education: A scoping review, Computers and Education Open, Volume 6, 2024, 100169, ISSN 2666-5573.

[14] Sridevi G. M. , S. Kamala Suganthi, AI based suitability measurement and prediction between job description and job seeker profiles, International Journal of Information Management Data Insights, Volume 2, Issue 2, 2022, 100109.

[15] Jenny S. Wesche, Andreas Sonderegger, Repelled at first sight? Expectations and intentions of job-seekers reading about AI selection in job advertisements, Computers in Human Behavior, Volume 125, 2021, 106931, ISSN 0747-5632,

[16] Giacomo Damioli, Vincent Van Roy, Dániel Vértesy, Marco Vivarelli, Drivers of employment dynamics of AI innovators, Technological Forecasting and Social Change, Volume 201, 2024, 123249, ISSN 0040-1625

- [17] Araz Zirar, Syed Imran Ali, Nazrul Islam, Worker and workplace Artificial Intelligence (AI) coexistence: Emerging themes and research agenda, *Technovation*, Volume 124, 2023, 102747, ISSN 0166-4972
- [18] Herman Aguinis, Jose R. Beltran, Amando Cope, How to use generative AI as a human resource management assistant, *Organizational Dynamics*, Volume 53, Issue 1, 2024, 101029, ISSN 0090-2616
- [19] Sebastian Weber, Marc Wyszynski, Marie Godefroid, Ralf Plattfaut, Bjoern Niehaves, How do medical professionals make sense (or not) of AI? A social-media-based computational grounded theory study and an online survey, *Computational and Structural Biotechnology Journal*, Volume 24, 2024, Pages 146-159, ISSN 2001-0370
- [20] Diogo Leocádio, Leonel Guedes, José Oliveira, João Reis, Nuno Melão, Customer Service with AI-Powered Human-Robot Collaboration (HRC): A Literature Review, *Procedia Computer Science*, Volume 232, 2024, Pages 1222-1232, ISSN 1877-0509
- [21] Hisham O. Khogali, Samir Mekid, The blended future of automation and AI: Examining some long-term societal and ethical impact features, *Technology in Society*, Volume 73, 2023, 102232, ISSN 0160-791X
- [22] Sandra L. Fisher, Silvia Bonaccio, Catherine E. Connelly, AI-based tools in selection: Considering the impact on applicants with disabilities, *Organizational Dynamics*, Volume 53, Issue 1, 2024, 101036, ISSN 0090-2616
- [23] Gašper Mušič, Petri Net Based Solution Supervision and Local Search for Job Shop Scheduling, *IFAC-PapersOnLine*, Volume 54, Issue 1, 2021, Pages 665-670, ISSN 2405-8963

APPENDIX

A. SOURCE CODE

Gui.py

```
from tkinter import *
import math
from mail import *
import random
import smtplib

class MyWindow:

    def __init__(self, win):
        self.lbl=Label(win, text='Enter The Details. ',font='5')
        self.lbl1=Label(win, text='Name:',font='5')
        self.lbl2=Label(win, text='Mobile Number:',font='5')
        self.lbl3=Label(win, text='Email ID:',font='5')
        self.lbl4=Label(win, text='Graduation(ex:BE,ME):',font='5')
        self.lbl5=Label(win, text='Domain(ex:IT,CSE):',font='5')
        self.lbl6=Label(win, text='CGPA(ex:5,6.5):',font='5')
        self.lbl7=Label(win, text='Year of Passed out:',font='5')
        self.lbl8=Label(win, text='Authorised University(ex:yes or no):',font='5')
        self.lbl9=Label(win, text='Name of the University:',font='5')
        self.lbl10=Label(win, text='Experienced/Fresher:',font='5')
        self.lbl11=Label(win, text='Work experience:',font='5')
        self.lbl12=Label(win, text='Field of experience:',font='5')

        self.lbl18=Label(win, text='Gender:',font='5')

        self.lbl20=Label(win, text='Citizenship:',font='5')
        self.lbl21=Label(win, text='Current CTC:',font='5')
        self.lbl22=Label(win, text='Expected CTC:',font='5')

        self.t1=Entry()
        self.t2=Entry()
        self.t3=Entry()
        self.t4=Entry()
        self.t5=Entry()
        self.t6=Entry()
        self.t7=Entry()
        self.t8=Entry()
        self.t9=Entry()
        self.t10=Entry()
        self.t11=Entry()
        self.t12=Entry()

        self.t18=Entry()

        self.t20=Entry()
        self.t21=Entry()
```

```

self.t22=Entry()

self.btn1 = Button(win, text='Submit',font='12')

self.lbl.place(x=450, y=10)
self.lbl1.place(x=100, y=50)
self.lbl2.place(x=100, y=100)
self.lbl3.place(x=100, y=150)
self.lbl4.place(x=100, y=200)
self.lbl5.place(x=100, y=250)
self.lbl6.place(x=100, y=300)
self.lbl7.place(x=100, y=350)
self.lbl8.place(x=100, y=400)
self.lbl9.place(x=600, y=50)
self.lbl10.place(x=600, y=100)
self.lbl11.place(x=600, y=150)
self.lbl12.place(x=600, y=200)

self.lbl18.place(x=600, y=250)

self.lbl20.place(x=600, y=300)
self.lbl21.place(x=600, y=350)
self.lbl22.place(x=600, y=400)

self.t1.place(x=450, y=50)
self.t2.place(x=450, y=100)
self.t3.place(x=450, y=150)
self.t4.place(x=450, y=200)
self.t5.place(x=450, y=250)
self.t6.place(x=450, y=300)
self.t7.place(x=450, y=350)
self.t8.place(x=450, y=400)
self.t9.place(x=800, y=50)
self.t10.place(x=800, y=100)
self.t11.place(x=800, y=150)
self.t12.place(x=800, y=200)
self.t18.place(x=800, y=250)
self.t20.place(x=800, y=300)
self.t21.place(x=800, y=350)
self.t22.place(x=800, y=400)

self.b1=Button(win, text='Submit', command=self.add_details)

self.b1.place(x=500, y=450)

def add_details(self):

    details=[]
    status='Selected'
    x=(self.t4.get())
    if x == 'BE':
        v=1
    elif x == 'ME':

