VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering

Project Report on

IdeaFlow: Flowchart Creation using Gen AI

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2023-24

Submitted by

Madhura Mhatre (D17 - C , Roll no - 32)
Ananya Pandey (D17 - C , Roll no - 42)
Harsh Rane (D17 - C , Roll no - 46)

Project Mentor
Asst. Prof. Priyanka Shah

(2023-24)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering

Certificate

This is to certify that ***Madhura Mhatre, Ananya Pandey, Harsh Rane*** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "IdeaFlow: Flowchart Creation using Gen AI" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor ***Asst. Prof. Priyanka Shah*** in the year 2023-24 .

This thesis/dissertation/project report entitled ***IdeaFlow: Flowchart Creation using Gen AI*** by ***Madhura Mhatre, Ananya Pandey, Harsh Rane*** is approved for the degree of ***Bachelor of Engineering (B.E.) Degree in Computer Engineering***.

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide:

Project Report Approval
For
B. E (Computer Engineering)

This thesis/dissertation/project report entitled ***IdeaFlow: Flowchart Creation using Gen AI*** by ***Madhura Mhatre, Ananya Pandey, Harsh Rane*** is approved for the degree of ***Bachelor of Engineering (B.E.) Degree in Computer Engineering***.

Internal Examiner

External Examiner

Head of the Department

Principal

Date:

Place:

Declaration

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

(Signature)

Madhura Mhatre (32)

(Signature)

Ananya Pandey (42)

(Signature)

Harsh Rane (46)

Date:

VIVEKANAND EDUCATION SOCIETY'S
INSTITUTE OF TECHNOLOGY

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Mrs. Priyanka Shah** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department

COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Chapter	Title	Page no.
	Abstract	11
1	**Introduction**	12
1.1	Introduction	12
1.2	Motivation	13
1.3	Problem Definition	13
1.4	Existing Systems	14
1.5	Lacuna of Existing Systems	15
1.6	Relevance of the Project	15
2	**Literature Survey**	16
2.1	Brief Overview of Literature Survey	17
2.2	Research Papers Referred	18
2.3	Comparison with the existing system	20
3	**Requirement Gathering for the Proposed System**	22
3.1	Introduction to requirement gathering	22
3.2	Functional Requirements	22
3.3	Non-Functional Requirements	23
3.4	Hardware, Software , Technology and tools utilized	24
3.5	Constraints	24

4	**Proposed Design**	25
4.1	Block diagram of the system	25
4.2	Modular Diagram	25
4.3	Detailed Design	26
5	**Implementation of the Proposed System**	27
5.1	Methodology employed for development	27
5.2	Algorithms and flowcharts for the respective modules developed	28
6	**Chapter 6: Testing of the Proposed System**	31
6.1	Introduction to testing	31
6.2	Types of tests Considered	31
6.3	Various test case scenarios considered	31
6.4	Inference drawn from the test cases	31
7	**Results and Discussion**	32
7.1	Screenshots of User Interface (UI) for the respective module	32
7.2	Performance Evaluation measures	34
7.3	Comparison of results with existing systems	35
8	**Conclusion**	36
8.1	Limitations	36
8.2	Conclusion	37

8.3	Future Scope	
	References	39
	Appendix	40
	Plagiarism Report	41
	Paper details	42
	Review Sheets	49

List of Figures

Fig No.	Figure Name	Page No.
1	Block Diagram of IdeaFlow	26
2	Modular Diagram of IdeaFlow	26
3	Interaction of model with given user prompt	27
4	Methodology Diagram	29
5	Flowchart to find area of circle	33
6	Flowchart to find average of three numbers	33
7	Flowchart to find sum of n natural numbers	34
8	Save options	24

List of Tables

Sr no.	Table Name	Pg no.
1.	Limitations of Existing Systems	15
2	Comparison of existing system with IdeaFlow	21
3	Comparison of results of existing system with IdeaFlow	35

ABSTRACT

This project aims to revolutionize the creation of flowcharts across various industries by leveraging Generative AI technology. Traditional methods of flowchart creation are often time-consuming, labor-intensive, and prone to human errors. To address these challenges, the project proposes the development of an innovative Generative AI system that automates the process of flowchart creation. By harnessing the power of Generative AI, professionals can save time, reduce errors, and ensure consistency in their diagrams.

The proposed system will handle challenges such as ambiguous descriptions and complex decision-making processes, enabling it to generate accurate and visually coherent flowcharts. With potential applications ranging from software development to project management, the system offers diverse benefits. It can streamline tasks such as debugging and documentation for software developers, aid in process optimization and design for engineers, and enhance project planning and execution for project managers.

Motivated by the need to revolutionize workflow documentation, the project seeks to overcome the limitations of existing tools and methods. By automating flowchart creation through Generative AI, the system aims to provide professionals with an efficient, reliable, and user-friendly solution for visualizing complex processes, ultimately improving productivity, standardization, and innovation across industries.

Chapter 1: Introduction

1.1. Introduction to the project

Flowcharts are essential tools used to visualize complex processes and workflows in various domains, including software development, engineering, project management, and business analysis. They provide a clear and concise representation of the steps and decisions involved in a process, facilitating communication, analysis, and optimization. However, the creation of flowcharts is often a time-consuming and labor-intensive task, requiring domain expertise and meticulous attention to detail.

By harnessing the power of Generative AI, a subfield of AI focused on generating content, this project aims to develop an innovative solution: a Generative AI system for automating flowchart creation.The primary objective of this project is to develop an AI model that can take textual descriptions of processes or workflows as input and generate accurate and visually coherent flowcharts as output. By automating the flowchart creation process, professionals can save time, reduce human errors, and ensure consistency across various diagrams.

Throughout the project, various challenges will be addressed, such as handling ambiguous descriptions, capturing complex decision-making processes, and ensuring the generated flowcharts are accurate and faithful to the input descriptions. Rigorous data preprocessing, model training, and evaluation methodologies will be employed to achieve reliable and effective results.

The potential applications of this Generative AI system for Flowchart Creation are vast and diverse. Professionals in software development can automate the generation of program flowcharts from written code descriptions, streamlining the debugging and documentation processes. Engineers can benefit from automated workflow visualization, aiding in process optimization and design. Project managers can create visual project timelines and resource allocation flowcharts with ease, enhancing project planning and execution.

In conclusion, this project represents a pioneering effort to harness the capabilities of Generative AI in simplifying and accelerating the creation of flowcharts. The outcomes of this research hold the promise of transforming how professionals across various industries visualize and communicate complex processes.

1.2 Motivation

The motivation behind developing Generative AI for flowchart creation is driven by the potential to revolutionize how processes and workflows are documented. This technology promises to significantly improve productivity by automating the creation of visual representations of complex processes, saving time and reducing the likelihood of errors in the documentation process. It democratizes flowchart creation by making it accessible to individuals who lack expertise or specialized software tools, and it has the potential to scale seamlessly, accommodating both simple and intricate processes. Standardization is another key benefit, as AI can enforce consistent formats and notations, ensuring that documentation adheres to established best practices. Furthermore, by automating routine flowchart creation tasks, professionals can redirect their efforts toward more creative and strategic aspects of their work, fostering innovation and efficiency in various industries and sectors.

1.3 Problem Definition

The problem addressed by Generative AI for flowchart creation revolves around the labor-intensive and error-prone nature of manually generating flowcharts. Creating flowcharts to represent complex processes or workflows can be a time-consuming and often challenging task for professionals across various domains, from software development to project management and industrial engineering. The primary issues include:

1. Time-Consuming: Manually designing flowcharts requires a substantial amount of time and effort. Professionals must painstakingly draw, connect, and label shapes to represent the process accurately.
2. Error-Prone: Human errors in flowchart creation can lead to misinterpretation, misunderstanding, and incorrect implementation of processes, potentially causing operational inefficiencies and project delays.
3. Expertise and Accessibility: Flowchart creation often demands expertise in diagramming tools and process notations, which can be a barrier for those who lack such skills or specialized software.
4. Inconsistency: The lack of standardized formats and notations can result in inconsistent documentation, making it challenging for team members to collaborate effectively and understand processes uniformly.
5. Scalability: Generating flowcharts for large and intricate processes can be particularly challenging and time-consuming, limiting scalability in process documentation.

Generative AI for flowchart creation aims to address these issues by automating the generation of flowcharts, offering the potential for significant time savings, enhanced accuracy, improved accessibility, standardization, and scalability. This technology seeks to empower professionals across industries to efficiently and consistently create clear, visual representations of their processes, ultimately fostering better communication, decision-making, and productivity.

1.4 Existing Systems

- Draw.io: A popular, free, and open-source diagramming tool. It's versatile and widely used for creating flowcharts, mind maps, and other diagrams.
- Lucidchart: A commercial web-based diagramming solution that offers collaboration features, extensive template libraries, and integrations with other productivity suites.
- Whimsical: A collaborative visual workspace emphasizing ease of use and a modern interface. It includes tools specifically designed for flowcharts, mind maps, and wireframes.

- Overleaf: A primarily LaTeX-based online editor, which also allows for the creation of basic flowcharts. Overleaf is known for its capabilities with scientific and mathematical documents.

1.5 Lacuna of the existing system

The traditional methods of manually making a flowchart can be very time consuming and effort demanding. Applications such as the draw.io fall under this category. Other applications include-

Table 1. Limitations of existing systems

Ref	Tools Name	Description	Advantages	Limitations
1	draw.io	Provides shapes for diagram creation. Users connect shapes to form diagrams.	Allows users to create their own templates. Can connect draw.io with Google Drive	Flow charts need to be created manually. Lack of proper tutorials and technical support. Does not support complex diagrams.
2	Whimsical	Allows users to create visual documents. Provides AI output	Unified workspace for thinking and collaboration. Fast, intuitive, and fun to use. Rich library of visual elements and templates.	Difficulty collaborating on large projects. Performance issues. Cluttered interface. Comments option not available.
3	Taskade	Enables project organization, visualization, and mapping	Real-time collaboration. Video chat. File sharing. Comments and mentions. Video chat. File sharing. Comments and mentions.	No integrations with other apps.Small file upload limit. No support for complex projects
4	Overleaf	Cloud-based LaTeX editor for scientific documents	Collaborate on flowcharts with others.	Requires knowledge of LaTeX.

			Export flowcharts to various formats. Integrate with tools like Jira and GitHub.	Can be slow. Not as user-friendly as some tools.
5	Lucidchart	Web-based diagramming tool for various diagrams	Wide range of shapes and connectors. Export diagrams to various formats. Collaborate on diagrams. Integrate with other tools.	Can be slow. Not as user-friendly Free plan is limited.

