SKETCHIFY

Submitted in partial fulfillment of the requirements of the degree of

BACHELOR OF COMPUTER ENGINEERING

by

Sahil Limaye - 20102115

Atharva Karekar - 20102117

Eeshan Joshi - 20102012

Jay Mohitepatil - 20102166

Guide:

Prof. Sachin Takmare

Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

(2022-2023)

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

CERTIFICATE

This is to certify that the Mini Project 2B entitled "**SKETCHIFY**" is a bonafide work of **"Sahil limaye(20102115), Atharva Karekar(20102117), Eeshan Joshi(20102012), Jay Mohitepatil(20102166)"** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Engineering.**

Guide: Prof.Sachin Takmare	Project Coordinator: Prof. D.S. Khachane	Head of Department Prof. S.H. Malave

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

Project Report Approval for Mini Project-2B

This project report entitled "**SKETCHIFY"** by Sahil Limaye***, Atharva Karekar, Eeshan Joshi, Jay Mohitepatil*** is approved for the partial fulfillment of the degree of ***Bachelor of Engineering*** in ***Computer Engineering***, ***2022-23***.

Examiner Name Signature

1._____ _____

2._____ _____

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I 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.

Sahil Limaye – 20102115

Atharva Karekar – 20102117

Eeshan Joshi – 20102012

Jay Mohitepatil – 20102166

Date:

Abstract

This report aims to explore the current state-of-the-art techniques in the field of AI text-to-image generation. The report discusses the challenges and opportunities presented by this technology, including the potential applications of AI-generated images in various domains, such as art, advertising, and entertainment. The report provides an overview of the different approaches used for text-to-image generation, including deep learning techniques, generative adversarial networks (GANs), and attention-based models. The report also discusses the key evaluation metrics used to assess the quality of the generated images, such as perceptual realism and semantic accuracy. Finally, the report presents some of the current limitations and future directions for research in this field, including the need for better data sets and the development of more efficient algorithms for text-to-image generation.

One of the most widely used approaches for text-to-image generation is the use of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These models are trained on large datasets of images and corresponding textual descriptions to learn the mapping between text and images.

In recent years, attention-based models have also emerged as a promising approach for text-to-image generation. These models use attention mechanisms to selectively focus on different parts of the textual description while generating the corresponding image.

Keywords: AI text-to-image generation, Deep learning techniques, Generative adversarial networks (GANs), Attention-based models, Evaluation metrics, Perceptual realism

CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

Artificial intelligence (AI) has revolutionized many areas of modern life, from healthcare and finance to transportation and entertainment. One area of AI research that has gained significant attention in recent years is AI text-to-image generation. This technology uses machine learning algorithms to generate images based on textual descriptions, and it has the potential to revolutionize various domains, such as advertising, art, and gaming.AI text-to-image generation is a complex and challenging task that requires the development of sophisticated deep learning models that can learn the mapping between text and images. Researchers have explored various approaches for text-to-image generation, including deep learning techniques, generative adversarial networks (GANs), and attention-based models. Each approach has its strengths and weaknesses, and the choice of approach often depends on the specific application and dataset.The evaluation of AI-generated images is a challenging task, as it requires a balance between perceptual realism and semantic accuracy. Perceptual realism refers to how visually realistic the generated image appears to human observers, while semantic accuracy refers to how well the image represents the corresponding textual description.

The purpose of this report is to provide an overview of the current state-of-the-art techniques in the field of AI text-to-image generation. The report discusses the challenges and opportunities presented by this technology, the different approaches used for text-to-image generation, the key evaluation metrics used to assess the quality of the generated images, and some of the current limitations and future directions for research in this field.

Text to image AI generator is an exciting technology that has emerged in recent years. This technology enables computers to generate images based on natural language descriptions provided by users. It uses deep learning algorithms to analyze and understand the meaning of text and then creates an image that matches the description.The technology behind text to image AI generators is based on two main components: Natural Language Processing (NLP) and Generative Adversarial Networks (GANs). NLP helps the computer understand the context and meaning of the text, while GANs generate images that match the text description.

Text to image AI generators have a wide range of applications, from generating images for marketing and advertising to creating art and visual storytelling. This technology has become increasingly popular in the e-commerce industry, where companies use it to create product images without the need for expensive photo shoots. It has also been used in the fashion industry, where designers use it to create clothing designs based on text descriptions.One of the major advantages of text to image AI generators is that they can create images that would be difficult or impossible for humans to produce. For example, they can generate images of fictional characters or imaginary landscapes that do not exist in the real world. This makes them ideal for use in video games, movies, and other forms of entertainment.Another advantage of text to image AI generators is that they can save time and money for businesses. Instead of hiring a photographer or graphic designer to create images, companies can use this technology to generate them automatically. This can be especially useful for small businesses with limited budgets.However, there are also some limitations to text to image AI generators. For example, they are not yet able to create images that are as realistic or detailed as those created by humans. They also have difficulty understanding abstract concepts or emotions, which can limit their ability to create certain types of images.

Chapter 2

Literature Survey

[1]Reed, S., et al. "Generative adversarial text to image synthesis." Proceedings of the 33rd International Conference on Machine Learning. 2016.

This paper introduced a deep generative model that uses convolutional and recurrent neural networks to generate images from textual descriptions.The convolutional neural network (CNN) component of the model is responsible for processing the input text description and extracting its features. The recurrent neural network (RNN) component, on the other hand, generates the corresponding image by conditioning on the extracted features from the CNN. This approach allows the model to capture the complex relationships between textual descriptions and corresponding visual features.

[2] Zhang, H., et al. "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks." Proceedings of the IEEE International Conference on Computer Vision. 2017.

This paper proposed a GAN-based model that uses attention mechanisms to selectively focus on different parts of the textual description while generating the corresponding image. Recent advancements in deep learning have enabled the development of generative models that can produce realistic images from textual descriptions. One such approach is the use of Generative Adversarial Networks (GANs) that have shown impressive results in image generation tasks. However, generating images from text is a challenging task because textual descriptions can be lengthy and complex, requiring a model to understand the semantics and context of the text to generate meaningful images.One such deep generative model, introduced in a recent paper, employs a combination of convolutional and recurrent neural networks to generate images from textual descriptions. The model is based on a conditional generative adversarial network (cGAN) architecture

[3]Xu, K., et al. "Attngan: Fine-grained text to image generation with attentional generative adversarial networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

This paper introduced an attention-based model that uses a combination of convolutional and recurrent neural networks to generate images from textual descriptions.The CNN component of the model extracts the textual features, which are then used to compute the attention weights. These weights determine the importance of each word in the input text description and guide the RNN in generating the corresponding image. The RNN generates the image by iteratively updating its hidden state based on the textual features and the attention weights. The output of the RNN is then passed through a decoder network, which generates the final image.

[4] Hong, S., et al. "Inferring semantic layout for hierarchical text-to-image synthesis." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

This paper proposed a hierarchical text-to-image synthesis model that infers the semantic layout of the scene from the textual description and generates images accordingly.The proposed model consists of two main components: a layout generation network and an image generation network. The layout generation network is responsible for inferring the semantic layout of the scene from the textual description. It employs a hierarchical attention mechanism to focus on different parts of the input text and generate a corresponding layout representation. The image generation network, on the other hand, takes the layout representation as input and generates the corresponding image.Its hierarchical approach to generating images from text allows it to capture the complex relationships between textual descriptions and corresponding visual features, leading to more realistic and diverse images.