```

```

        status='Not Selected'
        details.append(v)

x1=self.t5.get()
if x1 == 'CSE':
    v1=1
elif x1=='IT':
    v1=2
else:
    v1=3
if v1==3:
    status='Not Selected'
    details.append(v1)
    details.append(float(self.t6.get()))
    details.append(int(self.t7.get()))
x2=self.t8.get()
if x2=='yes':
    vv=1
else:
    vv=0
if vv==0:
    status='Not Selected'
    details.append(vv)

details.append(int(self.t11.get()))

if status=="Selected":

    msg="Status"+" Your recruitment status for CV selection is '+status+'."
else:

    msg="Status"+" Your recruitment status for CV selection is '+status

mail=self.t3.get()
with open('mailid.txt', 'w') as f1:
    f1.write(mail)
f1.close()

#mail =self.t3.get()
s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security
s.starttls()

# Authentication
s.login("recruitmentprocess@gmail.com", "process123")

# sending the mail
s.sendmail("recruitmentprocess@gmail.com", mail, str(msg))# Receiver mail id,
type your own id

# terminating the session
s.quit()

```

```

import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
from sklearn.metrics import classification_report, confusion_matrix#for visualizing tree
df = pd.read_excel('resume.xlsx')

features = df[["Graduation","Domain","CGPA","Year","University","Experience","E-CTC
lacs"]]
target = df["Result"]

print(features)
print(target)

# Split dataset into training set and test set
X_train,X_test, y_train, y_test = train_test_split(features, target, test_size=0.3,
random_state=1) # 70% training and 30% test
# Create Decision Tree classifier object
clf = DecisionTreeClassifier()
# Train Decision Tree Classifier
clf = clf.fit(X_train,y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)
#print(y_pred)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred)*100)

print("Classification report - \n", classification_report(y_test,y_pred))
#print(type(y_test))


window=Tk()
mywin=MyWindow(window)
window.title('AI Recruitment')
window.geometry("1000x500")
window.mainloop()


from mail import *

# Python program to create a simple GUI
# Simple Quiz using Tkinter

#import everything from tkinter
from tkinter import *

# and import messagebox as mb from tkinter
from tkinter import messagebox as mb

#import json to use json file for data
import json

```

```

#class to define the components of the GUI
class Quiz:
    # This is the first method which is called when a
    # new object of the class is initialized. This method
    # sets the question count to 0. and initialize all the
    # other methods to display the content and make all the
    # functionalities available
    def __init__(self):

        # set question number to 0
        self.q_no=0

        # assigns ques to the display_question function to update later.
        self.display_title()
        self.display_question()

        # opt_selected holds an integer value which is used for
        # selected option in a question.
        self.opt_selected=IntVar()

        # displaying radio button for the current question and used to
        # display options for the current question
        self.opts=self.radio_buttons()

        # display options for the current question
        self.display_options()

        # displays the button for next and exit.
        self.buttons()

        # no of questions
        self.data_size=len(question)

        # keep a counter of correct answers
        self.correct=0

    # This method is used to display the result
    # It counts the number of correct and wrong answers
    # and then display them at the end as a message Box
    def display_result(self):

        # calculates the wrong count
        wrong_count = self.data_size - self.correct
        correct = f"Correct: {self.correct}"
        wrong = f"Wrong: {wrong_count}"

        # calculates the percentage of correct answers
        score = int(self.correct / self.data_size * 100)
        result = f"Score: {score}%"
        runn(score)

        # Shows a message box to display the result
        #mb.showinfo("Result", f"{result}\n{correct}\n{wrong}")

```

```

# This method checks the Answer after we click on Next.
def check_ans(self, q_no):

    # checks for if the selected option is correct
    if self.opt_selected.get() == answer[q_no]:
        # if the option is correct it return true
        return True

# This method is used to check the answer of the
# current question by calling the check_ans and question no.
# if the question is correct it increases the count by 1
# and then increase the question number by 1. If it is last
# question then it calls display result to show the message box.
# otherwise shows next question.
def next_btn(self):

    # Check if the answer is correct
    if self.check_ans(self.q_no):

        # if the answer is correct it increments the correct by 1
        self.correct += 1

    # Moves to next Question by incrementing the q_no counter
    self.q_no += 1

    # checks if the q_no size is equal to the data size
    if self.q_no==self.data_size:

        # if it is correct then it displays the score
        self.display_result()

        # destroys the GUI
        gui.destroy()
    else:
        # shows the next question
        self.display_question()
        self.display_options()

# This method shows the two buttons on the screen.
# The first one is the next_button which moves to next question
# It has properties like what text it shows the functionality,
# size, color, and property of text displayed on button. Then it
# mentions where to place the button on the screen. The second
# button is the exit button which is used to close the GUI without
# completing the quiz.
def buttons(self):

    # The first button is the Next button to move to the
    # next Question
    next_button = Button(gui, text="Next",command=self.next_btn,
        width=10,bg="blue",fg="white",font=("ariel",16,"bold"))

    # palcing the button on the screen

