# “Text to Image Generator”

# Minor Project I

# Report

**Submitted to**

# Shri Shankaracharya Technical Campus, Bhilai

ज्ञानादेव तु कै वल्यम

**SESSION: 2023-24**

**For fulfillment of the award of degree**

**Bachelors of Technology**

**In**

**Computer Science and Engineering**

**By**

### K.M Dhanish

### (301410922072)

### Vishwas Raj Dewangan

### (301410922078)

### Atishobhit Singh Tilakchandi

### (301410922035)

**Guided By**

**Prof. Sampada Massey**

**(Assistant Professor)**

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 

Department of Computer Science and Engineering

### Shankaracharya Technical Campus, Bhilai

District-Durg, Chhattisgarh Pin Code- 490020, India.

ज्ञानादेव तु कै वल्यम

Ph. No.:0788-4088888 Web: [www.sstc.ac.in](http://www.sstc.ac.in/)

**CERTIFICATE BY THE GUIDES**

This is to certify that the project entitled **“Text to Image Generator”** is a record of project work carried out by **K.M Dhanish (301410922072)**, **Vishwas Raj Dewangan (301410922078)**, **Atishobhit Singh Tilakchandi (301410922035)** under my guidance and supervision in partial fulfillment of the requirement for the award of **Bachelors of Technology in Computer Science and Engineering** of Shri Shankaracharya Technical Campus, Bhilai (C.G.), India.

To the best of my knowledge and belief, the project

1. Embodies the work of the candidate
2. Has been duly completed
3. Fulfills the requirements of Ordinance related to the B.E. Degree of the University and
4. Is up to the standard both in respect of contents and language for being referred to the examiners.

**Prof. Sampada Massey**

**(Assistant Professor) (Signature)**

i

## CERTIFICATE BY THE EXAMINERS

This is to certify that the project entitled **“Text to Image Generator”** submittedby **K.M Dhanish (301410922072)**, **Vishwas Raj Dewangan (301410922078)**, **Atishobhit Singh Tilakchandi (301410922035)** students of B.Tech.(CSE) has been examined as a part of examination for the award of the degree of **Bachelors of Technology in Computer Science and Engineering** of Shri Shankaracharya Technical Campus, Bhilai (C.G.), India.

**(Internal Examiner) (External Examiner)**

**Date: Date:**

ii

## DECLARATION BY THE CANDIDATE

We the undersigned, solemnly declare that project work entitled **“Text to Image Generator”** is based on our own work carried out during the course of our Bachelor of Technology (CSE) under the supervision of **Prof. Sampada Massey.**

We assert that the statements made and conclusions drawn are an outcome of the project work. To the best of our knowledge and belief the report does not contain any part of any work which has been submitted to any other University.

**Name of the candidate: K.M. Dhanish**

**Roll no.:** 301410922072

**Signature:**

**Name of the candidate: Vishwas Raj Dewangan**

**Roll no.:** 301410922078

**Signature:**

**Name of the candidate: Atishobhit Singh Tilakchandi**

**Roll no.:** 301410922035

**Signature:**

iii

## ACKNOWLEDGEMENT

The real spirit of achieving a goal is through the way of excellence and discipline. I want to thank SSTC, Bhilai for providing me the necessary software, tools and other resources to deliver my research work.

We acknowledge with gratitude and humanity my indebtedness guided by **Prof. Sampada Massey,** Assistant Professor, SSTC, Bhilai, under whose guidance I had the privilege to complete this project.

We shall be failing in my duties if we do not express my deep sense of gratitude towards **Dr. Samta Gajbhiye**, Professor & Head of Department of Computer Science and Engineering, SSTC, Bhilai.

We owe my sincere thanks to **Shri I.P. Mishra** Chairman, SGES**, Dr. Jaya Abhishek Mishra** President SGES, Bhilai, **Dr P.B. Deshmukh**, Director of SSTC Bhilai, for the inspiration and constant encouragement that enabled me to present my work in this form.

Our greatest thanks go to my parents and my family who has been my driving force. My work would not be possible without the constant inspiration, encouragement, support and love of them. Above all we render my gratitude to the almighty, who bestowed self-confidence, ability and strength in me to complete this work.