1.6 Relevance of the project

Generative AI for flowchart creation is highly relevant in today's world. It can help organizations and professionals address the pressing challenges of increasing complexity, remote work, accessibility, and the need for efficiency and standardization.

Benefits of generative AI for flowchart creation:

1. Clarity and accuracy: AI-generated flowcharts are clear and accurate, reducing the risk of misinterpretation and errors.
2. Time savings: Automating flowchart creation saves professionals and organizations a substantial amount of time.
3. Improved communication: AI-generated flowcharts facilitate better communication and understanding among team members across different locations and time zones.
4. Accessibility: Generative AI makes flowchart creation more accessible to a wider range of individuals, including those who may not be proficient in diagramming tools.
5. Efficiency and innovation: By automating routine flowchart creation, professionals can redirect their efforts toward more creative and innovative aspects of their work.

6. Sustainability: Streamlining the documentation process with AI can reduce paper consumption and environmental impact.

Overall, generative AI for flowchart creation is a powerful tool that can help organizations and professionals improve their productivity, communication, and decision-making.

2 Literature Survey

2.1 Brief overview of Literature Survey

The literature survey exposes a distinct gap in existing research—the application of generative AI for automating flowchart creation remains uncharted territory. Recognizing this gap, the survey adopts a two-pronged approach. Firstly, it examines the broader field of generative AI research. Studies in this area demonstrate the remarkable ability of generative AI models to comprehend instructions and synthesize structured outputs. This capability hints at the potential for such models to generate flowcharts, which are inherently structured visual representations.

Secondly, the survey delves into established flowchart creation tools, techniques for mind mapping, and graph representation methods. These studies highlight the power of natural language processing (NLP) in breaking down and understanding user requirements. Additionally, rule-based systems emerge as critical components for refining and ensuring the correctness of generated flowcharts. This analysis suggests that a successful approach to automating flowchart creation could leverage the initial structure-generating capabilities of generative AI, coupled with rule-based systems to enforce proper formatting, labeling, and flow logic.

Building such a system would hinge on several key factors. The models would require rigorous training on massive datasets containing text descriptions paired with their corresponding flowchart representations. Furthermore, the success of this innovative approach relies heavily on the design of an intuitive user interface. This interface would be essential for providing textual instructions, as well as potentially offering feedback and refinements to the system's generated flowcharts.

2.2 Research Papers referred

Stefan Feuerriegel et .al [1] emphasizes capacity to create a transformative impact on work and communication. It discusses applications, economic implications, and highlights opportunities and challenges for the Business & Information Systems Engineering (BISE) community. This research helps us understand more about generative AI. Generative AI helps amplify human works in fields like creativity, information, etc. The limitations that we encountered were incorrect outputs and a risk of copyright violations.

Wang, Hongwei, et al. [2], in this study unifies two schools of graph representation learning methodologies, i.e., generative methods and discriminative methods, via adversarial training in a minimax game. They conducted empirical study which allowed them to justify what they proposed with their theory. Performance of prediction was done and also the project could be used for recommendation.

Abdeen, Mohammad, et al. [3] presented M2Gen, a software tool that automatically generates mind maps from input text. This study was very aligning with our problem statement since mind maps are fairly corresponding with flowcharts. They made use of five modules in order to generate the mind maps and were successful with the approach they adapted. The software tool is only restricted to generating mind maps.

Mirza, Mehdi, and Simon Osindero. [4], in this work they introduced the conditional version of generative adversarial nets, which can be constructed by simply feeding the data, we wish to condition on both the generator and discriminator. The generator and discriminator were conditioned which gives an added scope for user customization. Although the work was very well organized, the results shown were preliminary and
lacking in diversity.

Simonovsky, Martin, and Nikos Komodakis. [5] the possibility to predict detailed attributes of atoms and bonds in addition to the base structure, compared to SMILES decoder has been discussed. Advantages for flowchart generation include the ability to generate fully-connected graphs of a predefined maximum size directly at once, without the need for linearizing discrete structures. The proposed model is expected to be useful only for generating small graphs due to growth of GPU memory requirements and number of parameters.

Keskar, Nitish Shirish, et al. [6] , unlike traditional models, trained CTRL here to respond to control codes, allowing users to dictate the style, content, and even task-specific behavior of the generated text. This feature enables customizable text generation, which makes it a valuable tool for various natural language processing tasks. With 1.63 billion parameters, CTRL is the largest publicly released language model to date. It controls codes so that text generation can be more easily controlled by human users. This work could have been extended to more fields of NLP, particularly by including supervised data.

Cheema, Sehrish Munawar, et al. [7] discuss a semi-automated tool that uses Natural Language Processing techniques to generate Data Flow Diagrams from user queries. The tool aims to reduce human effort and assist in decision-making. There is a comprehensive discussion of the research methodology used to develop the proposed tool, including the formulation of rules for DFD development. This can be useful for the flowchart generation project as it can provide guidance on how to develop rules for flowchart generation.The evaluation of the proposed tool's effectiveness and efficiency is limited to a specific set of test cases and may not be generalizable to other scenarios.

You, Jiaxuan, et al. [8] discuss the challenges of modeling and generating complex distributions over graphs, and how GraphRNN addresses these challenges with minimal assumptions about graph structure. They have made the code for GraphRNN and the benchmark suite is made publicly available, allowing for easy replication and extension of the study. The evaluation metrics used in this study are based on graph statistics such as which may not capture all aspects of graph structure and may not be sufficient for evaluating certain applications.

2.3 Existing Systems

Table 2. Comparison of existing system with IdeaFlow

Feature	IdeaFlow	Draw.io	LucidChart	Whimsical	Overleaf
Accessibility	IdeaFlow offers a unique way to generate flowcharts using just a simple text description No specialized diagramming knowledge is required.	Requires users to manually drag, drop, and connect shapes.	User-friendly interface, but still requires some knowledge of diagram construction.	A visually appealing interface with accessible flowchart tools.	Designed for LaTeX users, may not be intuitive for those unfamiliar with markup languages.
Automation	The core strength of IdeaFlow is its ability to automatically generate the entire flowchart from a description.	No automation features. Relies on manual creation.	Limited automation options such as templating and shape libraries.	Offers some automation with templates and streamlined interface.	Limited automation related to flowchart creation.
Accuracy	IdeaFlow seeks to understand the nuances of the provided description and accurately render the specified flowchart.	Depends fully on a user's ability to manually place and connect shapes correctly.	Relies on the user's accuracy in selecting templates and creating the flowchart.	Depends on the quality of chosen templates and the user's ability to build the flowchart correctly.	Requires precise use of LaTeX commands to generate flowcharts.
Scalability	Handles both	Scalability	Can handle	Scalability	Scalability

	simple and complex descriptions . Well-trained Generative AI models can accommodate larger diagrams.	depends on a user's efficiency in using the software.	large diagrams but may require more manual effort.	depends on user proficiency and the complexity of the flowchart.	may be limited by the complexity of flowchart syntax in LaTeX.
Customization	IdeaFlow can potentially be fine-tuned to generate flowcharts adhering to specific styles or notations.	Highly customizable in terms of shapes, colors, and styles.	Offers extensive customization options for formatting and styles.	Customization options for shapes, colors, and layout.	Limited customization in terms of flowchart appearance when using basic packages.

Chapter 3 : Requirement Gathering for the Proposed System

3.1 Introduction to requirement gathering

The proposed solution involves developing an AI system that understands and interprets textual prompts, transforming them into coherent flowchart diagrams. By using advanced generative AI models, such as GPT-3, and fine-tuning them on a diverse dataset of prompts and corresponding flowcharts, the system can comprehend the intricate relationships between text and visual representations.

The goal is to provide users with an intuitive tool that simplifies the process of visually illustrating complex processes, plans, and storylines.

3.2 Functional Requirements

Functional requirements for an AI system to generate flowcharts from text:

1. Input: The system should be able to accept text prompts in natural language, including descriptions of processes, plans, and storylines.
2. Output: The system should generate coherent flowchart diagrams that accurately represent the input text.
3. Accuracy: The generated flowcharts should be accurate and complete, capturing all of the essential steps, decisions, and outcomes in the input text.
4. Clarity: The generated flowcharts should be clear and easy to understand, using standard flowchart symbols and notation.
5. Flexibility: The system should be able to generate flowcharts of varying complexity and size, from simple diagrams to complex workflows.
6. Usability: The system should be easy to use, with a user-friendly interface that allows users to quickly and easily generate flowcharts from text prompts.

In addition to these functional requirements, the AI system should also be able to handle the following:

Incomplete or ambiguous input: The system should be able to generate reasonable flowcharts even if the input text is incomplete or ambiguous.
Multiple interpretations: If the input text can be interpreted in multiple ways, the system should allow users to select the desired interpretation or generate multiple flowcharts for each possible interpretation.
Customizable output: The system should allow users to customize the output flowcharts, such as by changing the symbols, notation, and layout.
By meeting these functional requirements, the AI system can provide a valuable tool for simplifying the process of creating flowcharts and making them more accessible to a wider range of users.

3.3 Non Functional Requirements

Non-functional requirements for an AI system to generate flowcharts from text:

1. Performance: The system should be able to generate flowcharts quickly and efficiently, even for complex input text.
2. Availability: The system should be available 24/7 with minimal downtime.
3. Scalability: The system should be able to scale to handle a large number of users and requests simultaneously.
4. Security: The system should be secure and protect user data from unauthorized access.
5. Maintainability: The system should be easy to maintain and update.

In addition to these non-functional requirements, the AI system should also be:

1. Cost-effective: The system should be affordable to develop and maintain.
2. Deployable: The system should be easy to deploy and use in a variety of environments.
3. User-friendly: The system should be easy to use for users with all levels of technical expertise.

By meeting these non-functional requirements, the AI system can be a practical and reliable tool for generating flowcharts in a variety of real-world settings.

Here are some additional non-functional requirements that may be relevant for this project:

1. Accessibility: The system should be accessible to users with disabilities.
2. Internationalization: The system should be able to handle text in multiple languages.
3. Compliance: The system should comply with all applicable regulations and industry standards.