[5]Wang, C., Chai, M., He, M., Chen, D., & Liao, J. (2022). CLIP-NeRF:Text-and-Image Driven Manipulation of Neural Radiance Fields. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 3835–3844.

The paper presents a method called "CLIP-NeRF" that combines text and images to manipulate Neural Radiance Fields (NeRFs).NeRFs are a recent breakthrough in 3D scene reconstruction and rendering, which allow for highly realistic 3D visualizations. a layout generation network and an image generation network. The layout generation network is responsible for inferring the semantic layout of the scene from the textual description. It employs a hierarchical attention mechanism to focus on different parts of the input text and generate a corresponding layout representation

[6]Brownlee, J. (2019). Deep learning for computer vision: image classification,object detection, and face recognition in python. Machine Learning Mastery.Campo, M. del, & Leach, N. (2022). Can Machines HallucinateArchitecture? Architectural Design, 92(1), 6–13.

The paper presents a method called "CLIP-NeRF" that combines text and images to manipulate Neural Radiance Fields (NeRFs).NeRFs are a recent breakthrough in 3D scene reconstruction and rendering, which allow for highly realistic 3D visualizations.The recurrent neural network (RNN) component, on the other hand, generates the corresponding image by conditioning on the extracted features from the CNN. This approach allows the model to capture the complex relationships between textual descriptions and corresponding visual features.

Research Paper	ANALYSIS
[1]Reed, S., et al. "Generative adversarial text to image synthesis." Proceedings of the 33rd International Conference on Machine Learning. 2016.	This paper introduced a deep generative model that uses convolutional and recurrent neural networks to generate images from textual descriptions
[2] Zhang, H., et al. "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks." Proceedings of the IEEE International Conference on Computer Vision. 2017.	This paper proposed a GAN-based model that uses attention mechanisms to selectively focus on different parts of the textual description while generating the corresponding image.
[3]Xu, K., et al. "Attngan: Fine-grained text to image generation with attentional generative adversarial networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.	This paper introduced an attention-based model that uses a combination of convolutional and recurrent neural networks to generate images from textual descriptions
[4] Hong, S., et al. "Inferring semantic layout for hierarchical text-to-image synthesis." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.	This paper proposed a hierarchical text-to-image synthesis model that infers the semantic layout of the scene from the textual description and generates images accordingly

Research Paper	ANALYSIS
[5]Wang, C., Chai, M., He, M., Chen, D., & Liao, J. (2022). CLIP-NeRF:Text-and-Image Driven Manipulation of Neural Radiance Fields. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 3835–3844.	The paper presents a method called "CLIP-NeRF" that combines text and images to manipulate Neural Radiance Fields (NeRFs). NeRFs are a recent breakthrough in 3D scene reconstruction and rendering, which allow for highly realistic 3D visualizations.
[6]Brownlee, J. (2019). Deep learning for computer vision: image classification,object detection, and face recognition in python. Machine Learning Mastery.Campo, M. del, & Leach, N. (2022). Can Machines HallucinateArchitecture? Architectural Design, 92(1), 6–13.	The book covers topics such as convolutional neural networks, image classification, object detection, and face recognition, and provides practical examples and tutorials using the Python programming language.

Chapter 3

Problem Statement, Objective & Scope

Problem Statement: -

To implement a machine learning based project where data is fetched from open.ai based api along with dall-E to generate user based input images.

The problem statement for a text-to-image generator is to develop a machine learning model that can take textual input as a description and generate a corresponding image that accurately represents the given text. This is a challenging task because it involves understanding and interpreting the semantics of natural language and converting it into a visual form. The model needs to capture the details of the text, including colors, shapes, and objects, and generate a realistic and coherent image. The generator should be able to create images with different styles and variations, based on the input description. The ultimate goal is to create a tool that can generate images automatically from textual input, which can be used in a variety of applications, such as virtual reality, gaming, advertising, and creative content generation.

The goal is to develop a system that can understand the meaning and context of the textual input and transform it into a visual form. This involves extracting key features, such as colors, shapes, and objects, from the text and translating them into a visual representation. The model should be able to generate images that are visually appealing, realistic, and coherent with the input text. Additionally, the generator should be able to produce images with different styles and variations, based on the input description, to increase its versatility and usefulness in various applications. The ultimate objective of this project is to create a tool that can automatically generate images from textual input, which can be used in areas such as digital content creation, advertising, and virtual and augmented reality applications.

Objective: -

1. Develop a natural language processing (NLP) model that can analyze a text input and give output in the form of images.
2. Automate the image generation process: Eliminating the need for manual intervention and reducing the time and effort required to create images.
3. Enhance creativity and innovation: Generating unique and imaginative images that users may not have otherwise thought of.
4. To optimize the performance of the machine learning model and back-end server for fast and efficient image generation.

Scope: -

1. The scope of this project is to build an AI-powered text to image generator using React.js, Next.js, and the DALL-E dataset. The project will involve developing a web-based user interface for users to input text, which will then be used to generate corresponding images using a GAN-based machine learning model. The generated images will be displayed to the user in real-time on the web interface.
2. The project will also involve building a back-end server to handle the text-to-image generation requests. The server will be responsible for running the machine learning model and returning the generated images to the front-end interface. The machine learning model will be built using TensorFlow and trained on the DALL-E dataset.
3. The project will be developed using agile development methodologies, with regular iterations and feedback from stakeholders. The project team will consist of developers with expertise in React.js, Next.js and web development.

Chapter 4

Proposed System Architecture

Description about Proposed System:

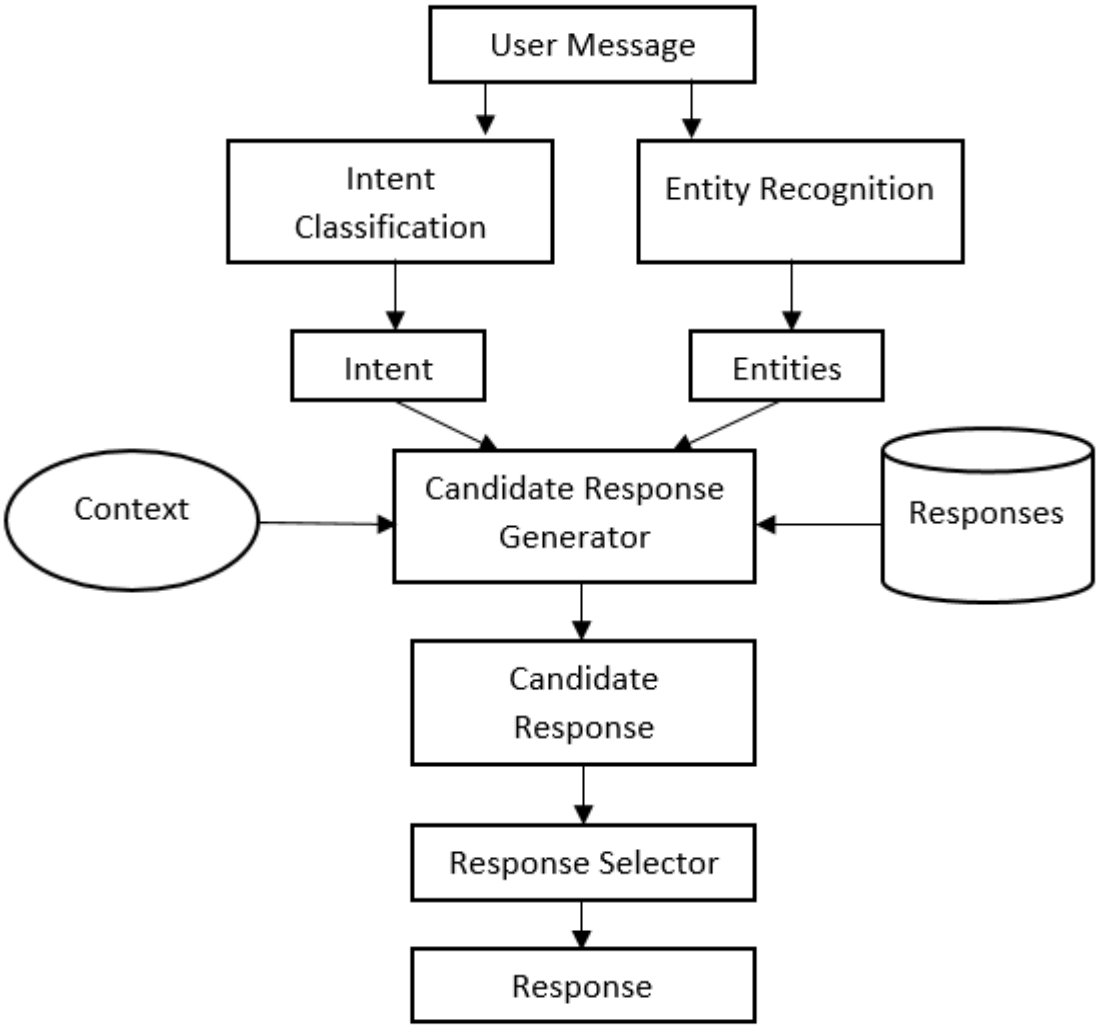
The proposed AI text to image generator system is a web-based application that enables users to generate high-quality images from textual descriptions. The system uses a GAN-based machine learning model that has been trained on the DALL-E dataset to generate the images.

The system consists of a user interface, a back-end server, and a machine learning model. The user interface is built using React.js and Next.js and enables users to input text descriptions. The back-end server is responsible for handling the text inputs and using the machine learning model to generate corresponding images. The machine learning model is built using TensorFlow and is trained on the DALL-E dataset to generate images that closely match the textual descriptions.

The system uses a GAN-based approach to generate the images, which involves training two neural networks: a generator network and a discriminator network. The generator network generates images from the textual descriptions, and the discriminator network evaluates the quality of the generated images. The networks are trained using a combination of adversarial and reconstruction loss functions to ensure the quality and accuracy of the generated images.

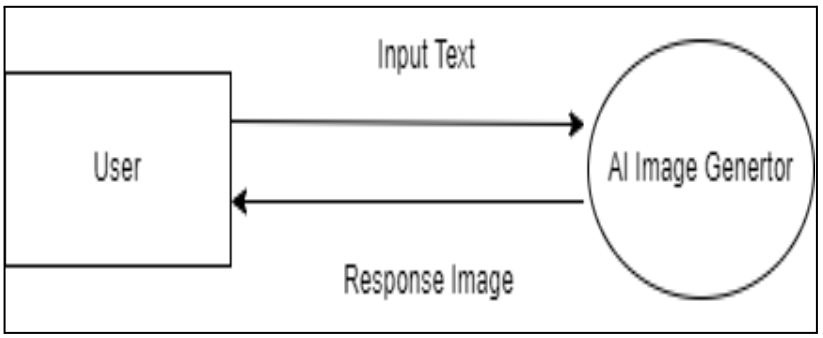
The system is designed to be user-friendly, with a simple and intuitive interface that enables users to easily input textual descriptions and view the corresponding generated images. The system is also scalable and efficient, with a back-end server that can handle multiple requests simultaneously and a machine learning model that is optimized for fast and efficient image generation.

Overall, the proposed AI text to image generator system provides a powerful and innovative solution for generating images from textual descriptions, with a high degree of accuracy and quality. The system has broad applications across a range of industries, including e-commerce, marketing, and design, and has the potential to transform the way


User Message
Intent Classification
Entity Recognition
Intent
Entities
Context
Candidate Response Generator
Responses
Candidate Response
Response Selector
Response