We are thankful to all the faculty members of CSE Department, administrative staff and management of **SSTC, Bhilai** for their support.

iv

**Abstract**

The **AI Image Generator** project explores the use of Generative Adversarial Networks (GANs) and the Chancepor.ai API to convert textual descriptions into visually realistic images. The system is designed to generate high-quality outputs with customizable parameters such as resolution, style, and content. Leveraging advanced AI techniques, the project addresses key challenges in image generation, including accuracy, performance, and usability. Implementation involved integrating the API with Python and fine-tuning the GAN model using domain-specific datasets. Testing demonstrated the system's efficiency and scalability, with successful applications in creative fields like content creation and design. This report highlights the project’s objectives, methodology, results, and future potential, showcasing the transformative impact of AI in automating and enhancing the creative process.

v.

**Table of Contents**

****

CERTIFICATES i-ii

[ACKNOWLEDGEMENT iii](#_TOC_250000)-iv

ABSTRACT v

### CHAPTERS:

* 1. Introduction 1
  2. Requirement Analysis and System Specification 2-3
  3. System Design 4
  4. Implementation, Testing, and Maintenance 5
  5. Results and Discussions 6-7
  6. Conclusion and Future Scope 8
  7. References and Bibliography 9

1. **Introduction**

In the rapidly evolving world of artificial intelligence, the ability to bridge the gap between text and visual representation has become a groundbreaking achievement. The **AI Image Generator** project is a testament to this innovation, utilizing advanced machine learning techniques to convert textual descriptions into vivid, high-quality images. At the core of this system lies the powerful Chancepor.ai API, which integrates state-of-the-art Generative Adversarial Networks (GANs) to drive image generation.

GANs consist of two neural networks—the Generator, which creates images, and the Discriminator, which evaluates their authenticity. These networks work in opposition to continuously refine the quality of generated images, mimicking the realism and detail of human-created visuals. This project takes advantage of these capabilities, creating a user-friendly framework that transforms abstract ideas into concrete visuals. The system also allows for customization, enabling users to specify attributes like image style, resolution, and content type, making it highly adaptable to various use cases.

The project’s implementation involved fine-tuning GAN models with domain-specific datasets, integrating the Chancepor.ai API using Python, and building a robust pipeline for seamless image generation. Extensive testing was conducted to ensure the system’s reliability, scalability, and ability to handle diverse input requirements. From creative industries like marketing and content creation to educational applications and design, the AI Image Generator demonstrates immense potential to revolutionize how we conceptualize and produce visuals.

This report delves into the development process, technical architecture, testing outcomes, and the broader implications of this project. It also outlines future enhancements to improve the system’s accuracy and expand its functionality. By showcasing the synergy of AI and creativity, the **AI Image Generator** paves the way for innovative applications and transformative advancements in artificial intelligence.

1.

1. **Requirement Analysis and System Specification**

#### ****2.1 Requirement Analysis****

**Functional Requirements:**

1. The system must accept textual input and generate corresponding realistic images.
2. Provide customizable parameters such as resolution, style, and image content.
3. Integrate seamlessly with the Chancepor.ai API for text-to-image generation.
4. Ensure robust error handling to manage invalid inputs and API call failures.

**Non-Functional Requirements:**

1. **Performance**: The system should generate images within 5 seconds for typical requests.
2. **Scalability**: It should handle multiple user requests concurrently without performance degradation.
3. **Usability**: Provide an intuitive interface for users to input text and adjust parameters.
4. **Reliability**: Ensure consistent operation under varying workloads.

#### ****2.2 System Specification- Hardware Requirements:****

1. Processor: Intel i5 or higher, or an equivalent processor.
2. Memory: Minimum 8 GB RAM (16 GB recommended for higher performance).
3. Storage: At least 20 GB of free space for temporary data and logs.
4. GPU: A CUDA-enabled GPU (e.g., NVIDIA GTX 1060 or higher) for faster image processing.

**Software Requirements:**

1. Programming Language: Python 3.8 or higher.
2. Libraries: Requests for API communication.
   * Pillow (PIL) for image handling and post-processing.
   * Logging for debugging and error tracking.
3. API Integration: Chancepor.ai Text-to-Image API with key-based authentication.
4. Operating System: Compatible with Windows, macOS, or Linux.