```

```

next_button.place(x=350,y=380)

# This is the second button which is used to Quit the GUI
quit_button = Button(gui, text="Quit", command=gui.destroy,
width=5,bg="black", fg="white",font=("ariel",16," bold"))

# placing the Quit button on the screen
quit_button.place(x=700,y=50)


# This method deselect the radio button on the screen
# Then it is used to display the options available for the current
# question which we obtain through the question number and Updates
# each of the options for the current question of the radio button.
def display_options(self):
    val=0

    # deselecting the options
    self.opt_selected.set(0)

    # looping over the options to be displayed for the
    # text of the radio buttons.
    for option in options[self.q_no]:
        self.opts[val]['text']=option
        val+=1


# This method shows the current Question on the screen
def display_question(self):

    # setting the Question properties
    q_no = Label(gui, text=question[self.q_no], width=60,
font=( 'ariel' ,16, 'bold' ), anchor= 'w' )

    #placing the option on the screen
    q_no.place(x=70, y=100)


# This method is used to Display Title
def display_title(self):

    # The title to be shown
    title = Label(gui, text="AI Recruiter",
width=50, bg="green",fg="white", font=("ariel", 20, "bold"))

    # place of the title
    title.place(x=0, y=2)


# This method shows the radio buttons to select the Question
# on the screen at the specified position. It also returns a
# list of radio button which are later used to add the options to
# them.
def radio_buttons(self):

```

```

# initialize the list with an empty list of options
q_list = []

# position of the first option
y_pos = 150

# adding the options to the list
while len(q_list) < 4:

    # setting the radio button properties
    radio_btn = Radiobutton(gui, text=" ", variable=self.opt_selected,
                             value = len(q_list)+1, font = ("ariel", 14))

    # adding the button to the list
    q_list.append(radio_btn)

    # placing the button
    radio_btn.place(x = 100, y = y_pos)

    # incrementing the y-axis position by 40
    y_pos += 40

# return the radio buttons
return q_list

# Create a GUI Window
gui = Tk()

# set the size of the GUI Window
gui.geometry("800x450")

# set the title of the Window
gui.title("Online Test")

# get the data from the json file
with open('data.json') as f:
    data = json.load(f)

# set the question, options, and answer
question = (data['question'])
options = (data['options'])
answer = (data['answer'])

# create an object of the Quiz Class.
quiz = Quiz()

# Start the GUI
gui.mainloop()

```

Mail.py

```

import smtplib
from email.mime.text import MIMEText

```

```

from email.mime.multipart import MIMEMultipart

def send_email(recipient, score):
    # Email configuration
    sender_email = "nanibhaskar793@gmail.com"
    password = "fuwl rlys vzcq pako"
    smtp_server = 'smtp.gmail.com'
    smtp_port = 587

    try:
        # Create a connection to the SMTP server
        with smtplib.SMTP(smtp_server, smtp_port) as server:
            # Start TLS for security
            server.starttls()

            # Login to the email account
            server.login(sender_email, password)

            # Prepare the email content
            subject = "Interview Notification"
            body = "You are selected, kindly fill resume details!" if
score > 75 else "Sorry, you are not selected this time. Better luck next
time!"

            message = MIMEMultipart()
            message['From'] = sender_email
            message['To'] = recipient
            message['Subject'] = subject
            message.attach(MIMEText(body, 'plain'))

            # Send the email
            server.sendmail(sender_email, recipient,
message.as_string())

            print("Email sent successfully!")

    except Exception as e:
        print(f'An error occurred: {e}')

def runn(score):
    recipient = input("Please enter your email address: ")

```

```

# Add input validation if necessary
# if not is_valid_email(recipient):
#     print("Invalid email address.")
#     return

send_email(recipient, score)

# Example usage
score = float(input("Enter the score: "))
runn(score)

import smtplib
def runn(score):
    # creates SMTP session
    s = smtplib.SMTP('smtp.gmail.com', 587)

    # start TLS for security
    s.starttls()

    # Authentication
    s.login("recruitmentprocess@gmail.com", "process123")

    mail=input("Please enter your mail id:")

    if score >75:

        # message to be sent
        message = "You are selected for Face to Face interview!"
    else:
        message = "Sorry!,You are not selected for this time.Better
luck next time!"

    # sending the mail
    s.sendmail("recruitmentprocess@gmail.com", mail, message)#
Receiver mail id, type your own id

    # terminating the session
    s.quit()

```

```

import smtplib
def runni(score1):
    # creates SMTP session
    s = smtplib.SMTP('smtp.gmail.com', 587)

    # start TLS for security
    s.starttls()

    # Authentication
    s.login("recruitmentprocess@gmail.com", "process123")

    mail=input("Please enter your mail id:")

    if score >75:

        # message to be sent
        message = "Congratulations!! you are selected and soon
you will receive further details. "
        else:
            message = "Sorry!,You are not selected for this
time.Better luck next time!"

        # sending the mail
        s.sendmail("recruitmentprocess@gmail.com", mail,
message)# Receiver mail id, type your own id

    # terminating the session
    s.quit()

```

Part.py

```

import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split
function
from sklearn import metrics #Import scikit-learn metrics module for accuracy
calculation
from sklearn.metrics import classification_report, confusion_matrix#for visualizing
tree
df = pd.read_excel('resume.xlsx')

```

```

features = df[["Graduation","Domain","CGPA","Year","University","Experience","E-
CTC lacs"]]
target = df["Result"]

print(features)
print(target)

# Split dataset into training set and test set
X_train,X_test, y_train, y_test = train_test_split(features, target, test_size=0.3,
random_state=1) # 70% training and 30% test
# Create Decision Tree classifier object
clf = DecisionTreeClassifier()
# Train Decision Tree Classifier
clf = clf.fit(X_train,y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)
#print(y_pred)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred)*100)

print("Classification report - \n", classification_report(y_test,y_pred))
#print(type(y_test))

```

Data

```

{
  "question": [
    "Q1. Who developed Python Programming Language?",
    "Q2. Which of the following is not a valid C variable name?",
    "Q3. Which component is used to compile, debug and execute the java
programs?",
    "Q4. Which of the following is the correct extension of the Python file?",
    "Q5. Which is valid C expression?"
  ],
  "answer": [3,
    4,
    3,
    3,
    2],
  "options": [

    ["Wick van Rossum",

```

```

    "Rasmus Lerdorf",
    "Guido van Rossum",
    "Niene Stom"
  ],
  ["int number;",
    "float rate;",
    "int variable_count;",
    "int $main;"
  ],
  ["JRE",
    "JIT",
    "JDK",
    "JVM"
  ],
  [".python",
    ".pl",
    ".py",
    ".p"
  ],
  ["int my_num = 100,000;",
    "int my_num = 100000;",
    "int my num = 1000;",
    "int $my_num = 10000;"
  ]
]
}