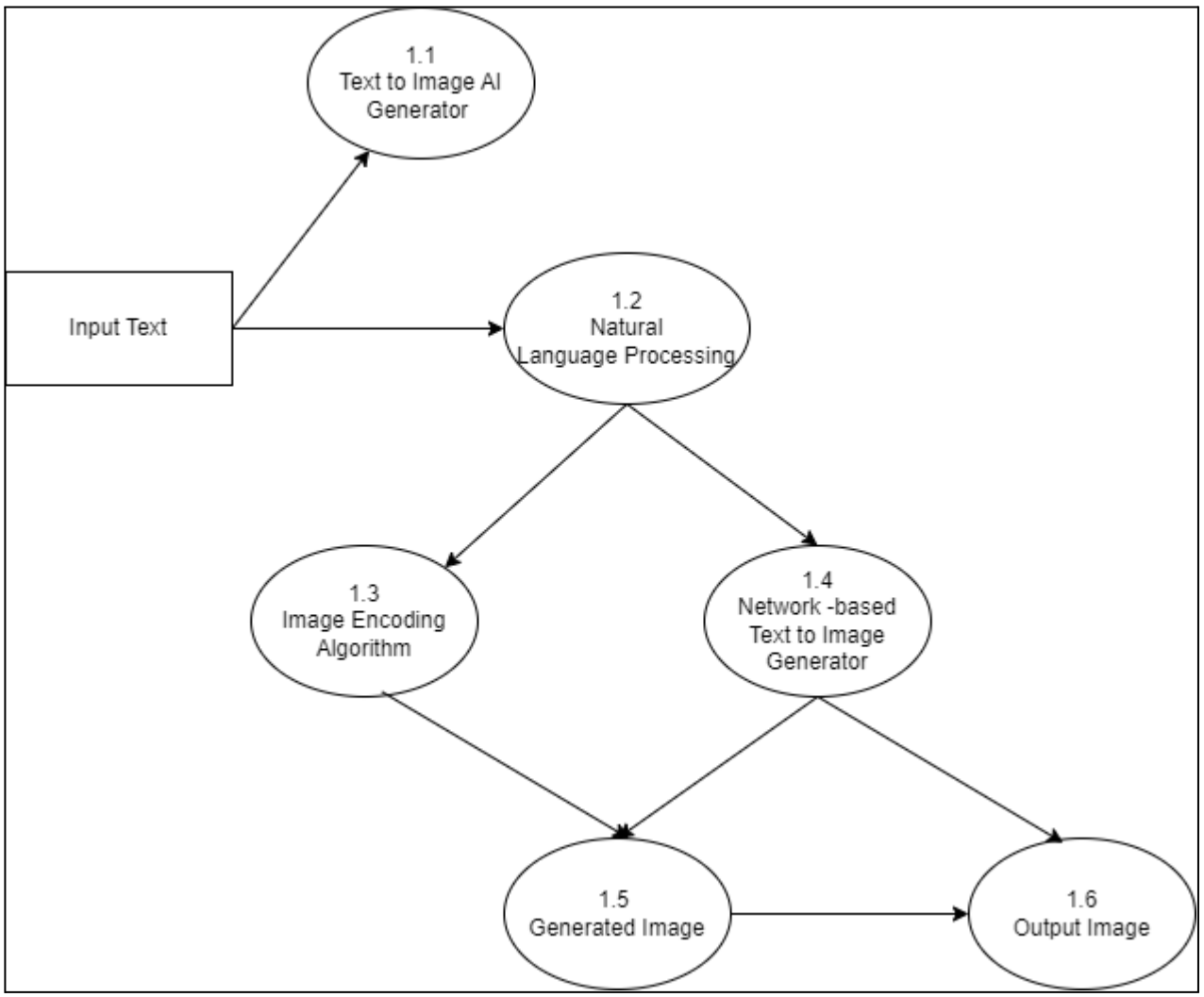
Data Flow Diagram

LEVEL 0



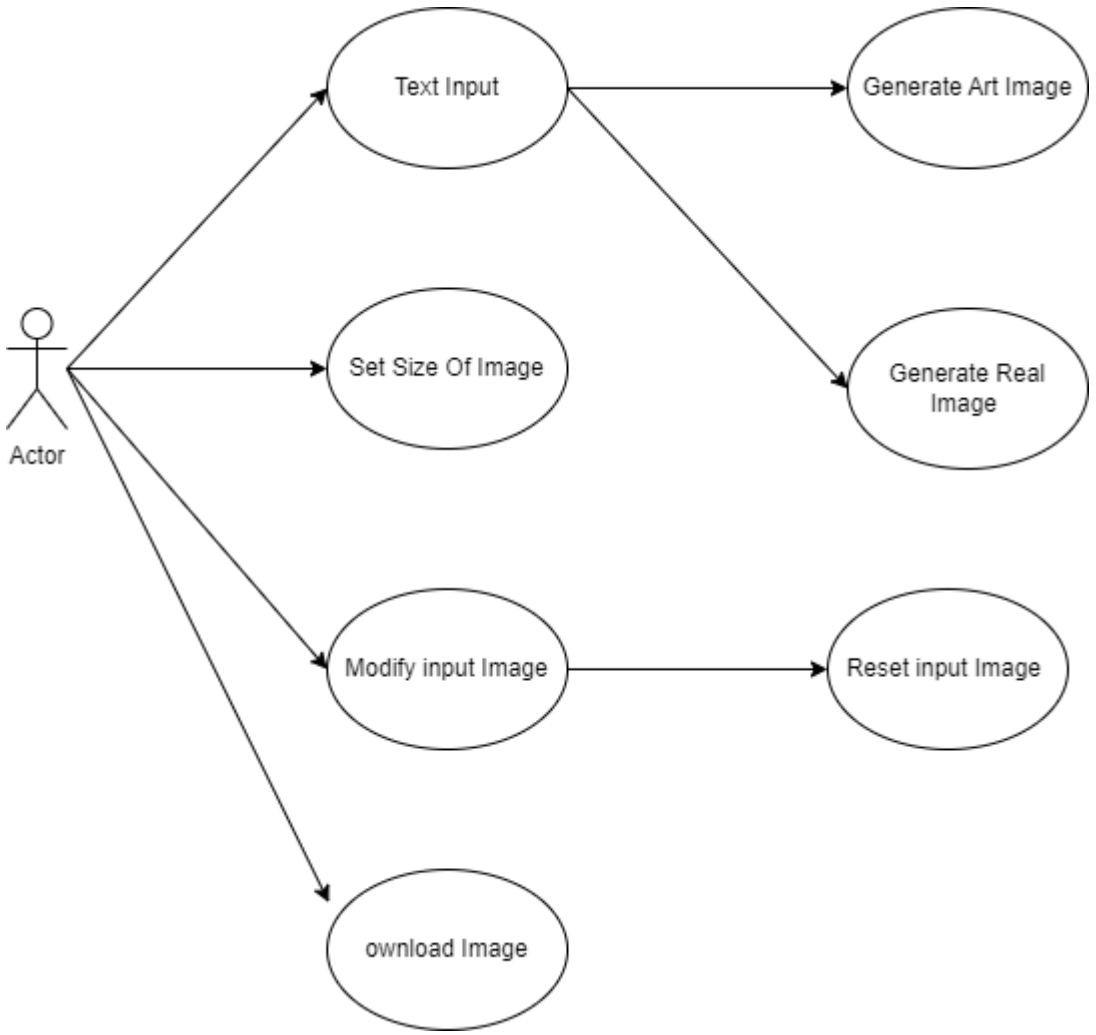
At level 0 of a Text to Image AI Generator, the data flow diagram provides an overview of the system components and the data flow between them. The level 0 diagram is a high-level representation that focuses on the main inputs, processes, and outputs of the system. It does not show detailed information about each component or data entity.

LEVEL 1



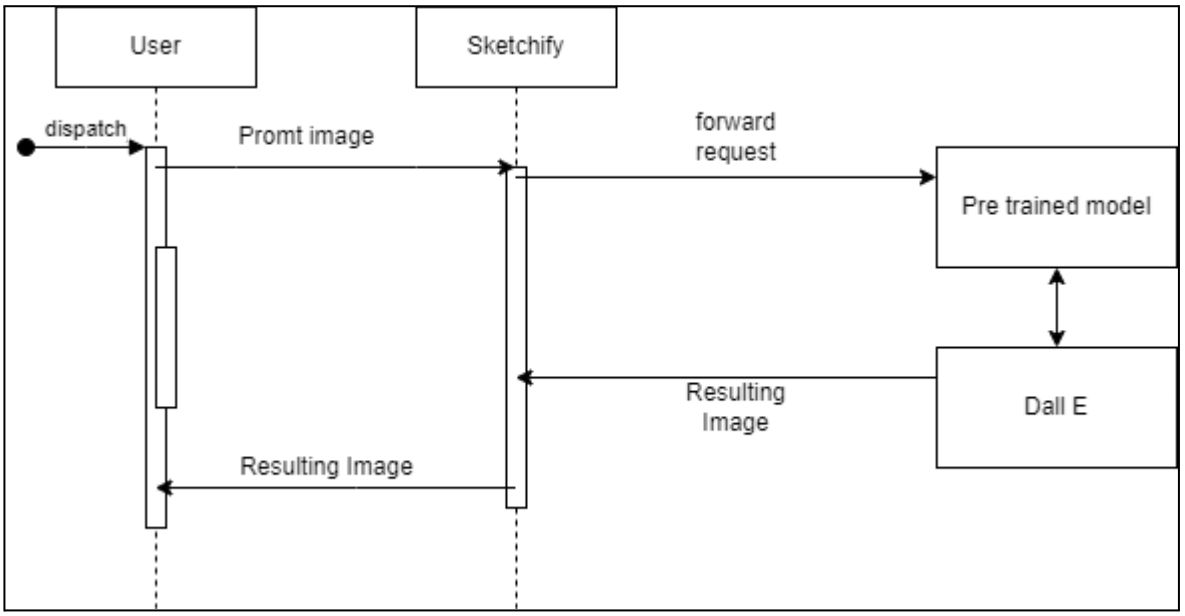
The diagram shows the different components and the flow of data in the system. The input is text, which is processed by natural language processing algorithms to extract the meaning and context of the text. The text is then fed into a neural network-based text-to-image generation algorithm, which uses this information to generate an image. The generated image data is then outputted as the final result.This diagram shows a high-level view of the data flow in a Text to Image AI Generator.