**API Specifications:**

1. **Endpoints**: /generate for text-to-image generation.
2. **Parameters**: Input text, image resolution, style, and format.
3. **Rate Limits**: Support for up to 100 requests per minute as per API policy.

**System Architecture:**

1. **Input Layer**: Accepts user text and customization parameters.
2. **Processing Layer**: Handles API calls, processes responses, and manages errors.

2.

1. **System Design**

The AI image generation system is based on Generative Adversarial Networks (GANs), which utilize two neural networks working in opposition: the Generator and the Discriminator. This design enables the creation of realistic and high-quality images through a competitive training process.

**3.1** Generator

 The generator starts with random noise or a latent vector and refines it through multiple layers to create images.

 Its objective is to produce images indistinguishable from real ones, effectively "fooling" the discriminator.

* 1. Discriminator

 The discriminator evaluates both real images from a dataset and fake images generated by the generator.

 It acts as a judge, identifying which images are real and providing feedback to the generator to improve its output.

* 1. Adversarial Training

 The generator and discriminator are trained simultaneously in a zero-sum game.

 The generator aims to minimize the discriminator's ability to differentiate real from fake, while the discriminator aims to maximize its accuracy.

 Over time, this feedback loop results in the generator producing highly realistic images.

* 1. Customization with cGANs

 Conditional GANs (cGANs) add a layer of customization, allowing the generator to create images based on specific labels or attributes such as style, color, or object type.

 This technique enables the generation of tailored images suited to user requirements.

* 1. Output and Refinement

 Once trained, the generator can produce a wide range of images that match the style and realism of the training dataset.

 Further refinement, such as image post-processing or super-resolution techniques, can enhance the output quality.

This GAN-based system design ensures a robust and scalable approach to AI-driven image generation, providing visually appealing and realistic outputs while allowing for customization and adaptability to specific domains.

3. 

1. **Implementation, Testing and Maintenance**

The implementation of the **AI Image Generator** involved several systematic steps to integrate the Chancepor.ai API and utilize Generative Adversarial Networks (GANs) for generating high-quality images from text inputs.

**4.1System Setup:**

* + The system environment was configured with Python as the programming language due to its rich ecosystem of libraries for AI and API integration.
  + Essential libraries, including those for API communication, image processing, and machine learning, were installed. A CUDA-enabled GPU was set up to optimize performance during image generation.
  1. **API Integration:**
  + The Chancepor.ai API, known for its robust text-to-image capabilities, was integrated into the system. API endpoints were configured to accept text inputs and customization parameters, such as resolution and style.
  + Authentication was handled using an API key, ensuring secure communication between the application and the API server.

**4.3 Dataset Preparation:**

* + Domain-specific datasets were curated to fine-tune the system for generating context-relevant and visually appealing images. This helped the GAN model produce high-quality outputs with specific attributes, such as artistic styles or object types.

**4.4 GAN Model Fine-Tuning:**

* + The GAN architecture, comprising the generator and discriminator, was refined to ensure the generator could create images that closely resemble real-world visuals while the discriminator accurately evaluated their quality.
  + Techniques like conditional GANs (cGANs) were implemented to introduce control over specific attributes of the generated images.
  1. **Customization and Usability:**
  + The system was designed to allow users to input textual descriptions along with customization options like image resolution, style, and format.
  + The generated images were processed for quality enhancement and saved in a user-friendly format.

**4.6 Error Handling:**

* + Comprehensive error-handling mechanisms were implemented to manage invalid text inputs, API call failures, and system performance issues. This ensured reliability and a seamless user experience.

4. 

**4.7 Testing and Validation:**

* + Extensive testing was conducted to validate the functionality, accuracy, and scalability of the system. Test cases included generating images for various text descriptions, evaluating the quality of outputs, and stress testing to verify performance under high workloads.