```

Data 1

```

"question": [
  "Q1. How much you rate about yourself?",
  "Q2. In which shift are u comfortable to work?",
  "Q3. Which location dou you prefer?",
  "Q4. Will you work on weekends?",
  "Q5. Within how many months you will be able to join?"
],
"answer": [1,
  4,
  4,
  1,

```

```

1],
"options": [

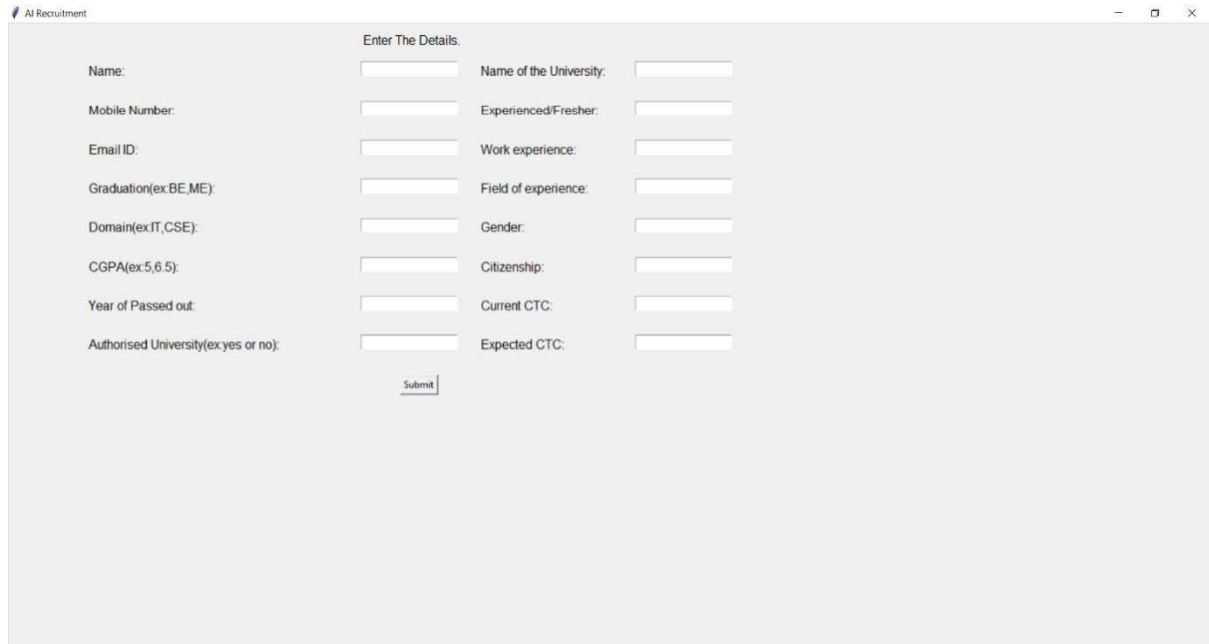
  [" 10",
    "9",
    "8",
    "5"
  ],
  [" 5:00 - 3:30 ",
    "9:00 - 6:00",
    "2:00 - 11:00",
    "any shift"
  ],
  ["Mumbai",
    "Pune",
    "Chennai",
    "Banglore"
  ],
  ["yes",
    "no",
    "based on work",
    "extra pay"

  ],
  ["1",
    "2",
    "3",
    "4"

  ]
]
}

```

B. SNAPSHOTS

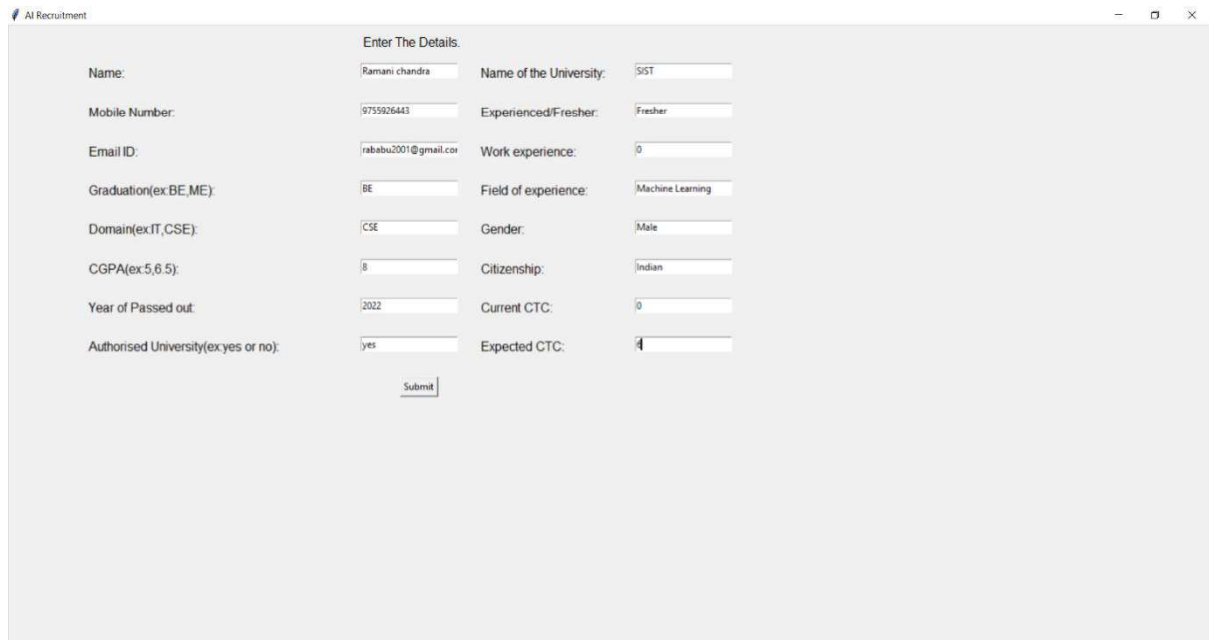


AI Recruitment

Enter The Details.