3.4 Hardware & Software Requirements

- Hardware: Standard computer hardware with sufficient computational power for AI model training and rendering.
- Software:
  - ➢ Programming languages (Python) for AI model integration and data preprocessing
  - ➢ PythonAI Libraries: OpenAI's GPT-3 API for fine-tuning and generating flowchart content.
  - ➢ Web development frameworks.
- Tools: Text editors, version control (Git), virtual environments.
- Visualization Tools: Libraries like Matplotlib or Graphviz for rendering flowchart diagrams.

3.5 Constraints

- **Rate Limits:** The OpenAI API imposes rate limits on the number of requests that can be made within a specific time frame. To address this, we implemented rate limit handling to ensure uninterrupted operation.

- **Resource Usage:** When working with large and complex flowcharts, memory and computational resources become important considerations.

- **Connectivity:** Given that our implementation relies on web-based APIs like OpenAI's GPT-3, a stable internet connection is essential for consistent performance.

Chapter 4: Proposed Design

4.1 Block Diagram of the proposed system

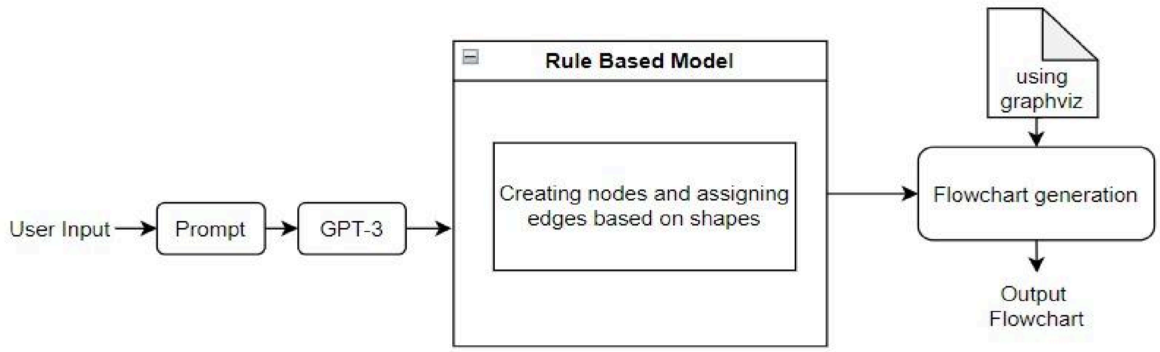

Fig 1. Block Diagram of IdeaFlow

4.2 Modular Diagram

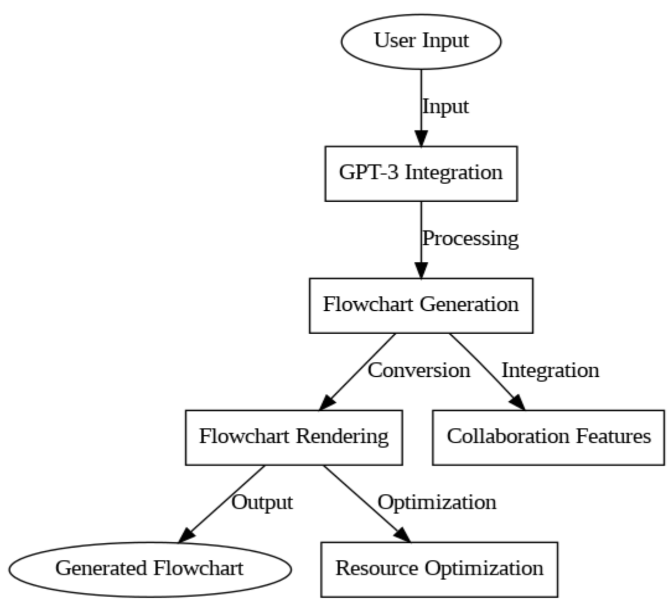

Fig 2. Modular Diagram of IdeaFlow

In the modular diagram, we identify three primary modules:

1) GPT-3 Integration: This module manages interactions with the GPT-3 API, sending prompts and receiving generated flow chart descriptions.

2) Flowchart Generation: Here, we process the descriptions and use Graphviz to create the flowchart's structural representation.

3) Rendering: The rendering module handles the final rendering of the flowchart, including shapes, labels, and connectors

4.3 Detailed Design

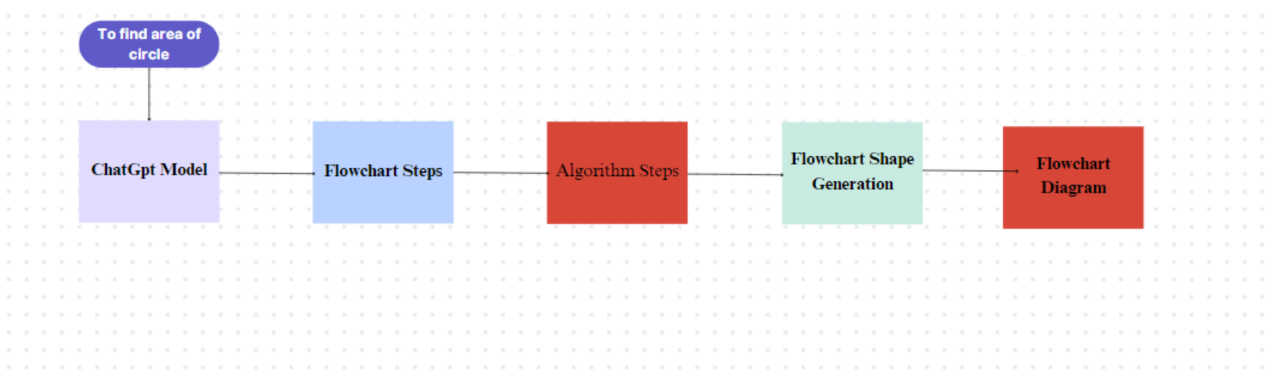

Fig 3. Interaction of model with given user prompt

5. Implementation of the Proposed System

5.1 Methodology employed for development

Data Collection: Gather a diverse dataset containing pairs of textual prompts and their corresponding flowchart diagrams. This dataset serves as the foundation for training and fine-tuning the generative AI model.

1. **Model Selection & Fine-Tuning:** Choose a powerful generative AI model like GPT-3 and fine-tune it using the collected dataset. Train the model to understand the relationship between prompts and flowchart diagrams, enabling it to generate meaningful and accurate flow charts based on textual input.
2. **Input Transformation:** Design a preprocessing mechanism to convert textual prompts into a structured format that the AI model can interpret. This might involve tokenizing the text and providing context information to guide the AI's understanding.
3. **Flowchart Language Definition:** Define a set of symbols, shapes, and rules that the AI model can use to represent flowchart elements such as processes, decisions, start/end points, and connectors.
4. **Output Rendering:** Develop a rendering engine that translates the structured output generated by the AI model into a visual flowchart. This engine needs to arrange symbols, lines, and labels to create a coherent and visually appealing diagram.
5. **User Interface & Testing:** Create an intuitive user interface where users input prompts, and the AI generates corresponding flowchart diagrams. Test the system with a variety of prompts to assess the accuracy, coherence, and usability of the generated diagrams.

Diagram

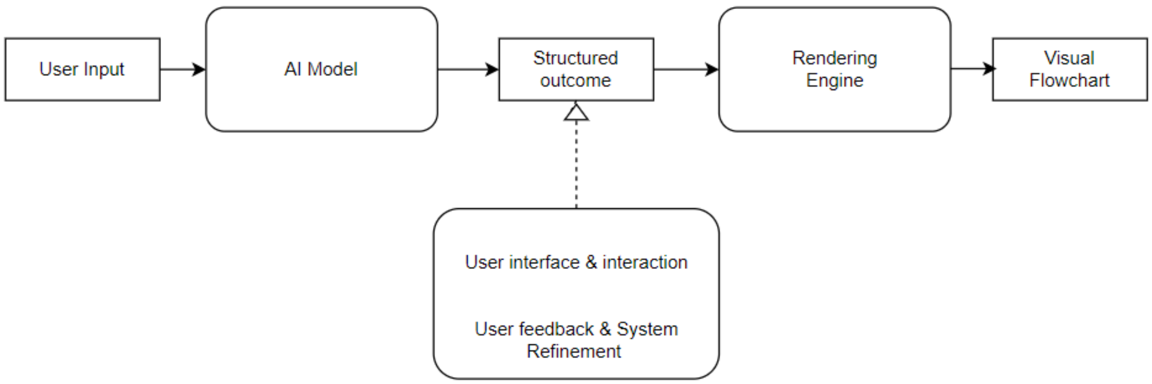

Fig 4. Methodology Diagram

- User Input: Users provide textual prompts describing processes, plans, or storylines.
- AI Model: The fine-tuned generative AI model (e.g., GPT-3) processes the input and generates structured output.
- Structured Output: The AI-generated output is a structured representation that encodes the flowchart elements.
- Rendering Engine: This component converts the structured output into a visual flowchart diagram.
- Visual Flowchart: The generated flowchart diagram is displayed to users for visualization.
- User Interface & Interaction: Users interact with the system via a user-friendly interface to input prompts and receive flowcharts.
- User Feedback & System Refinement: User feedback helps refine the AI model, rendering engine, and overall user experience.

5.2 Algorithms and flowcharts for the respective modules developed

Step 1- Prompt: The user inputs a relevant search query to create a flowchart in a specific domain.

For instance: "I want to create a flowchart to find the radius of a circle?"

Step 2 - Preprocessing of User Input: Irrelevant information, such as "I want to create a," is removed, retaining only the pertinent text: "flowchart to find the radius of a circle?"

Step -3 - GPT Model:

Interaction- The generate_flowchart function is applied to the processed input which orchestrates interaction with the OpenAI API, utilizing the GPT-3.5-turbo model. This function serves as a pivotal component in the proposed methodology as it prepares the context by using a system message and including the user's input prompt. The subsequent request to the API yields a response, from which the model's reply is extracted. This reply encapsulates the generated flowchart steps, forming a crucial element of the automated flowchart creation process.

Response- The GPT-3.5-turbo model generates a structured output, outlining flowchart steps based on the provided prompt. The generated steps are crucial for the subsequent stages of the automated flowchart creation process.

The GPT model generates a structured output based on the prompt, such as

"Step 1: Start.

Step 2: Input the value of the radius of the circle.

Step 3: Calculate the area of the circle using the formula: pi*r*r

Step 4: Output the calculated area of the circle.

Step 5: End.

Step 4 - Flowchart Steps Formatting: Generated flowchart steps undergo the removal of any extraneous information to focus on the core structure. The format_flowchart_steps function employs regular expressions to parse the raw steps, systematically splitting them based on a numerical pattern and next-line indentations. This step ensures that only relevant information is retained thereby enhancing the visual clarity and comprehension of the subsequent flowchart.