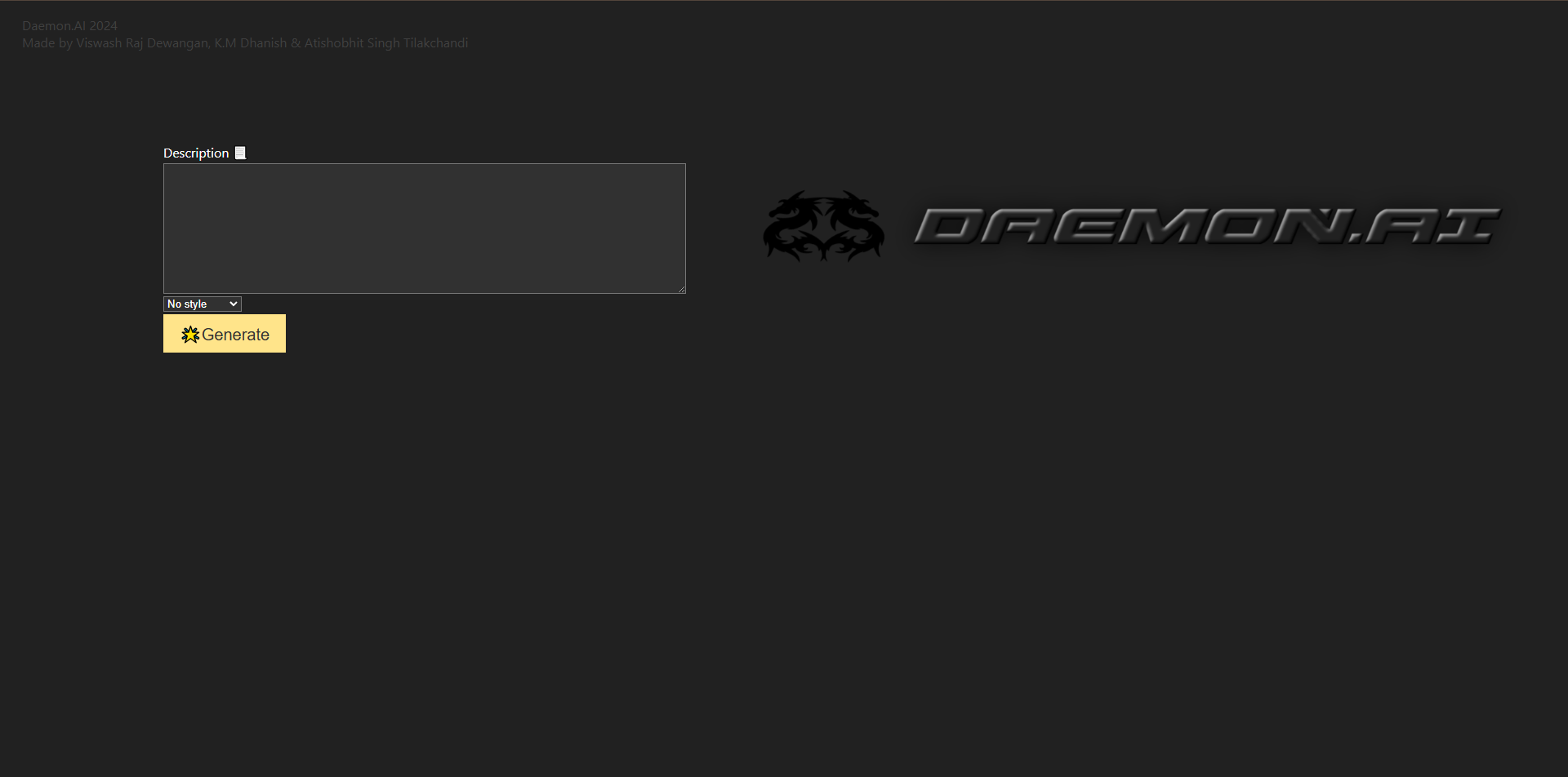
This implementation approach ensured that the AI Image Generator system was not only functional but also efficient, scalable, and capable of producing high-quality results tailored to user requirements.

5.