Name:		Name of the University:	
Mobile Number:		Experienced/Fresher:	
Email ID:		Work experience:	
Graduation(ex.BE,ME):		Field of experience:	
Domain(ex.IT,CSE):		Gender:	
CGPA(ex.5,6.5):		Citizenship:	
Year of Passed out:		Current CTC:	
Authorised University(ex.yes or no):		Expected CTC:	

Fig B.1 Enter the details



AI Recruitment

Enter The Details.

Name:	Ramani chandra	Name of the University:	SIST
Mobile Number:	9755926443	Experienced/Fresher:	Fresher
Email ID:	rababu2001@gmail.com	Work experience:	0
Graduation(ex.BE,ME):	BE	Field of experience:	Machine Learning
Domain(ex.IT,CSE):	CSE	Gender:	Male
CGPA(ex.5,6.5):	8	Citizenship:	Indian
Year of Passed out:	2022	Current CTC:	0
Authorised University(ex.yes or no):	yes	Expected CTC:	4

Fig B.2 Fill User Details

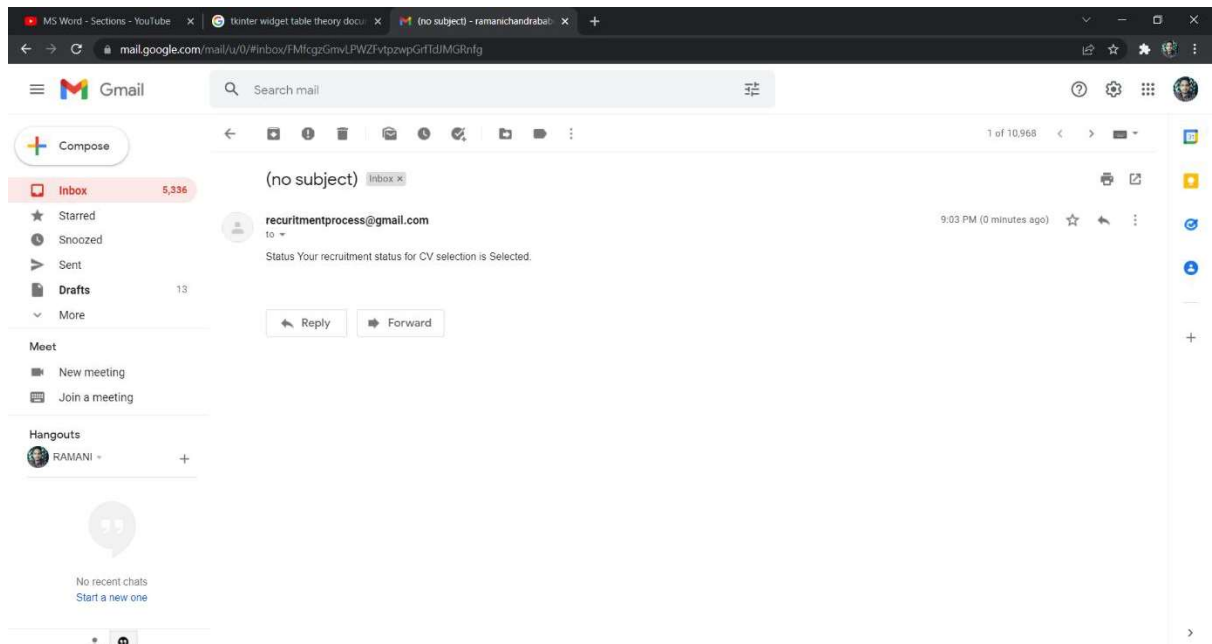


Fig B.3 Recruitment status mail

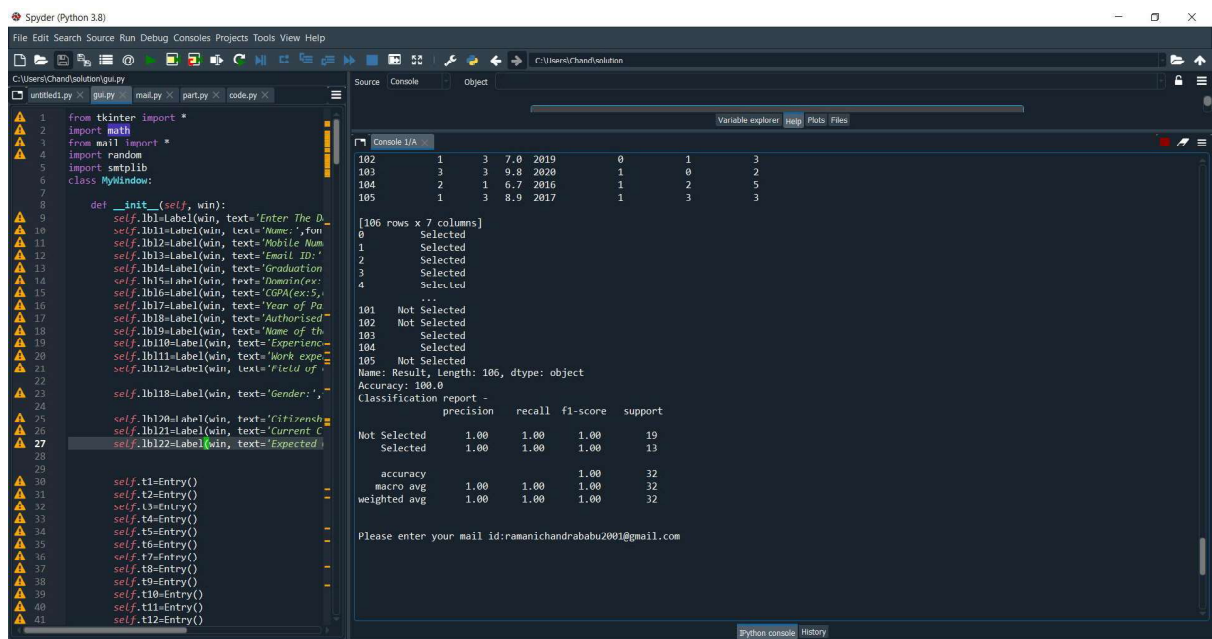


Fig B.4 Accuracy Report

C. RESEARCH PAPER

JOB SEARCH ASSISTANT FOR THE UNEMPLOYED

T Dhanush

*Department of Computer Science &
Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
dhanushreddy192003@gmail.com*