Use Case Diagram:



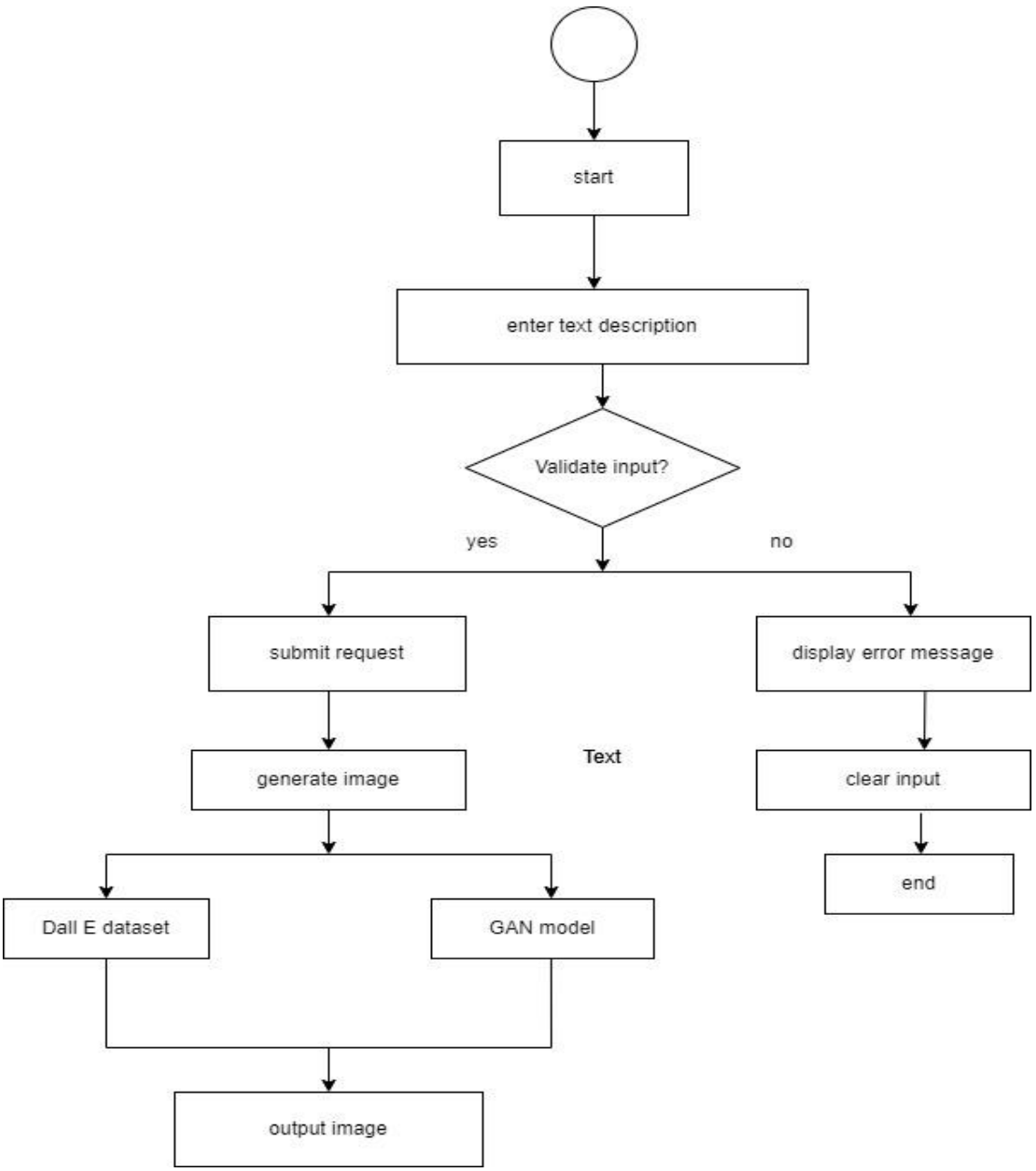
The main representation of system/software requirements for a new, developing software programme is a use case diagram. Use cases describe expected behaviour, not the precise way it will be accomplished (how). Once established, use cases can be represented visually or textually. The relationship between the use cases is known as the uses relationship when one use case is shown to be utilising the functionality of another use case. The cliche ">" refers to an extended relationship.

Sequence Diagram:



Sequence Diagrams are interaction diagrams that describe the steps used to complete an operation. They depict how items interact within the framework of a cooperation. Sequence Diagrams are time-focused, and they use the vertical axis of the diagram to represent the time at which messages are sent, thereby visually illustrating the order of the interaction

Activity Diagram :



Another crucial behavioral diagram in the UML used to depict the system's dynamic elements is the activity diagram.

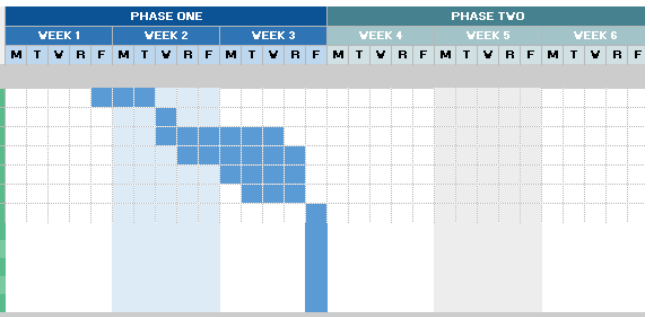
Chapter 5

Project Planning

Gantt Chart:

PROJECT TITLE	SKETCHIFY	COMPANY NAME	A.P Shah Institute Of Technology
PROJECT GUIDE	Prof. Sachin Takmare	DATE	1-15-23