The above output will be formatted as,

“Start

Input radius

Calculate Area = pi*r*r

Output Area

End

Step 5 - Algorithmic Representation (Rule-Based Model): The formatted output derived from the GPT model is converted into an algorithmic representation called a digraph suitable for visualization using the Graphviz library. This involves identifying the labels, assigning labels with shapes based on predefined label-shape associations, and creating nodes and corresponding edges.

Following is the conversion of the above output to a digraph - this representation adheres to predefined shapes and connections

```
digraph
steps with labels and associated shapes
start [label=Start shape=oval]
input [label="Input: Radius" shape=trapezium]
calculate [label="Calculate Area = $\pi$*$r^2$" shape=box]
output [label="Output: Area" shape=trapezium]
end [label=End shape=oval]
```

Step 6 -Visual Flowchart Creation: The algorithmic representation (digraph) is then converted into a visual flowchart as shown below, using the Graphviz library, which produces a flowchart with various corresponding spaces and arrows depicting each step in the process.

Step 7 -User Interface: The Gradio interface, encapsulated within the flowchart_interface function, provides a user-friendly means to interact with the automated flowchart generation system. Upon receiving user input, it constructs a specific prompt tailored for generating numbered flowchart steps. Subsequently, it leverages the generate_flowchart function to procure the raw flowchart steps. To enhance user readability, the format_flowchart_steps function is then invoked to generate the formatted steps along with the final image output.

Chapter 6: Testing of the Proposed System

6.1 . Introduction to Flowchart Testing

This involves providing an overview of the process, emphasizing its importance in software development, and discussing the objectives and methodologies used. It may include discussing how testing ensures the accuracy and functionality of the generated flowcharts and how GenAI simplifies the testing process through automated techniques.

6.2. Types of Flowchart Tests Considered:

In this section, various types of tests considered during flowchart testing with GenAI are discussed. This may include functional testing, boundary testing, stress testing, and integration testing of the flowcharts. Each type of test serves a specific purpose in validating the correctness and robustness of the generated flowcharts.

6.3 Various Test Case Scenarios for Flowcharts:

Here, different test case scenarios considered for flowchart testing using GenAI are outlined. This involves identifying and describing various scenarios, inputs, and conditions under which the flowcharts are tested. Test case scenarios may include sequential flow, decision points, loops, and complex data structures to ensure comprehensive testing coverage.

6.4. Inference drawn from the test cases:

This section presents insights drawn from the testing of the generated flowcharts using GenAI. It includes analyzing the effectiveness and accuracy of the flowcharts, identifying any areas of improvement or refinement, and assessing the overall quality of the flowchart generation process. Additionally, the strengths and limitations of GenAI in automating flowchart testing may be discussed, along with recommendations for future enhancements.