Devarapalli Bhaskar Satyanarayana

*Department of Computer Science &
Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
dbaskarnani@gmail.com*

Dr. Veena K

*Assistant Professor, Department of
Computer Science & Engineering
Sathyabama Institute of Science and
Technology
Chennai, India
veena.cse@sathyabama.ac.in*

Abstract—Any enterprise, business or corporation wishes great employees to survive and acquire its dreams. In this age of competition. All of them are at the start of the fourth commercial revolution. Everyone wishes to be competitive in this virtual global Looking for talented, dynamic and lively personnel. Organizations can lease with a powerful recruitment strategy the right individual to guide the digital global and enhance matters. Yes, recruiting is a warfare an essential element for any employer is to hire certified employees who can do their paintings greater successfully and successfully. Since focused on military recruitment is a center feature of government, evaluation of facts is without a doubt required for choice- making. Manufacturing manner, manufacturing process. Data analysis is known as “Artificial Intelligence” and performs a critical position in making hiring choices. Artificial intelligence, in the most fundamental terms, is the creation of clever machines by way of humans. AI can act and act like a human its final intention is to enable computers to perform obligations typically done through humans. AI is advancing at an extremely good pace and hard work. The predominant cause of this newsletter is to explore how synthetic intelligence is getting used as a recruitment approach.

Keywords—*Decision Tree, Supervised Machine Learning Algorithm, AI.*

Introduction

Artificial intelligence is technologically superior and superior to create robots. A PC or product is designed for human imitation and ways to find conclusions. Artificial intelligence is one of the most famous enterprise regions in finance, era and technology these days. With recent technological advances, the age of gift giving may be a completely glad one. Artificial

Intelligence and Machine Learning. Central offerings that had been once executed bodily are actually finished through software and automated structures [1]. John McCarthy, some other well-known American laptop scientist, artificially describes artificial intelligence (AI) and intelligent vehicles. Machines paintings tough and do their task beneath tough conditions. Excellent use of perceptual intelligence, that's manipulated, solves issues, masters common sense and recognizes complicated ideas, learns fast from revel in, and develops plant intelligence.

Literature Survey

Literature evaluate is an crucial step within the software program development procedure. Before the tool is evolved, it's far important to determine the time elements, budget and strengths of the business enterprise. With a portion of these circumstances met, the accompanying step is to figure out which working gadget and language can be utilized to expand the instrument. At the point when software engineers begin growing a device, they may want more help. This help can be acquired from senior programs, books or web sites. Before designing the gadget, the above

factors are stored in mind at the same time as designing the gadget. The most crucial part of the work of the challenge development branch is the precise evaluation of all of the requirements which can be important to develop the task. Literature review is the most crucial region of software improvement for any task. Before growing the equipment and the related plan, it's miles necessary to pick out and examine the factors of time, aid necessities, manpower, economic and organizational strengths. After some of these questions have been replied and the research executed, the following step is to determine the specification of the software program inside the laptop assignment, what type of running device may be required for the purpose, and what software program might be required for the mission. ., paintings Making tracks like equipment and associated activities.

Necessities approval methods: An exact review

Prerequisites approval guarantees that the necessities recorded inside the Product Necessities Determination (SRS) are whole, steady, and meet customer requirements. This affords validity to consumer necessities by using removing ambiguity and inconsistency from DRS. Many components of verification necessities had been discussed within the literature. This paper provides an outline of the verification necessities for the prevailing business technologies, including discovery necessities, prototype requirements, checking out requirements, and attitude-primarily based verification necessities. This article also highlights the advantages and downsides of these strategies. When checking out is needed, unique consideration is given to TCD research.

A survey on programming necessities approval and consistency the board

The want for software program improvement have to be emphasized to keep away from errors along with inconsistency, incompleteness, and inaccuracy. Based on this common sense, the validation technique ought to don't forget the necessities for consistency, completeness and correctness ("3Cs") to reap

the software best specification. This paper presents a evaluate of present requirements validation and versatility control literature to pick out current understanding gaps in the software program specification process. This article starts off evolved with an outline of the definitions of 3C in line with the information of 3C. A more particular assessment of relevant work on identified sustainability management practices is offered: a diagnostic and analytical method. This is showed by showing a warmness map of relevant research inside the ratios, ratios, types and massive categories used in coherent management. Since semi-formal specifications have been located to be a commonplace representation of necessities, the types of fashions used as semi-formal specs to symbolize necessities are mentioned. Overall, this paper identifies numerous gaps within the necessities consistency overview and management process to avoid reinventing the wheel in a numerous and tremendous expertise of requirements itself.

Improving software quality using machine learning

Software is an organization that is continuously evolving and present process steady adjustments to improve its capability and preserve its effectiveness. In software program improvement, problems rise up regardless of a cautiously planned, nicely-documented and properly-managed process. These defects by some means affect the great of the software program, leading to mistakes. Therefore, in contemporary aggressive surroundings, it's miles essential to control and decrease those defects in software development. Software predictive models are normally used to show software program models which can alternate. This article highlights important analyzes within the getting to know and motivation area of the connection among metrics representing object orientation and possibility of change. This often leads us to an exact check to discover all of the possible values in a given set. We have approaches: (1) degree the parameters that affect the high-quality, functionality and performance of this system. (2) This software

program uses device learning talents to make predictions. The reason of the research paintings is to examine all the education methods to carry out the parameter with the statistical technique and the method of transmission. The points given are the idea of the advertisement patterns.