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION(Weeks)	PCT OF TASK COMPLETE
1	**Project Conception and Initiation**					
1.1	Problem search	Atharva, Eeshan, Jay	15/1/23	9-2-23	3	100%
1.1.1	Problem finalization	Atharva, Eeshan, Jay	15/1/23	9-2-23	3	100%
1.2	Project Title	Atharva, Eeshan, Jay	15/1/23	9-2-23	3	100%
1.3	Abstract	Atharva	10-2-23	25/2/23	1	100%
1.4	Problem Definition	Atharva	10-2-23	25/2/23	1	100%
1.5	Objectives	Eeshan	10-2-23	25/2/23	1	100%
1.6	Scope	Eeshan, Atharva	3-3-23	20/3/23	1	100%
1.7	Existing System/Project	Jay	3-3-23	20/3/23	1	100%
1.8	Technology stack	Jay,Sahil	3-3-23	20/3/23	1	80%
1.9	Benefits for environment	Eeshan	3-3-23	26/3/23	1	100%
1.1	Benefits for society	Eeshan	3-3-23	26/3/23	1	80%
1.11	Applications	Eeshan,Sahil	3-3-23	26/3/23	1	100%
2	**Project Design**					
2.1	Proposed System	Atharva, Eeshan, Jay	1-4-23	9-4-23	1	70%
2.2	Design(Flow Of Modules)	Atharva, Eeshan	1-4-23	9-4-23	1	70%
2.3	Data Flow Diagram	Eeshan,Sahil	1-4-23	9-4-23	1	50%
2.4	Modules	Eeshan,Sahil	15/1/23	9-4-23	1	50%
2.4.1	Module-1	Atharva,Sahil	10-2-23	9-4-23	1	30%
2.4.2	Module-2		10-2-23	9-4-23	1	
2.4.3	Module-3		3-3-23	9-4-23	1	
2.4.4	Module-4		3-3-23	9-4-23	1	
2.5	Preparation Of Report		1-4-23	9-4-23	1	
3	**Project Implementation**					
3.1	Module-1	X			0	0%
3.2	Module-2	Y			0	0%
3.3	Module-3	Y			0	0%
3.4	Module-4	Z			0	0%
4	**Testing**					
4.1	Design of Test Cases	Y			0	0%
4.2	Testing	Y			0	0%
5	**Results and Analysis**					
5.1	Analysis Of Results	Y			0	0%
5.2	Graphical Representation	Y			0	0%
5.3	Report Preparation					



Chapter 6

Experimental Setup

Software Requirements: -

- Operating system: Windows, macOS, or Linux.
- Node.js runtime environment for running the JavaScript code (version 12 or higher).
- React.js and Next.js frameworks for the front-end development
- DALL-E dataset for image generation

Hardware Requirements: -

- CPU: Intel Core i5 or equivalent processor
- GPU: 4 GB VRAM
- RAM: 4 GB RAM
- STORAGE: Recommended to have at least 500 GB of available storage to accommodate the different components of your project.

Chapter 7

Implementation Details

- Backend API:
  The actual text-to-image generation is performed on the backend using a deep learning framework like TensorFlow or PyTorch. You can create a RESTful API using a web framework like Flask or FastAPI to receive text inputs from the frontend and return the generated images. The API should be designed to handle requests in parallel to maximize efficiency.

- Frontend Form:
  The frontend form should allow users to input text and submit it to the backend API. You can use a form library like Formik or react-hook-form to create a form with validation and error handling.

- Displaying Generated Images:
  Once the image is generated on the backend, it can be returned to the frontend and displayed to the user. You can use an image display library like react-image to display the generated image.

- Loading State:
  Since text-to-image generation can take a few seconds or longer, it is important to provide feedback to the user that the request is being processed. You can use a loading state library like react-loading to display a spinner or other loading animation while the image is being generated.

- Optimizing Image Size:
  Generated images can be large in size, and displaying them on the frontend can be slow. To optimize image size and loading times, you can use a library like next/image, which automatically optimizes images for different devices and screen sizes.

- Error Handling:
  Since text-to-image generation is a complex task, errors can occur on both the frontend and backend.

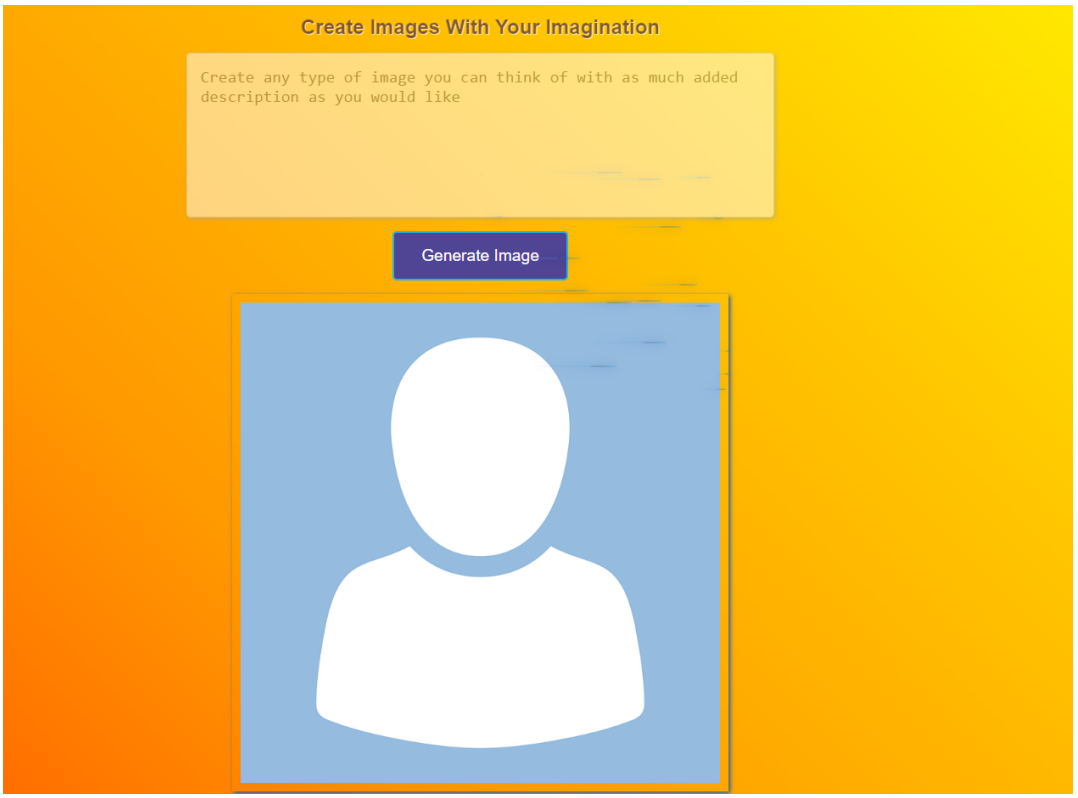
  You should handle errors gracefully and display helpful error messages to the user. You can use a library like react-toastify to display error messages in a user-friendly way.