Chapter 7: Results and Discussion

7.1. Screenshots of User Interface (UI) for the respective module

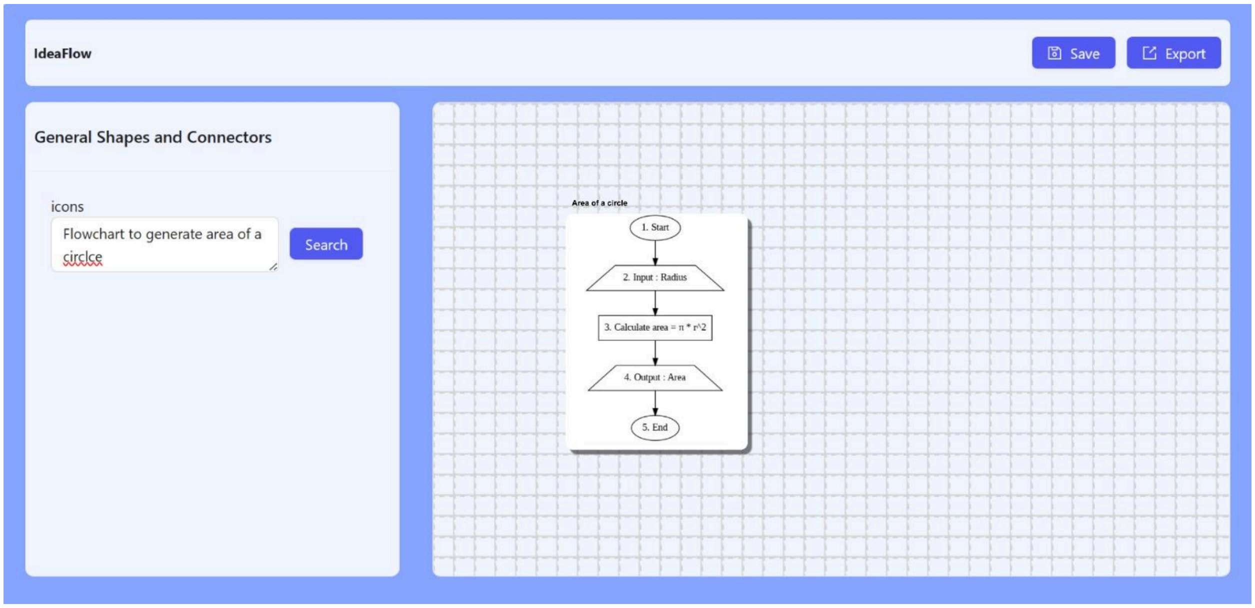

Fig. 5 Flowchart to generate area of circle

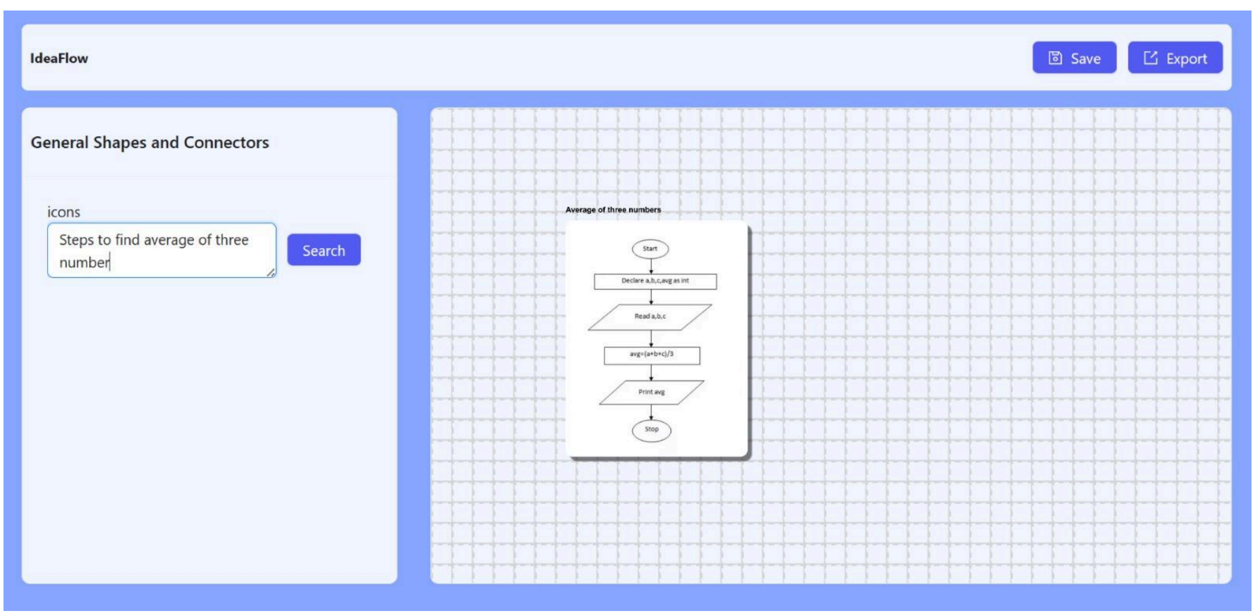

Fig. 6 Flowchart to find avg of three number

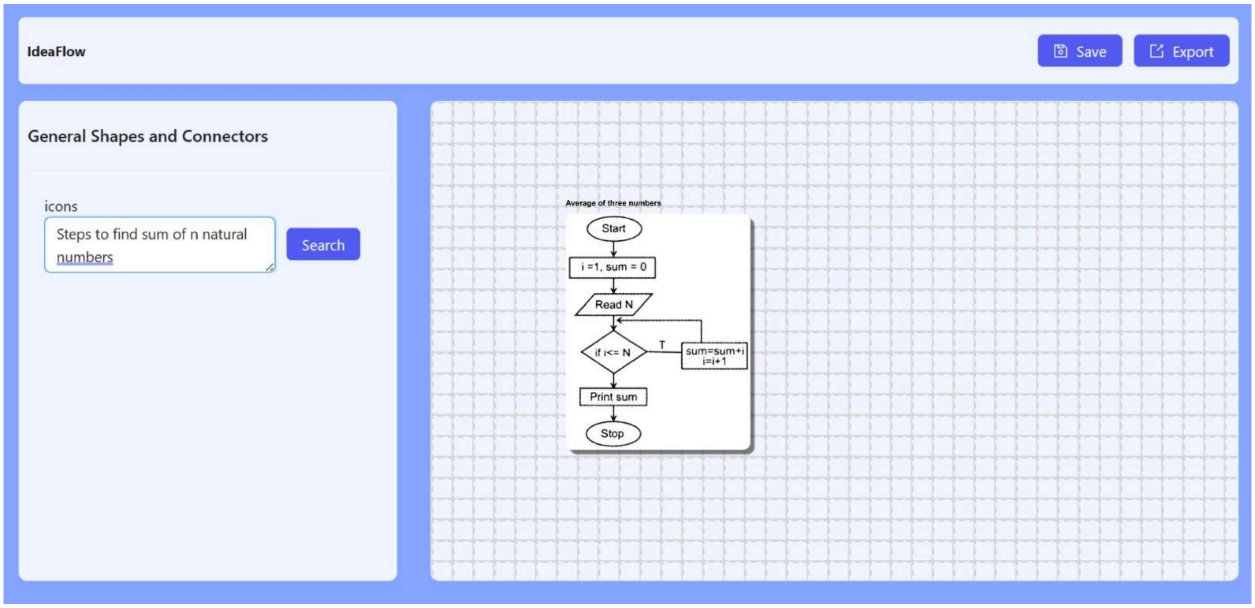

Fig. 7 Flowchart to find sum of n natural numbers

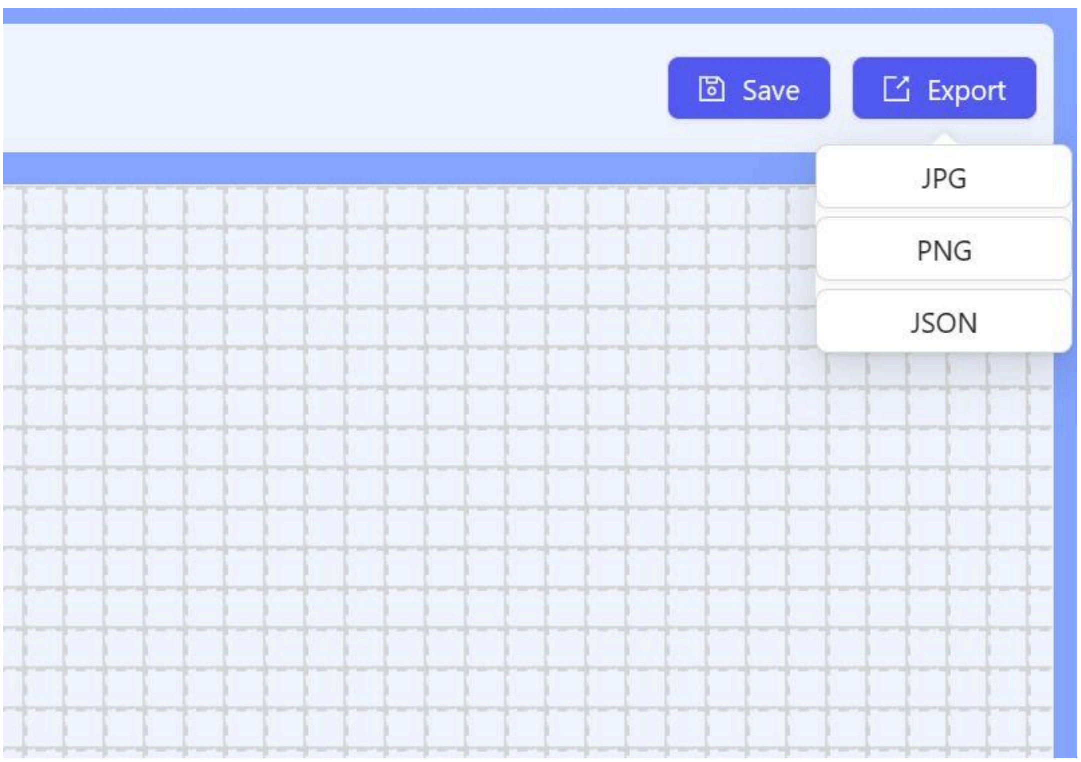

Fig 8. Types of save option

7.2. Performance Evaluation measures

We compared results from both the generative AI model and ML model to conclude which model is better for automatic flowchart creation. The metrics that project concentrates is how accurate the flowchart is, which maps to are the shapes associated with content accurately.

7.3. Comparison of results with existing systems

Table 3. Comparison of results of existing system with IdeaFlow

Feature	IdeaFlow	Draw.io	LucidChart	Whimsical	Overleaf
Accessibility	IdeaFlow offers a unique way to generate flowcharts using just a simple text description No specialized diagramming knowledge is required.	Requires users to manually drag, drop, and connect shapes.	User-friendly interface, but still requires some knowledge of diagram construction.	A visually appealing interface with accessible flowchart tools.	Designed for LaTeX users, may not be intuitive for those unfamiliar with markup languages.
Automation	The core strength of IdeaFlow is its ability to automatically generate the entire flowchart from a description.	No automation features. Relies on manual creation.	Limited automation options such as templating and shape libraries.	Offers some automation with templates and streamlined interface.	Limited automation related to flowchart creation.
Accuracy	IdeaFlow seeks to understand the nuances of	Depends fully on a user's ability to manually	Relies on the user's accuracy in selecting templates and	Depends on the quality of chosen templates and	Requires precise use of LaTeX commands to

	the provided description and accurately render the specified flowchart.	place and connect shapes correctly.	creating the flowchart.	the user's ability to build the flowchart correctly.	generate flowcharts.
Scalability	Handles both simple and complex descriptions. Well-trained Generative AI models can accommodate larger diagrams.	Scalability depends on a user's efficiency in using the software.	Can handle large diagrams but may require more manual effort.	Scalability depends on user proficiency and the complexity of the flowchart.	Scalability may be limited by the complexity of flowchart syntax in LaTeX.
Customization	IdeaFlow can potentially be fine-tuned to generate flowcharts adhering to specific styles or notations.	Highly customizable in terms of shapes, colors, and styles.	Offers extensive customization options for formatting and styles.	Customization options for shapes, colors, and layout.	Limited customization in terms of flowchart appearance when using basic packages.

Chapter 8 : Conclusion

8.1. Limitations

1. Ambiguity in Natural Language: The model may struggle to correctly interpret ambiguous or nuanced descriptions.
2. Handling Complexity: The model may have difficulty capturing complex processes or decision logic.
3. Dataset Dependence: The performance of the system is highly reliant on the quality and diversity of the training dataset.
4. Computational Resources: Training and running large generative AI models can be computationally demanding.
5. API Limits: External APIs may have rate limits or cost considerations that impact the scalability of the project.
6. Limited User Control: Users might require more granular control over the generated flowchart for fine-tuning.
7. Explainability: Understanding the reasoning behind the model's output can be difficult.
8. Bias: Potential for biases in the training data to be reflected in the generated flowcharts.
9. Accuracy and Completeness: Generated flow charts may not always be 100% accurate or perfectly reflect the input description.

8.2. Conclusion

1. Generative AI models for flowchart creation offer great potential to revolutionize how we convey complex information.
2. They can automate the production of coherent and accurate flowcharts, aiding communication and understanding.
3. Generative AI's impact on flowchart generation holds promise across education, documentation, and various sectors, with ongoing research and innovation driving this transformative technology forward.

8.3. Future Plans

Enhance Flowchart Extraction Accuracy: We will refine our methods to improve the accuracy of extracting flowchart elements from text descriptions, ensuring more faithful representation.

Optimize Resource Usage: As our projects grow in complexity, optimizing resource usage will remain a priority to ensure that the tool runs efficiently even for large and intricate flowcharts.

To expand the scope of the project by also building:
Mind Maps
Data Flow Diagrams
UML DiagramState
Chart Diagram

References

[1] Maestre, Raúl Jaime, Javier Bermejo Higuera, Nadia Gámez Gómez, Juan Ramón Bermejo Higuera, Juan A. Sicilia Montalvo, and Lara Orcos Palma. "The application of blockchain algorithms tothe management of education certificates." Evolutionary Intelligence 16, no. 6 (2023): 1967-1984.

[2] Tumati, Tarun Vihar. "SBTCERT: A SOULBOUND TOKEN CERTIFICATE VERIFICATION SYSTEM." PhD diss., CALIFORNIA STATE UNIVERSITY, NORTHRIDGE, 2023.

[3] Shrivastava, Ajay Kumar, Chetan Vashistth, Akash Rajak, and Arun Kumar Tripathi. "A decentralized way to store and authenticate educational documents on private blockchain." In 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), vol. 1, pp. 1-6. IEEE, 2019.

[4] Cheng, Jiin-Chiou, Narn-Yih Lee, Chien Chi, and Yi-Hua Chen. "Blockchain and smart contracts for digital certificates." In 2018 IEEE international conference on applied system invention (ICASI), pp. 1046-1051. IEEE, 2018.

[5] Igboanusi, Ikechi Saviour, Jae Min Lee, and Dong-Seong Kim. "Batch Minting-enabled Digital Certificates Based on Soulbound Token for Achievement Verification."

[6] Zhao, Xiongfei, and Yain-Whar Si. "NFTCert: NFT-based certificates with online payment gateway." In 2021 IEEE International Conference on Blockchain (Blockchain), pp. 538-543. IEEE,2021.

[7] Reddy, Siddhant, and Dharmender Singh Kushwaha. "Framework for privacy preserving credential issuance and verification system using soulbound token." In ITM Web of Conferences, vol. 56, p. 06002. EDP Sciences, 2023.

[8] Ndlovu, L. and Leslie, A.B., 2022. "False or Fake Qualifications in an Employment Context: A South African Perspective." Yuridika, 37(3), p.715.

[9] Certificate Authority Market Share, Size, Trends, Industry Analysis Report, By Vertical (BFSI, Retail & eCommerce, Government & Defense, Healthcare & Life Sciences, IT & Telecom, Travel & Hospitality, Education, Others); By Certificate; By Services; By Organization Size;ByRegion;SegmentForecast,

2022https://www.polarismarketresearch.com/industry-analysis/certificate-authority-market
[10]Available:https://press.careerbuilder.com/2017-12-07-Nearly-Three-in-Four-Employers-Affected-by-a-Bad-Hire-According-to-a-Recent-CareerBuilder-Survey40

[11] D'Alessandro, S., Girardi, A. and Tiangsoongnern, L., 2012. Perceived risk and trust as antecedents of online purchasing behavior in the USA gemstone industry. Asia pacific journal of marketing and logistics, 24(3), pp.433-460\

[12] Lim, Joe Onn, and Diyana Kamarudin. "NON-FUNGIBLE TOKENS: ITS POTENTIAL ROLE IN COMBATING CERTIFICATE FRAUDULENCE IN MALAYSIAN EDUCATION." International Journal of Entrepreneurship, Business and Technology 1, no. 1 (2023).

[13] Saeidnia, Hamid Reza. "Welcome to the Gemini era: Google DeepMind and the information industry." Library Hi Tech News ahead-of-print (2023).

[14] Alonso, Cristian, Tanuj Bhojwani, Emine Hanedar, Dinar Prihardini, Gerardo Uña, and Kateryna Zabska. Stacking up the benefits: Lessons from India's digital journey. International Monetary Fund, 2023.