Programming prerequisite investigation: Research difficulties and specialized approaches

Prerequisites examination is one of the most extreme essential obligations in programming program designing ventures. The particular and the board of consumer necessities has an expansion of impacts on software improvement and continues to strengthen within the educational and commercial fields. Models which include CMMI describe the development and control of requirements, and outline precise desires and methods for them. This article lists the main needs control troubles and issues related to the benchmarking hobby called CMMI.

Requirement validation model for virtual distributed system

Engineering requirements plays an crucial role in all areas of pc engineering and development. Various technologies or fashions were proposed for the system itself. All of those models use one-of-a-kind techniques to resolve engineering issues. Several well-known sports are common to all approaches, specifically: necessities elicitation, requirements evaluation, requirements specification, requirements validation and necessities management. During this time, many research were proposed in numerous checking out methods. All these strategies proposed through researchers particularly attention on centralized schemes. This work researches requirements validation framework for distributed digital enterprises. The essential aim of this framework is to enhance the exceptional of virtual disbursed systems with the aid of providing a simple and systematic manner to validate the necessities of such systems. Distributed digital environments are consumer structures placed in specific areas of

the world and using exceptional networks. They're prepared without any end consumer requirements. Our systematic method gives a scientific way for the requirements engineers to systematically examine the necessities that have arisen from the end customers, in certain factors together with completeness, correctness, and so forth. Three approaches were used to broaden the proposed framework; From prototype, assessment and trying out. The a success implementation of the proposed requirement in a disbursed digital surroundings can have a wonderful effect at the development of excellent software in the future.

Existing System

Most of the preceding literature mentioned the loss of jobs because of automation or artificial intelligence that's an clever system that will become a chance to people even as it turns into a hazard to human activities. However, this take a look at is essential from the factor of view of human resources, due to the fact not all jobs can trade because of the fourth wave of the industry. According to a record by means of the World Economic Forum (WEF), notwithstanding the danger of layoffs, jobs requiring personal competencies continue to be in call for. Furthermore, Ireland has followed the idea of further studies on this region, which isn't the least favorable for the united states amongst its European opposite numbers. In this text, the researcher tried to apprehend the relationship between standard recruitment managers and the effectiveness of AI adoption in finding top expertise, which is vital for lengthy-term organizational boom. Due to extreme gaps in get admission to to facts and related assets. Finding enough information and mixing it with the examine goals became a venture.

Disadvantages of Existing System

- Low maintenance
- Greatest enrollment

Using the Template

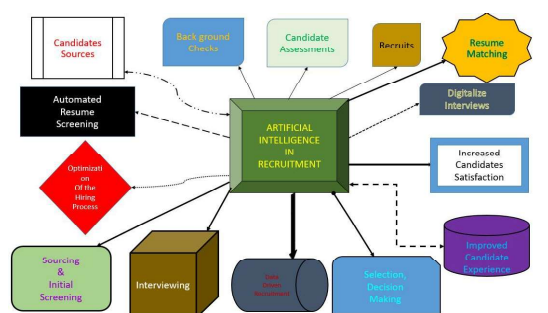
Artificial intelligence — this generation, that could act intelligently just like the human brain, in various conditions. Compared to traditional

strategies of employee's choice, automation of employee's choice methods increases attention and importance. The election of managers is a relevant interest inside the paintings of all businesses. Now the personnel choice enterprise is growing because of the creation of complex techniques of employee's selection, that is, personnel recruitment with the assist of synthetic intelligence. Many industries can be running diligently to make changes in the employee's selection procedure. Synthetic intelligence era has a extraordinary impact in the procedure of personnel selection, as it allows the recruiter to optimize all the informative biographical statistics of candidates, optimize profiles, as well as to recognize the abilities and abilities required by way of the enterprise. In current times, recruiters think that synthetic intelligence technology compete with them inside the recruitment system. But this is software program, created by guy. To entire the function of AI, it's far a aggregate of human beings and AI, which leads to the conservation of records, economy of assets, and time of companies, extra accuracy and accessibility to the whole procedure of employees choice.

Advantages of Proposed System

- except for time
- Economy of fee
- Sheet steel

Block Diagram



Proposed Calculation

Choice Trees for Characterization: An AI Calculation

Introduction

Decision trees are a type of system gaining

knowledge of in rate (i.E. You provide an explanation for what the inputs are to the schooling facts and what the corresponding outputs are) in which the facts is continuously divided in step with a sure module. Conclusion A tree can be described by using two elements referred to as nodes and leaves. Returns the result or rejects the result. Leaf nodes are places wherein statistics is separated.



An instance of a spanning tree can be illustrated the usage of a binary tree. Let's say you need to expect whether or not a person is healthy the use of records including age, weight loss program, and bodily pastime. "What age?", "Does he play?", "Does he practice?" Do you eat quite a few pizza? And leaves consequences as proper or ineligible. In this example, the hassle is a binary division (sure, now not a gender hassle).

There are two kinds of choice bushes;

Classification Tree (Yes / No Categories)

What we have seen above is an example of a classification tree that effects in variables together with "in shape" or "fail". The final results here is special.

Regression Tree (Data Type Continuous)

The output or output variable here is non-stop, as an instance. A wide variety like 123

Work

Now that we recognize what a choice tree is, let's have a look at how it works internally. There are many algorithms for building selection bushes, however the first-class one is referred to as the ID3 set of rules. ID3 stands for "repeat binary code three".

Before we discuss the ID3 algorithm, permit's give a few definitions.

Module Description

Information Obtaining Module

This implies that today, as a scout, you really

want to explore virtual entertainment to choose who the candidates are. In addition, AI-powered computerized software to robotically suit you along with your applicants primarily based on standards. Automatically file records from surveys and analyzes with social media to resumes, so that your facts approximately making plans tells the truth about the candidate revel in. Artificial intelligence algorithms can capture and analyze hundreds of thousands of profiles. It is apparent that that is something that guy can't do. At the coronary heart of AI is interplay with humans and applicants as the system maintains and the paintings of screening and verification is finished day by day. This will make sure all and sundry's participation.