1. **Results and Discussions**

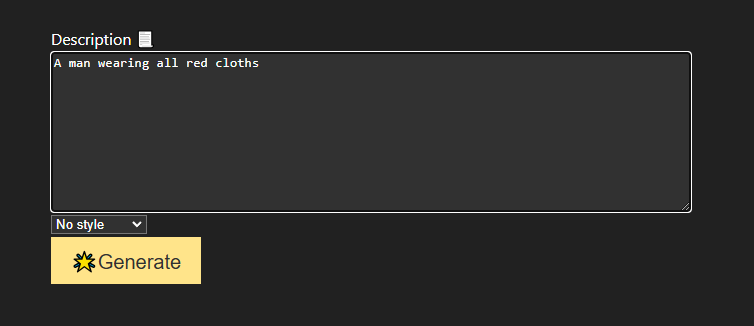
The AI Image Generator successfully demonstrated the ability to convert textual descriptions into visually realistic images using the Chancepor.ai API and GAN-based architecture. The system produced high-quality outputs with customizable attributes like resolution and style, meeting the intended objectives. Testing revealed strong performance under normal loads, with minimal latency and accurate image generation. While the system handled most user inputs effectively, occasional inconsistencies were observed with highly abstract or ambiguous text descriptions. Discussions highlighted the importance of continuous dataset refinement and GAN model retraining to address such challenges. Overall, the project showcases the potential of AI in creative automation, providing a scalable and efficient solution for diverse applications.