Appendix

Plagiarism Report

Flowchart creation using Generative AI

ORIGINALITY REPORT

9%	9%	6%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.nsam.ac.in Internet Source	3%
2	www.i-scholar.in Internet Source	2%
3	www.arxiv-vanity.com Internet Source	1%
4	builtin.com Internet Source	1%
5	Submitted to Adama Science and Technology University Student Paper	1%
6	M. Abdeen. "Direct automatic generation of mind maps from text with M^{2}Gen", 2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH), 09/2009 Publication	1%
7	Submitted to University of Essex Student Paper	1%

A novel approach for creating flowcharts using Generative AI

Madhura Mhatre
Department of Computer Engineering,
Vivekanand Education Society's Institute of Technology,
Mumbai, India

Ananya Pandey
Department of Computer Engineering,
Vivekanand Education Society's Institute of Technology,
Mumbai, India

Harsh Rane
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology,
Mumbai, India

Prof. Sunita Sahu
Department of Computer Engineering,
Vivekanand Education Society's Institute of Technology,
Mumbai, India

***Abstract*—Flowcharts have been widely used to visualize and communicate complex processes in a wide variety of domains such as education, technology, science, medicine, and manufacturing. In this paper we propose a methodology to automate the process of flowchart creation using generative AI, aiming to streamline the process and enhance the efficiency of flowchart creation. By using a large dataset of labeled flowchart-text pairs, the model learns to understand the semantics and relationships within the input text and converts them into a visually coherent and accurate flowchart representation. The methodology ensures the efficiency of the model by performing data preprocessing, model training, and performance evaluation. Automating the flowchart creation process will result in cost-effectiveness as well as consistency and accuracy. We aim to harness the power of Generative AI to automate flowchart creation, making it a valuable tool for professionals seeking an efficient and reliable means of visualizing complex processes.**

***Index Terms*—Flowchart, Generative AI, Model Training, Performance Evaluation, Process Visualization.**

I. INTRODUCTION

Imagine a bustling company navigating a complex supply chain, where every step, decision point, and feedback loop needs precise and visual representation. It is here that the power of flowcharts shines by untangling intricate processes into readily comprehensible diagrams. This enables seamless communication and collaborative problem-solving.

Flowcharts are a widely used visual representation tool that helps businesses navigate complex supply chains, provide seamless communication, and enable collaborative problem-solving. A flowchart is a graphical or symbolic representation of a process, where each step in the process is represented by a different symbol and contains a short description of the process step. The symbols are linked together with arrows showing the process flow direction. Common flowchart symbols include Terminator, Process, Decision, Connector, and Data.

Flowcharts are ubiquitous across diverse industries, and statistics show that they are essential for algorithm design, program documentation, project planning, and execution. For instance, a recent study by the Association for Computing Machinery revealed that 98% of software developers leverage flowcharts daily. Similarly, a survey conducted by the Project Management Institute found that 95% of project managers consider flowcharts essential to efficient project management. In educational institutions, flowcharts have become indispensable tools for both educators and students to map out intricate processes and provide a clear understanding of complex concepts.

Despite their importance, creating flowcharts has been a manual and intricate process that consumes time and introduces inconsistencies and errors. To address these challenges, we propose a novel methodology aimed at automating the flowchart creation process. Our research paper introduces a web-based tool that harnesses generative AI, offering an automated solution for seamless flowchart creation. This entails implementing a rule-based model to develop applications for prompt-based flowchart creation, utilizing the capabilities of the OpenAI API.

In this research paper, we will describe our proposed methodology for automating flowchart creation and discuss the benefits and limitations of using generative AI for this purpose. We will also compare our approach to traditional flowchart creation methods and present a case study to demonstrate the effectiveness of our tool. Finally, we will discuss the implications of our research and suggest avenues for future work.

II. LITERATURE SURVEY

We conducted a literature survey on flowchart creation and automation tools used in the field. We discovered that there are no existing papers that specifically address the use of generative AI for creating flowcharts. Therefore, we divided our literature review into two sections: research papers that focus on generative AI and research papers that explore flowchart creation technologies and other graph representation techniques.

In [3], Stefan Feuerriegel et al. emphasize on the capacity of AI to have a transformative impact on work and communication. This paper mentions applications, economic implications, highlights, opportunities, challenges and scope for the Business Information Systems Engineering(BISE) community. Referring to this paper helped us understand more about how generative AI finds its significance throughout industries. This research acted as an introduction to our discoveries in the field of automating the generation process. Generative AI helps amplify human work in fields like creativity, information, etc. The limitations of generative AI observed in this paper are incorrect outputs, the risk of copyright violations, bias and fairness.

In [4], Hongwei Wang et al. researched and developed a generative adversarial network for graph representation. They unified two schools of graph representation learning methodologies, i.e. generative methods and discriminative methods, via training in a minimax game.They have discussed the framework of their graph generative adversarial networks. Further, they focus on the respective methodology optimization for accurate results. They performed a set of experiments spread over 5 datasets to produce their desired outputs. They conducted experiments that successfully based their theory. Due to the learning flexibility offered by adversarial training, GraphGAN improved the accuracy of arXiv-AstroPh and arXiv-GrQc by 1.18% to 4.27% and 0.59% to 11.13%, respectively Apart from that, they used their models for predictions and recommendations. The research does not have any real-world application.

In [5], M. Abdeen et al. gave us a segue for our search for finding relevant studies to our problem statement. As we know, mind maps are another efficient way of concise graph representation. Although manually constructing mind maps requires knowledge, research and time, employing generative artificial intelligence proved to be a breakthrough here. Their M2Gen tool involved multiple modules for successful generation including morphological analyzer, parser, syntax and semantic analyzer and finally the mind map conversion module. A detailed work on language processing and context recognition helped them with the M2gen tool. This is the first tool that has been deployed in the context of automating mindmaps. Due to the vastness of the English language, the model accepted only a subset of the data.

In [6], Mirza Mehdi and Simon Osindero introduced the conditional version of generative adversarial nets, which can be constructed by simply feeding the data we wish to condition on both the generator and discriminator. Generative adversarial networks are innovative ways of training generative models. Whereas in the conditional generative model, conditioning the model on extra information allows us to decide the direction that the generation process follows. Related work follows the multi-modal learning for image labeling. This section discusses the various issues faced by supervised neural networks, primarily the inefficiency of accommodating a large number of prediction categories and the prominence of learning one-to-one mappings and their corresponding solutions. The study also illustrates how conditional generative adversarial networks can be used in multi-modal models and other examples.

Although the results provided are preliminary, they act as boosts in the creation of various new creative applications. The research does not provide a thorough and concrete way of the performance measurement.

In [7], Martin Simonovsky and Nikos Komodakis discussed how deep learning on graphs has always concentrated on learning graph embedding tasks, as opposed to advances in generative models. The related work walks us through existing technologies for graph decoding, discrete data decoding and molecule decoding. The study goes against the flow and attempts to generate small graphs using methods formulated as variational autoencoders. The research provided applications in cheminformatics and visualization of the organic molecules belonging to the QM9 and ZINC datasets. Future scope presents itself in the form of optimized incorporation of prior distribution or adding recurrent mechanisms for correcting mistakes.

In [8], Nitish Shirish Keskar et al. introduced CTRL, a 1.63 billion-parameter conditional transformer language model. The lacuna for this research lies in the limited availability of customization in the text generation capabilities of large-scale language models.CTRL is a conditional language model that is conditioned on a control code and learns the distribution of the code over the various sequences of symbols of the language. They train on 140GB text drawings from a variety of domains such as Wikipedia. The future scope of this work includes finer-grained control, extending the study to other aspects of NLP and making the interface between user and models more intuitive and more exclusive.

Sehrish Munawar Cheema, Saman Tariq and Ivan Miguel Pires [9] developed a natural language interface for the automatic generation of data flow diagrams. They made use of web extraction techniques for the dataset purposes. The research methodology consisted of steps such as data collection, extraction of terms, data repositioning and mapping the data with a rule-based algorithm. A semi-automated tool was developed for breaking various business logic into DFD. The validity of the system was verified with the corresponding state-of-the-art DFDs for the selected business logic. Enhancement of the tool for automatic detection and generation is required.

In [10], Dias, Rukshan Piyumadu, et al. presented a unique way of generating use case diagrams by analyzing the given user input (prompt) using NLP and ML. The software development life cycle makes very good use of use case diagrams.

Hence, introducing AI can help in automating the designing process and save a lot of manual effort and time. This study also swims through the various features required for efficient UCD generation and deals with the issues that the current systems face. The proposed system follows four approaches: the spelling correction, the NLP approach, the ML approach and finally the diagram generation. They also were able to formulate the confusion matrix which makes it easier for performance analysis. The accuracy of the ML classification model is 75.52%. Apart from that, the NLP approach successfully identified the appropriate actor 86.67% of the time and 71.68% of the use cases were correctly generated.

TABLE I
STUDY OF EXISTING TOOLS FOR FLOWCHART CREATION

Ref	Tools Name	Description	Advantages	Limitations
11	/draw.io	Draw.io provides shapes for diagram creation. Users write content and connect shapes to form diagrams.	• Allows users to create their own templates. • Can connect draw.io with Google Drive.	• Flow charts need to be created manually. • Lack of proper tutorials and technical support. • Does not support complex diagrams.
12	Whimsical	Whimsical.com allows users to create and share visual documents and boards. Provides AI-generated outputs.	• Unified workspace for thinking and collaboration. • Fast, intuitive, and fun to use. • Rich library of visual elements and templates.	• Difficulty collaborating on large projects. • Performance issues. • Cluttered interface. • Comments option not available.
13	Taskade	Taskade enables organization, visualization, and mapping of projects.	• Real-time collaboration. • Video chat. • File sharing. • Comments and mentions.	• No integrations with other apps. • Small file upload limit. • No support for complex projects.
14	Overleaf	Overleaf is a collaborative cloud-based LaTeX editor used for scientific documents.	• Collaborate on flowcharts with others. • Export flowcharts to various formats. • Integrate with tools like Jira and GitHub.	• Requires knowledge of LaTeX. • Can be slow. • Not as user-friendly as some tools.
15	Lucidchart	Lucidchart is a web-based diagramming tool for various diagrams.	• Wide range of shapes and connectors. • Export diagrams to various formats. • Collaborate on diagrams. • Integrate with other tools.	• Can be slow. • Not as user-friendly. • Free plan is limited.

A. *Streamlined Flowchart Creation with Generative AI*

Generative AI is a form of artificial intelligence in which algorithms automatically produce content in the form of text, images, audio and video. These systems are trained on massive amounts of data and work by predicting appropriate output.

The application utilizes a Generative Pre-trained Transformer (GPT) model to extract the key steps and decision points from a user-provided text prompt. This generated flowchart structure is then refined and enhanced using a Rule-Based Model, ensuring proper labeling, shape association, and graphical context.

III. PROPOSED SYSTEM

We propose a novel web application that leverages Generative AI for automatic flowchart creation, streamlines the process to save time, eliminates the need for manual intervention, and ensures consistency. This empowers users, regardless of their expertise, to generate visually appealing and error-free representations of complex processes effortlessly.