Continue Screening Module

The selection system can also be improved. You may not waste any time as you have got already identified treasured candidates based totally in your statistics. Know that your skills is needed for your industry and begin doing it fast. Qualified candidates will perform better and the right candidate will emerge faster. Facial reputation enables you to inform the right candidate, and misidentification remains a problem of the past.

Applicant Matching Module

Candidate recruitment is pushed by using a customized candidate experience, and you may tailor the candidate revel in to your desires with the assist of effective AI-assisted algorithms. Finding a person with five to 10 years of experience with amazing capabilities, connections and understanding will be an smooth assignment, as the variety and variety of facts will permit accurate analysis and information extraction. Use candidate customization to apprehend precisely what your candidates love. You may also understand when to intervene for an extended life. For example, in case you see your candidate converting roles and obligations each 3 years, you can are expecting and offer something new before it reaches that tipping point.

Really taking a look at Reference

Module

This makes it less complicated to test your notes. Your AI set of rules will use all forms of records. It can connect to humans and create networks with starfish connections. Touch their notes, if possible, or take a look at out what they've written or stated on social media and deliver your opinion.

Lawful Angle Module

After a lot of these steps are completed, the prison aspect will become automatic. The felony crew can be assured that they will have access to all information amassed automatically during the recruitment technique. Contracts will take account of all statistics pushed through the recruitment method and might be finished with human enter as required by way of law.

Results And Discussion

Artificial intelligence is an invention that can act very much like the human mind in diverse conditions. There is more interest and importance in mechanizing the accounting structure compared to standard recruitment techniques. Enumeration is the driving pressure behind all institutions. Now, the memory enterprise is growing human focus through the creation of an intelligent catalog choice device. Furthermore, numerous efforts have been made to promote development inside the registration technique.

Conclusion

Innovations in synthetic intelligence have a notable effect on recruiting, due to the fact that they permit the choice of dealers to adapt truly casual and futuristic resumes, to identify and adapt the potential of knowledge from the business to create a coherent profile. In brand new generation, statisticians consider that innovations in laptop intelligence will compete with additive manufacturing. Either way, this is an inventive software designed to preserve matters jogging smoothly. To complete the work of computer intelligence, the aggregate of human and artificial intelligence provides data guide, fee-saving and collaborative time for a extra correct and comprehensive admissions

technique.

Working Principle

Enter The Details

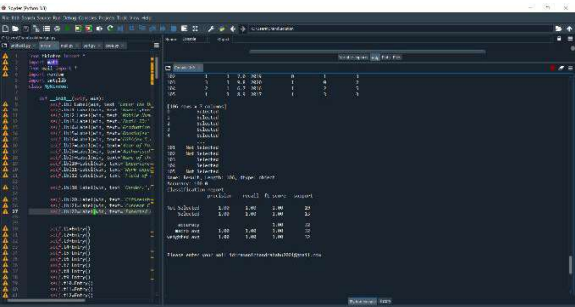
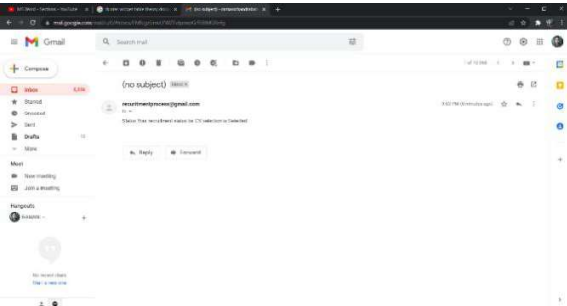
Name	Name of the University
Mobile Number	Experience (S/E/I)
Email ID	Field of experience
Graduation (B.E./M.E.)	Gender
Dissertation (IT/CS/E)	Citizenship
CDP Paper (S.E./S)	Current CTC
Year of Passed out	Expected CTC
Authorized Universities (yes or no)	

Submit

Enter The Details

Name	Name of the University
Mobile Number	Experience (S/E/I)
Email ID	Field of experience
Graduation (B.E./M.E.)	Gender
Dissertation (IT/CS/E)	Citizenship
CDP Paper (S.E./S)	Current CTC
Year of Passed out	Expected CTC
Authorized Universities (yes or no)	

Submit



REFERENCES

[1] H.A. Bilal, M. Ilyas, Q. Tariq and M. Hummayun, "Requirements validation techniques: An empirical study", International Journal of Computer Applications, vol. 148, no. 14, 2016.

[2] M. Kamalrudin and S. Sidek, "A review on software requirements validation and consistency management", International Journal of

Software Engineering and Its Applications, vol. 9, no. 10, pp. 39-58, 2015.

[3] K. Chandra, G. Kapoor, R. Kohli and A. Gupta, "Improving software quality using machine learning" in 2016 ICICCS-INBUSH, IEEE, pp. 115-118, February 2016.

[4] S.T. Demirel and R. Das, "Software requirement analysis: Research challenges and technical approaches", 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1-6, 2018, March.

[5] T. Kiran, S. Farhan, H. Tauseef and M. A. Fahiem, "Requirement validation model for virtual distributed system", Int. J. Multidiscip. Sci. Eng., vol. 4, no. 8, pp. 34-42, 2013.

[6] Dharendra Pandey, Waris Khan and Vandana Pandey, ROLE OF REQUIREMENT VALIDATION IN REQUIREMENT DEVELOPMENT, 2012.

[7] U.A. Raja, "Empirical studies of requirements validation techniques", 2009 2nd International Conference on Computer Control and Communication, pp. 1-9, 2009, February.

[8] .X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning", Synthesis lectures on artificial intelligence and machine learning, vol. 3, no. 1, pp. 1-130, 20

D. CERTIFICATE