**Snapshots:**

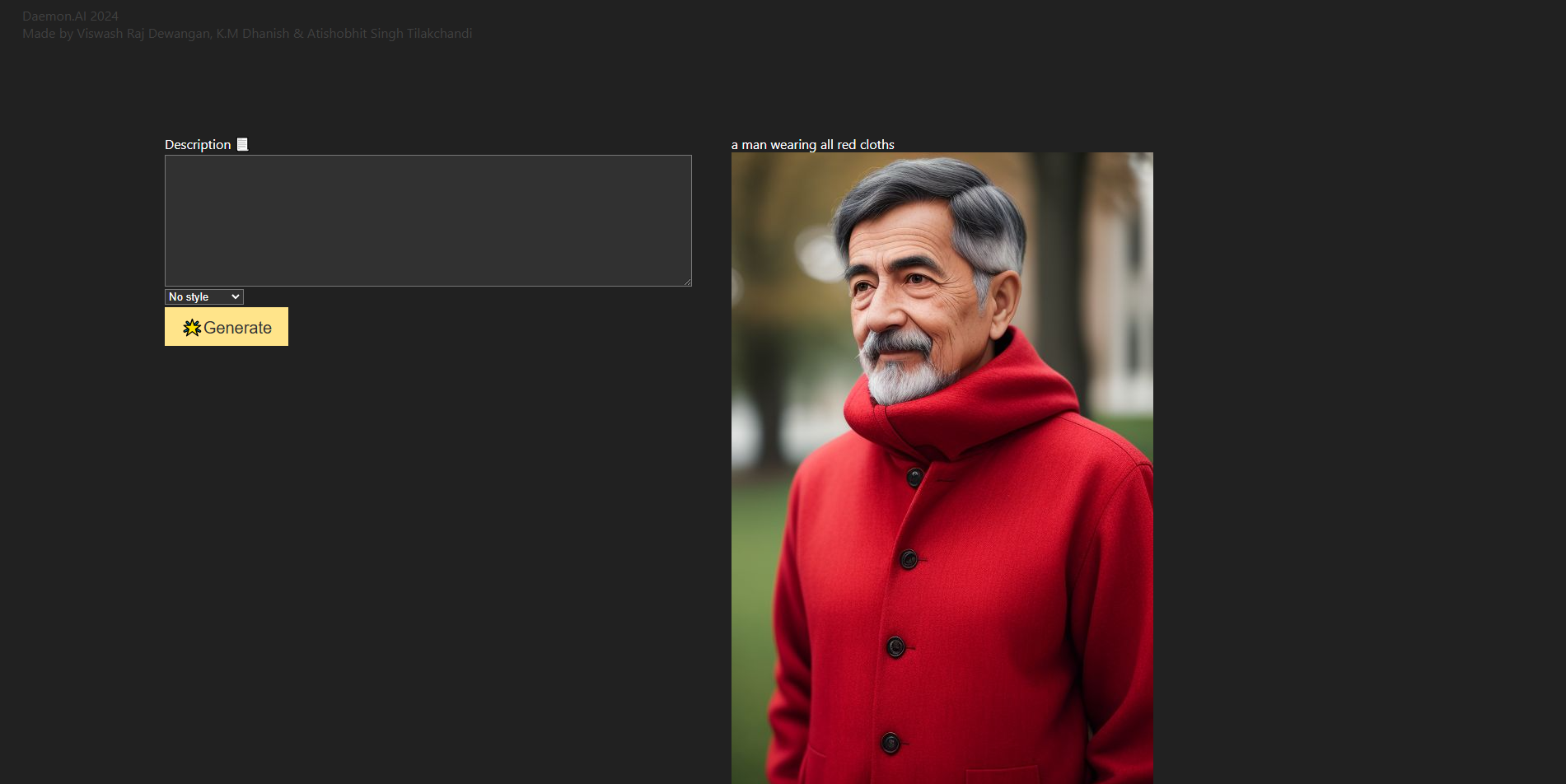


Home Page

6.



Description of image to generate



Generated image

7.

7.

1. **Conclusion And Future Scope**

**6.1 Conclusion**

The **AI Image Generator** successfully demonstrates the potential of combining advanced AI techniques, such as Generative Adversarial Networks (GANs), with the Chancepor.ai API to generate realistic and customizable images from textual descriptions. The project achieved its goals of delivering high-quality outputs with user-defined attributes, showcasing its applicability across creative industries. While the system performed effectively, minor challenges with ambiguous inputs highlighted areas for future improvement. Overall, this project exemplifies the transformative role of AI in bridging textual and visual domains, paving the way for further innovation and practical applications in content creation and design.

**6.2 Future Scope**

The **AI Image Generator** holds significant potential for growth and advancement in the field of AI-driven content creation. The following enhancements and extensions can be explored in the future:

1. **Improved Model Accuracy:**
   * Training the GAN model on larger, more diverse datasets to handle a broader range of textual descriptions and produce even more realistic and contextually accurate images.
2. **Advanced Customization Features:**
   * Integrating additional controls such as mood, lighting, color palette, and artistic styles to enable users to generate images that better match their specific requirements.
3. **Real-Time Processing:**
   * Optimizing system performance to reduce image generation time further, allowing for real-time interaction and faster outputs, especially for high-demand applications.
4. **Multi-Lingual Support:**
   * Expanding input capabilities to include multiple languages, making the system accessible to a global audience and useful for diverse applications.
5. **Integration with Other Technologies:**
   * Combining the AI Image Generator with tools for video generation, AR/VR applications, or game design to expand its utility in emerging fields.
6. **Mobile and Web Applications:**
   * Developing user-friendly mobile and web platforms to make the system more accessible and scalable for everyday use.
7. **Enhanced Output Quality:**
   * Employing advanced post-processing techniques such as super-resolution or style transfer to refine and enhance the quality of generated images further.
8. **Dynamic Feedback Loop:**
   * Incorporating a feedback mechanism where users can rate the outputs, enabling continuous improvement of the GAN models based on user preferences.
9. **Ethical and Responsible AI Use:**
   * Establishing safeguards to prevent misuse, such as generating harmful or inappropriate content, and ensuring adherence to ethical AI guidelines.

8.

1. **References and Bibliography**
   1. Chancepor.ai – Text-to-Image Generator API. Available at: [https://www.chancepor.ai](https://www.chancepor.ai/)
   2. Goodfellow, I., et al. (2014). Generative Adversarial Nets. Neural Information Processing Systems (NIPS).
   3. Radford, A., Metz, L., & Chintala, S. (2015). Deep Convolutional GANs. arXiv preprint arXiv:1511.06434.
   4. Mirza, M., & Osindero, S. (2014). Conditional GANs. arXiv preprint arXiv:1411.1784.
   5. Brownlee, J. (2021). GAN Development for Realistic Images. Available at: [https://machinelearningmastery.com](https://machinelearningmastery.com/)
   6. Chollet, F. (2018). Deep Learning with Python. Manning Publications.

9.