The proposed system for flowchart creation elucidates the step-by-step method which involves the following: A user initiates the flowchart creation process by entering a prompt or input text. Subsequently, the text undergoes preprocessing to eliminate noise and irrelevant information. The preprocessed text is then input into a Generative Pre-trained Transformer (GPT) model, responsible for generating flowchart steps.

These steps are further processed by a Rule-Based Model, illustrated in Fig. 1 In this model, each step is analyzed to identify labels such as 'Start,' 'Input,' 'Output,' 'If,' 'Calculate,' 'Print,' 'End,' and others, without considering case sensitivity. Secondly, each label is linked to a predefined shape, such as Start [shape = oval], Input [shape = parallelogram], Calculate [shape = box], If [shape = diamond]. Thereafter, these identified labels and shapes are simultaneously associated with each step. For example,
step_0 [label="1. Start." shape=oval],
step_1 [label="2. Input the value of the radius of the circle." shape=parallelogram].

In a graphical context, each step is treated as a node, with the arrows connecting two steps forming links or edges in the flowchart. This step involves the creation of nodes and the assignment of edges like

The structured output from the rule-based model is transformed into an algorithmic representation, specifically a digraph, using a custom algorithm. Subsequently, the Graphviz library converts this algorithmic representation into a visual flowchart.

The generated flowchart is then displayed on the user

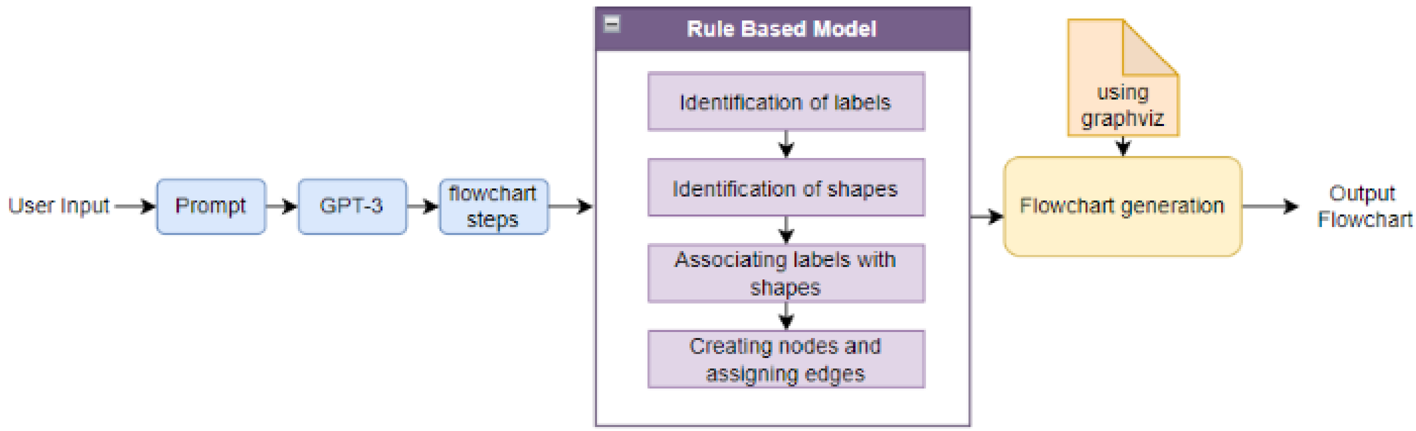

Fig. 1. Stages of flowchart creation

interface, facilitating user interaction and comprehension.

IV. Methodology

This section outlines the detailed methodology employed for automating the generation of flowchart steps, incorporating the OpenAI GPT-3.5-turbo model and the Graphviz library that finally gives the output.

Step 1 - Prompt: The user inputs a relevant search query to create a flowchart in a specific domain.
For instance: "I want to create a flowchart to find the radius of a circle?"

Step 2 - Preprocessing of User Input: Irrelevant information, such as "I want to create a," is removed, retaining only the pertinent text: "flowchart to find the radius of a circle?"

Step 3 - GPT Model:
Interaction- The generate_flowchart function is applied to the processed input which orchestrates interaction with the OpenAI API, utilizing the GPT-3.5-turbo model. This function serves as a pivotal component in the proposed methodology as it prepares the context by using a system message and including the user's input prompt. The subsequent request to the API yields a response, from which the model's reply is extracted. This reply encapsulates the generated flowchart steps, forming a crucial element of the automated flowchart creation process.

Response- The GPT-3.5-turbo model generates a structured output, outlining flowchart steps based on the provided prompt. The generated steps are crucial for the subsequent stages of the automated flowchart creation process.

The GPT model generates a structured output based on the prompt, such as

"Step 1: Start.
Step 2: Input the value of the radius of the circle.
Step 3: Calculate the area of the circle using the formula: $A = \pi \cdot r^2$ *(where* A *represents the area and* r *represents the radius).*
Step 4: Output the calculated area of the circle. Pour hot water over the coffee grounds.
Step 5: End."

Step 4 - Flowchart Steps Formatting: Generated flowchart steps undergo the removal of any extraneous information to focus on the core structure. The format_flowchart_steps function employs regular expressions to parse the raw steps, systematically splitting them based on a numerical pattern and next-line indentations. This step ensures that only relevant information is retained thereby enhancing the visual clarity and comprehension of the subsequent flowchart.

The above output will be formatted as,
"Start
Input radius
Calculate Area $= \pi * r^2$
Output Area
End"

Step 5 - Algorithmic Representation (Rule-Based Model): The formatted output derived from the GPT model is converted into an algorithmic representation called a digraph suitable for visualization using the Graphviz library. This involves identifying the labels, assigning labels with shapes based on predefined label-shape associations, and creating nodes and corresponding edges.

Following is the conversion of the above output to a digraph - this representation adheres to predefined shapes and connections,

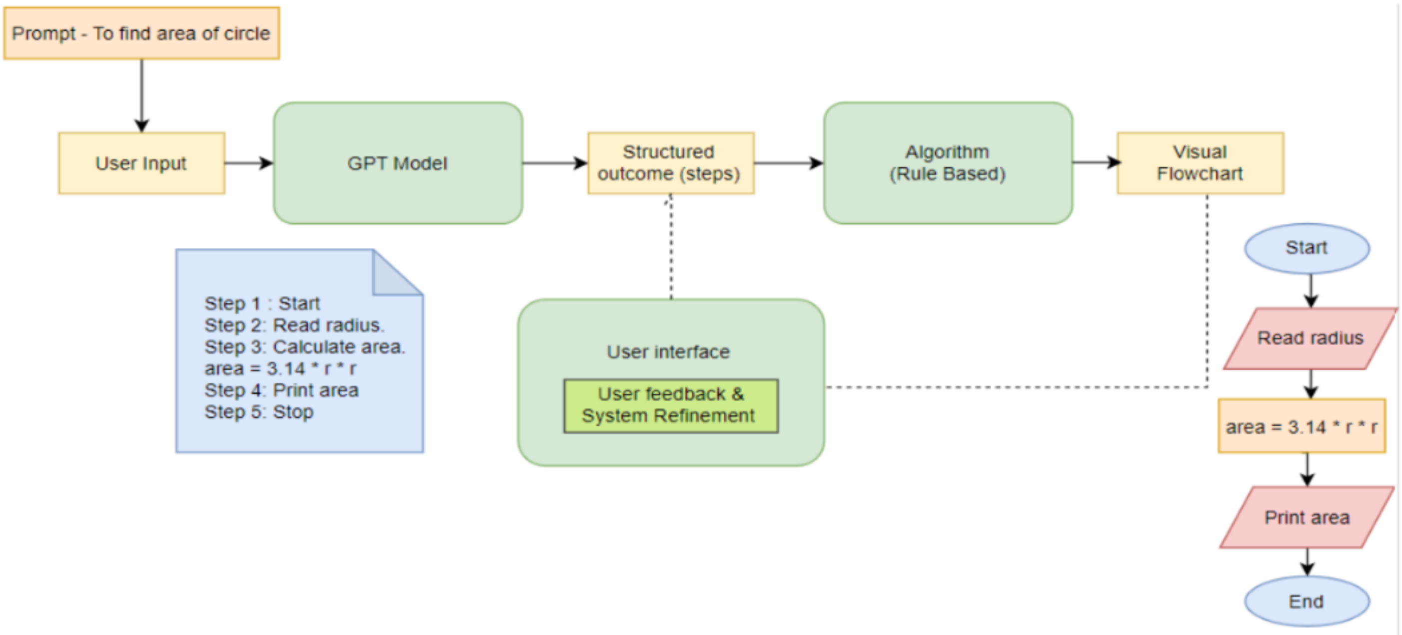

Fig. 2. Methodology Diagram

digraph {
// steps with labels and associated shapes
start [label=Start shape=oval]
input [label="Input: Radius" shape=trapezium]
calculate [label="Calculate Area = $\pi * r^2$*" shape=box]*
output [label="Output: Area" shape=trapezium]
end [label=End shape=oval]
// link or edge that connects nodes(steps)
start - input
input - calculate
calculate - output
output - end
}

Step 6 -Visual Flowchart Creation: The algorithmic representation (digraph) is then converted into a visual flowchart as shown below, using the Graphviz library, which produces a flowchart with various corresponding spaces and arrows depicting each step in the process.

Step 7 -User Interface: The Gradio interface, encapsulated within the flowchart_interface function, provides a user-friendly means to interact with the automated flowchart generation system. Upon receiving user input, it constructs a specific prompt tailored for generating numbered flowchart steps. Subsequently, it leverages the generate_flowchart function to procure the raw flowchart steps. To enhance user readability, the format_flowchart_steps function is then invoked to generate the formatted steps along with the final image output. This interface offers a seamless experience for researchers, students, teachers, and practitioners. They can also provide feedback for the overall betterment of this proposed model.

V. Implementation

The development of the user interface has been meticulously executed to produce a sophisticated web-based platform. This platform empowers users to input their specific requirements for a flowchart with precision and ease. Through a seamless blend of intuitive controls and interactive elements, users are guided through the process of defining the logic and structure of their desired flowchart.

Beneath the surface, the implementation harnesses the capabilities of advanced Python libraries, notably Graphviz. Renowned for its adeptness in generating intricate decision diagrams, Graphviz is seamlessly integrated into the system. This integration ensures that the generation of flowchart images is not only efficient but also streamlined, offering users a smooth and responsive experience.