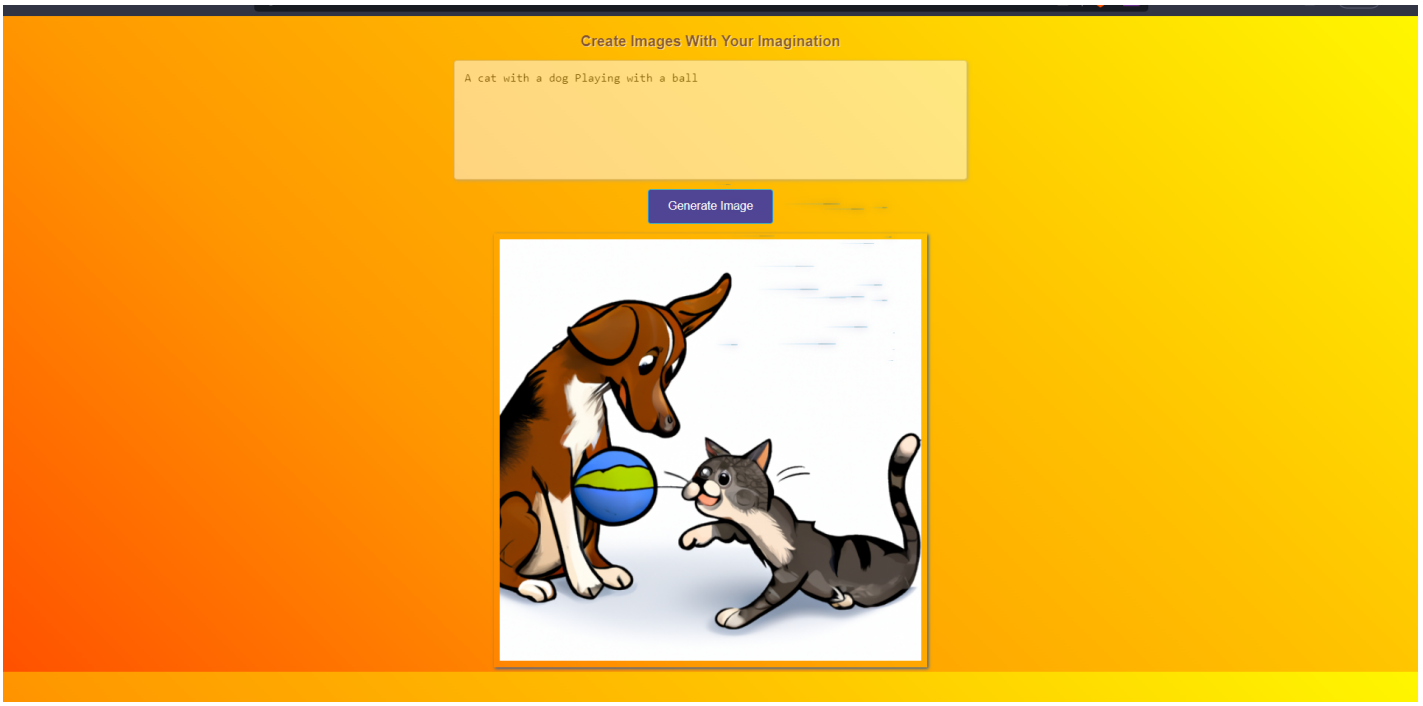
- Performance Optimization:
  Text-to-image generation can be a computationally intensive task, and you want to make sure that the application is performing optimally. To optimize performance, you can use techniques like code splitting and lazy loading to minimize the amount of code that needs to be loaded by the browser.

- User Interface:
  The user interface of the application should be designed to be user-friendly and intuitive. You can use a UI library like Material UI or Ant Design to create a consistent and visually .

Chapter 8

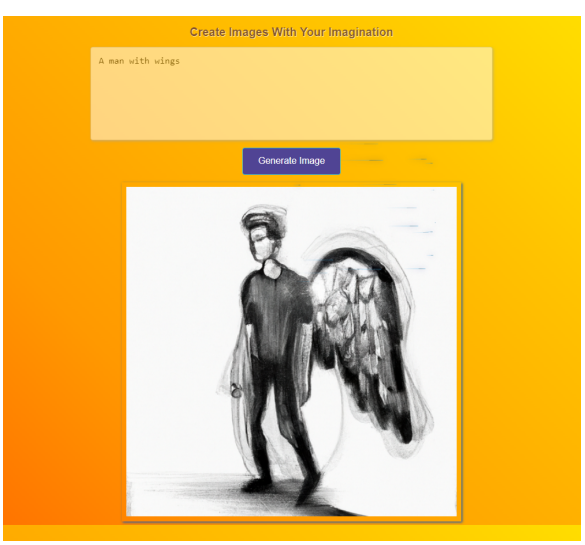
Result



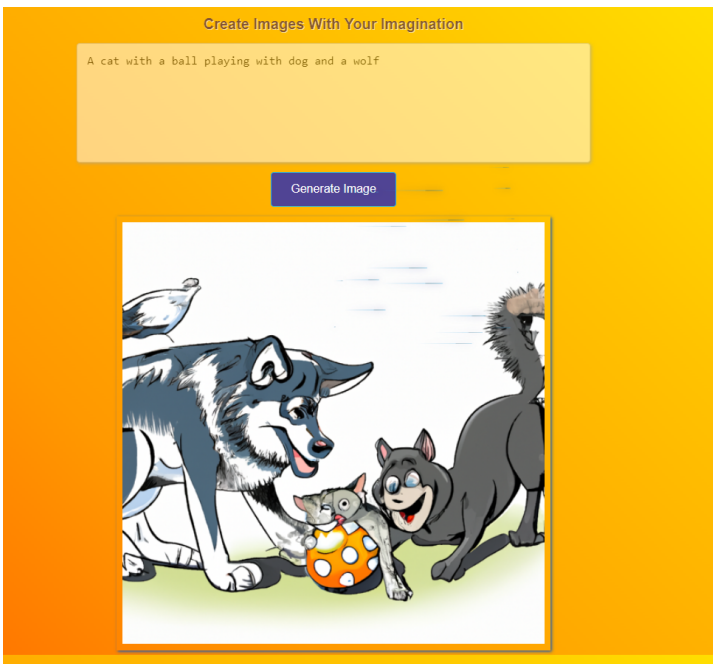


Implementation of Multi Image Collaboration

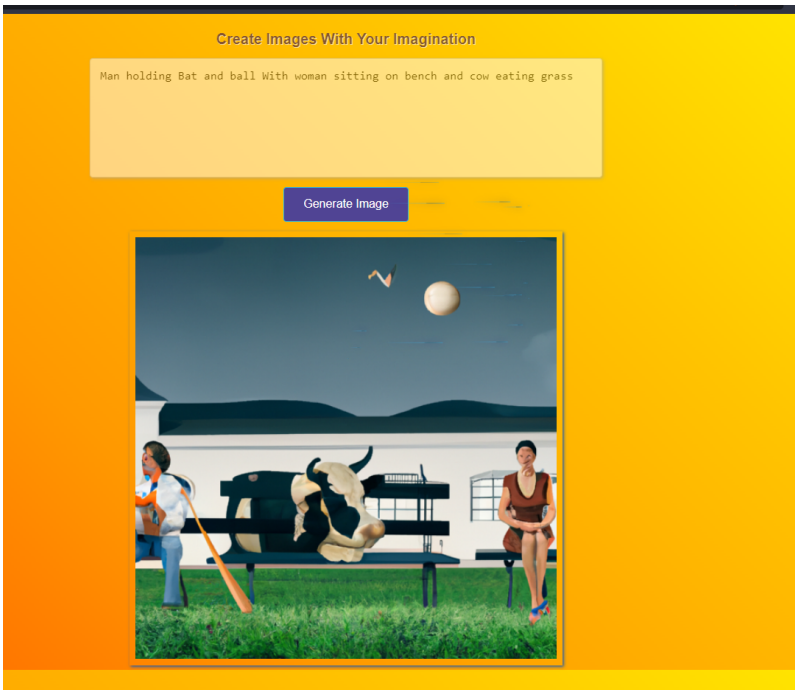
2 Objects:



4 objects:



8 objects:



Chapter 9

Conclusion

Sketchify: An AI-Based Image Generation Web Application:
Sketchify is an AI-based image generation web application that allows users to create custom images using artificial intelligence technology. The application is built using pre-trained models such as Dall-E, ReactJS, Tailwind, and other React elements to ensure that it is fast, responsive, and user-friendly. With Sketchify, users can create unique and visually stunning images with just a few clicks.Sketchify offers a wide range of features that make it stand out from other image generation web applications. Some of its notable .User-Friendly Interface: Sketchify's intuitive user interface makes it easy for users to create custom images quickly. The interface is designed to be user-friendly, allowing users to navigate the application effortlessly.Wide Range of Parameters: Sketchify allows users to choose from a wide range of parameters, including size, color, and content. Users can also specify the type of image they want to create, such as animals, landscapes, or abstract art.AI-Based Image Generation: Sketchify utilizes pre-trained models such as Dall-E to create unique and visually stunning images that are not only accurate but also aesthetically pleasing.Fast and Responsive: Sketchify is built using ReactJS, Tailwind, and other React elements to ensure that it is fast and responsive, providing users with a smooth and enjoyable experience.Customization: Sketchify allows users to customize their images by adjusting various parameters, such as brightness, contrast, and saturation. This ensures that users can create images that are tailored to their specific needs and preferences.In conclusion, the AI image generation web application developed using pre-trained models such as Dall-E, ReactJS, Tailwind, and other React elements, is an outstanding example of the power of modern technology. With its user-friendly interface, users can easily create customized images using a wide range of parameters, including size, color, and content.

Chapter 10

References

[1]Abdi, H., Valentin, D., & Edelman, B. (1999). Neural networks. Sage.Anadol, R. (2022). Space in the Mind of a Machine: Immersive Narratives.Architectural Design, 92(3), 28–37. https://doi.org/https://doi.org/10.1002/ad.2810

[2]Berg, N. (2022a). AI tools like DALL-E 2 and Midjourney are helpingarchitects—and their. RetrievedNovember9,2022,from https://www.fastcompany.com/90780871/ai-tools-like-dall-e-2-and-midjourney-arehelping-architects-and-their-clients-design-new-buildings

[3]Berg, N. (2022b). No. 38: The summer of AI: from fearful to fearless.Retrieved November 9, 2022, from https://futureyou.substack.com/p/no-38-thesummer-of-ai-from-fearful
Bernstein, P. (2022). Machine Learning. London: RIBA Publishing. https://doi.org/10.4324/9781003297192

[4]Brownlee, J. (2019). Deep learning for computer vision: image classification,object detection, and face recognition in python. Machine Learning Mastery.Campo, M. del, & Leach, N. (2022). Can Machines HallucinateArchitecture? Architectural Design, 92(1), 6–13.

[5]Dhariwal, P., & Nichol, A. (2021). Diffusion Models Beat GANs on ImageSynthesis. 35th Conference on Neural Information Processing Systems, 1–15.Florian, M. C. (2022a). Can Artificial Intelligence Systems like DALL-E or Midjourney Perform Creative Tasks? | ArchDaily. Retrieved November27,2022,from https://www.archdaily.com/987228/can-artificial-intelligencesystems-like-dall-e-or-midjourney-perform-creative-tasks

[6]Florian, M. C. (2022b). “This House Does Not Exist” Uses AI to Generate Images Inspired by ArchDaily’s Modern Architecture Projects | ArchDaily. Retrieved November 9, 2022, from https://www.archdaily.com/988606/thishouse-does-not-exist-uses-ai-to-generate-archdaily-style-images-of-modernarchitecture

[7]Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley,D., Ozair, S., … Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 3(January), 2672–2680. https://doi.org/10.3156/jsoft.29.5_177_2

[8]He, X., & Deng, L. (2017). Deep Learning for Image-to-Text Generation:A Technical Overview. IEEE Signal Processing Magazine, 34(6), 109–116.https://doi.org/10.1109/MSP.2017.2741510
TEXT-TO-IMAGE GENERATION A.I. IN ARCHITECTURE 119 How Google’s AlphaGo Beat Lee Sedol, a Go World Champion - TheAtlantic. (2016). Retrieved August 10, 2022, from https://www.theatlantic.com/technology/archive/2016/03/the-invisible-opponent/475611/

[9]Koh, I. (2020). Architectural Plasticity: The Aesthetics of Neural Sampling.Architectural Design, 92(3), 86–93.Liu, Z., Wang, Y., Qi, X., & Fu, C.-W. (2022). Towards Implicit TextGuided 3D Shape Generation.

[10]Proceedings of the IEEE/CVF Conference onComputer Vision and Pattern Recognition (CVPR), 17896–17906. Retrieved from https://github.com/liuzhengzhe/Towards-ImplicitNadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011,September).

[11] Natural language processing: An introduction. Journal of the American Medical Informatics Association, Vol. 18, pp. 544–551. https://doi.org/10.1136/amiajnl-2011-000464 Newell, A., & Simon, H. A. (1956). Current Developments in Complex Information Processing. California: Rand.

[12]Pieters, R., & Winiger, S. (2016). Creative AI: On the Democratisation & Escalation of Creativity | by creative.ai | Medium. Retrieved November 27, 2022, from https://medium.com/@creativeai/creativeai-9d4b2346faf3

[13]Shirai, K., & Fujisawa, H. (1974). An algorithm for spoken sentence recognition and its application to the speech input output system. IEEE Transactions on Systems, Man and Cybernetics, SMC 4(5), 475–479. https://doi.org/10.1109/tsmc.1974.4309346
Turing, A. M. (1950). Computing Machinery and Intelligence. Mind,59(236), 433–460.

[14]Wang, C., Chai, M., He, M., Chen, D., & Liao, J. (2022). CLIP-NeRF:Text-and-Image Driven Manipulation of Neural Radiance Fields. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 3835–3844.

[15]WDCH Dreams - Refik Anadol. (2019). Retrieved August 21, 2022, from https://refikanadol.com/works/wdch-dreams/Weizenbaum, J. (1966). ELIZA-A Computer Program for the Study ofNatural Language Communication Between Man and Machine. Communications of the ACM, 9(1), 36–45.

Acknowledgement

We have great pleasure in presenting the mini project report on “SKETCHIFY”. We take this opportunity to express our sincere thanks to our guide Prof. Sachin Takmare, Department of Computer Engineering, APSIT Thane for providing the technical guidelines and suggestions regarding a line of work. We would like to express our gratitude for her constant encouragement, support, and guidance throughout the development of the project. We thank Prof. Sachin Malave, Head of Department, and Prof. Deepak Khachane, Project Coordinator, Computer Engineering, APSIT for his encouragement during the progress meeting and for providing guidelines to write this report. We also thank the entire staff of APSIT for their invaluable help during this work. We wish to express our deep gratitude towards all our colleagues at APSIT for their encouragement.

Student Name 1: Sahil Limaye

Student ID: 20102115

Student Name 2: Atharva Karekar

Student ID: 20102117

Student Name 3: Eeshan Joshi

Student ID: 20102012

Student Name 4: Jay Mohitepatil

Student ID: 20102166