The user-provided prompt undergoes a meticulous conversion process, resulting in the creation of a digraph code. This code serves as a critical intermediary, facilitating the mapping of statements to corresponding flowchart shapes. Moreover, it functions as the foundational blueprint for generating the flowchart image. Additionally, users can conveniently save this code for future reference or export it in alternative formats, enhancing the flexibility and usability of the platform.

Upon finalizing the input parameters, users are presented with the option to effortlessly download the resulting flowchart image. This user-centric approach ensures that users can obtain their customized flowchart representations with minimal hassle and inconvenience.

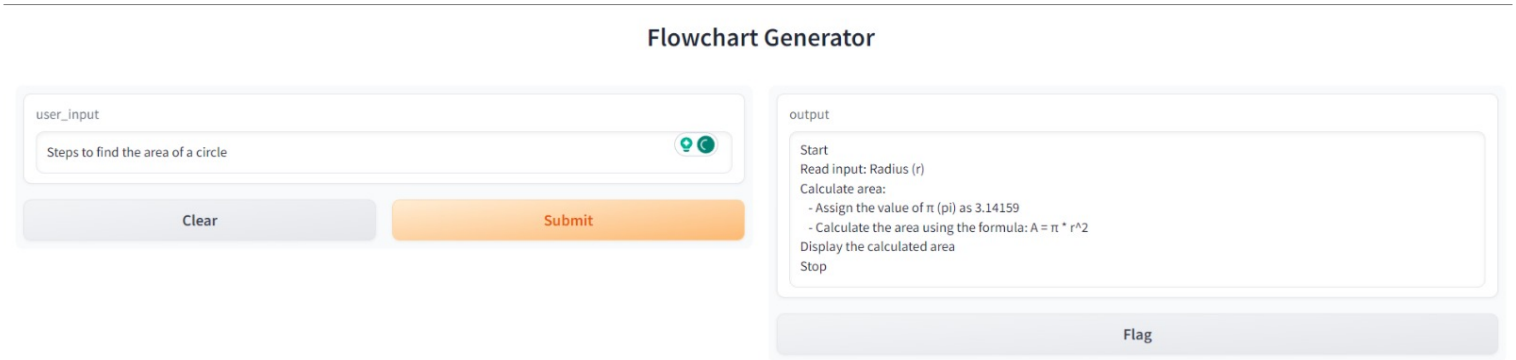

Fig. 3. Intermediary steps for the corresponding prompt.

The digraph code, shown in figure 4, provides a comprehensive representation of the shapes associated with each statement within the generated flowchart. It offers valuable insights into the structural organization and logical flow of the flowchart, elucidating decision points, conditional statements, and sequential actions. As a fundamental tool for researchers, the digraph code enriches their understanding of the flowchart's composition and facilitates in-depth analysis and interpretation.

flowchart.png flowchart

```
digraph {
  step_0 [label="1. Start" shape=oval]
  step_1 [label="2. Input : Radius" shape=trapezium]
  step_2 [label="3. Calculate area = π * r^2" shape=box]
  step_3 [label="4. Output : Area" shape=trapezium]
  step_4 [label="5. End" shape=oval]
  step_0 -> step_1
  step_1 -> step_2
  step_2 -> step_3
  step_3 -> step_4
}
```

Fig. 4. Example of generated digraph code.

The generated flowchart image, as shown in figure 5, presents a visual representation of the logical structure and decision-making process defined by the user's input i.e. to calculate the area of a circle. Comprising various shapes and connecting arrows, the flowchart offers a clear and concise depiction of the sequence of steps and decision points involved in the task or process under consideration.

The flowchart is organized hierarchically, with the top-down flow indicating the sequential progression of steps. Decision points branch out into multiple paths, reflecting different possible outcomes or conditions based on the criteria defined by the user.

In addition to visualizing the logical flow of the process, the flowchart may include annotations or labels to provide additional context or explanation for each step or decision point. These annotations enhance the clarity and comprehensibility of the flowchart, aiding users in understanding the underlying logic and rationale behind each action or decision.

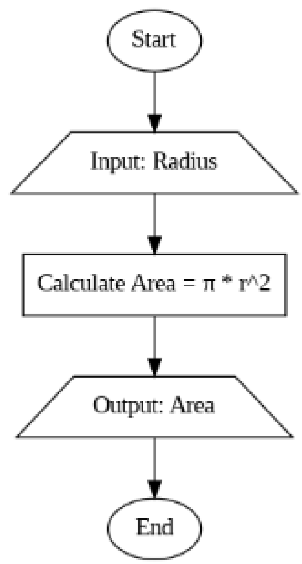

Fig. 5. Example of generated flowchart.

VI. Conclusion

Flowcharts are a widely used tool for visually representing workflows and procedures across various industries, ranging from software development to educational institutions. According to surveys conducted by the Association for Computing Machinery and Project Management Institute, flowcharts have proven to be extensively used for achieving visual clarity, effective communication, coordination, analysis, and problem-solving.

We are proud to propose a simple yet effective methodology for automating the creation of flowcharts using Generative AI and Natural Language Processing. Our methodology consists of three main modules: data preprocessing, model training, and user interface. As manual flowchart creation is a time-consuming and error-prone process that requires expertise and skills, our proposed methodology enables people from non-technical backgrounds to use a flowchart creation tool and leverage its advantages. With our automated process, users only need to enter the prompt for which the flowchart needs to be created, and the tool produces precise flowcharts effortlessly.

In conclusion, our proposed methodology for automating flowchart creation through Generative AI introduces a novel and efficient approach to streamline the traditionally manual and time-consuming process. We leverage a Generative Pre-trained Transformer (GPT) model and a Rule-Based Model to generate visually coherent and accurate flowcharts effortlessly. The structured methodology, encompassing data preprocessing, model training, and user interface, enhances efficiency and caters to users of varying expertise levels. This automated approach ensures precision, consistency, and significant time savings, making flowchart creation accessible and beneficial across diverse domains. As we navigate the complexities of various industries, from software development to education, the application of Generative AI proves to be a valuable tool for professionals seeking an efficient and reliable means of visualizing intricate processes.

In our future endeavors, we aspire to broaden the horizons of our flowchart generation tool by incorporating diverse domains. We also intend to enhance the user experience by incorporating an interactive interface that would enable users to modify the shape of any instruction or step as per their requirements. Furthermore, we aim to expand our output formats beyond the currently supported jpeg, png, and json formats. This research paper presents an automated and time-efficient methodology for creating flowcharts that can be utilized by anyone, regardless of their technical expertise."

References

[1] E. Myasnikova and A. Spirov, "Comparative analysis of algorithmic approaches simulating the directed evolution techniques," 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, pp. 660-665, doi: 10.1109/SIBIRCON56155.2022.10016777.

[2] R. G. Staples, "Flowcharting—an aid to project management," Project Management Quarterly, vol. 3, no. 3, pp. 7–10, 1972.

[3] S. Popenici et al., "A critical perspective on generative AI and learning futures. An interview with Stefan Popenici," Journal of Applied Learning and Teaching, vol. 6, no. 2, 2023.

[4] H. Wang et al., "Graphgan: Graph representation learning with generative adversarial nets," Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1, 2018.

[5] M. Abdeen et al., "Direct automatic generation of mind maps from text with M 2 Gen," 2009 IEEE Toronto International Conference Science and technology for Humanity (TIC-STH), 2009.

[6] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.

[7] M. Simonovsky and N. Komodakis, "GraphVAE: Towards generation of small graphs using variational autoencoders," Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I, vol. 27, 2018.

[8] N. S. Keskar et al., "Ctrl: A conditional transformer language model for a controllable generation," arXiv preprint arXiv:1909.05858, 2019.

[9] S. M. Cheema, S. Tariq, and I. M. Pires, "A natural language interface for automatic generation of data flow diagrams using web extraction techniques," 2023.

[10] R. P. Dias et al., "Automated use case diagram generator using NLP and ML," arXiv preprint arXiv:2306.06962, 2023.

[11] G. Alder, "About draw.io," Draw.io, [Online]. Available: https://www.drawio.com/about. [Accessed: date].

[12] M. Schindler and S. Schoeffel, "Whimsical crunchbase profile," Whimsical, [Online]. Available: https://whimsical.com/. [Accessed: date].

[13] J. Xie, D. Loire, and S. Chang, "Taskade's website," Taskade, [Online]. Available: https://www.taskade.com/about. [Accessed: date].

[14] J. Hammersley and J. Lees-Miller, "Overleaf," Overleaf, [Online]. Available: https://www.overleaf.com/about. [Accessed: date].

[15] B. Dilts and K. Sun, "LucidChart Official Website," LucidChart, [Online]. Available: https://lucid.co/. [Accessed: date].

[16] PMI, "Flowcharting: Project management's critical decision-making tool," PMI, [Online]. Available: https://www.pmi.org/learning/library/flowcharting-project-management-critical-decisions-5759. [Accessed: date].

[17] IEEE Xplore, "IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)," IEEE Xplore, [Online]. Available: https://ieeexplore.ieee.org/document/10016777. [Accessed: date].

Review Sheets

Review 1:

Inhouse/ Industry_Innovation/Research:

Sustainable Goal: Goal 9 Build resilient infrastructure, sustainable industrialization, & foster innovation

Project Evaluation Sheet 2023 - 24

Class: D17 A/B/C

Group No.: 43

Title of Project: Flowchart creation using Generative AI

Group Members: Madhura Mhatre (32), Ananya Pandey (42), Harsh Rane (48)

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
05	04	04	03	05	02	02	02	02	02	02	03	03	03	04	46

Comments: Domain specific flowchart generation; Comparison between proposed & existing systems

Dr. G Bhatia

Name & Signature Reviewer1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
05	04	04	03	05	02	02	02	02	02	02	03	03	03	03	45

Comments:

Date: 10th february, 2024

Priyanka.

Name & Signature Reviewer 2

Review 2

Inhouse/ Industry_Innovation/Research:

Sustainable Goal:

Project Evaluation Sheet 2023 - 24

Class: D17 A/B/C

Group No.: 43

Title of Project: Flowchart creation using Generative AI

Group Members: Madhura Mhatre (32), Ananya Pandey (42), Harsh Rane (48)

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
05	04	04	02	04	02	02	02	02	02	02	02	01	02	04	40

Comments:

Dr. Grishma Bhatia 7/3/24

Name & Signature Reviewer1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
05	04	04	02	04	02	02	02	02	02	02	02	01	02	04	40.

Comments:

Date: 9th March, 2024

Priyanka Nagdev

Name & Signature Reviewer 2