**SYNOPSIS**

**Report on**

**PicStream**

**by**

RUPESH GUPTA: 2300290140150

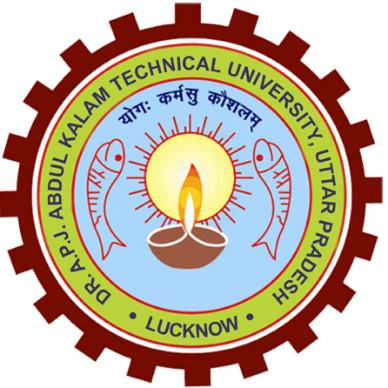
ROHIT KUMAR: 2300290140148

**Session: 2024-25 (III Semester)**

Under the supervision of

**Prof. DR. VIPIN KUMAR <<Associate Professor >>**

### KIET Group of Institutions, Delhi-NCR, Ghaziabad



### Department Of Computer Applications

**KIET GROUP OF INSTITUTIONS, DELHI-NCR, GHAZIABAD-201206**

(2024-25)

**ABSTRACT**

This project implements an AI-powered image generator using the MERN (MongoDB, Express.js, React.js, Node.js) stack, integrating OpenAI's ChatGPT API to create dynamic image prompts. The system allows users to input descriptions, which are processed through the ChatGPT API to generate creative prompts for image generation. The top three AI-generated images, based on specified size criteria, are stored in a MongoDB database alongside the search prompt and other metadata.

The front-end, built with React.js, enables users to input a description, which is sent to the Express.js and Node.js back-end. The back-end calls the ChatGPT API to generate a text prompt, and an AI image generation API is then used to create images. After fetching multiple images, the top three images are selected based on predefined size specifications and stored in MongoDB with the original user input prompt, timestamps, and other relevant data.

This project demonstrates the integration of AI-driven creativity with a modern full-stack web application, leveraging the MERN stack to manage user input, data storage, and real-time interactions. It provides a seamless user experience while ensuring efficient handling and storage of AI-generated content.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| 1 | Introduction | 4 |
| 2 | Literature Review | 5 |
| 3 | Objective | 6 |
| 4 | Technologies Used | 7 |
| 5 | Data Flow Diagram | 9 |
| 6 | E-R Diagram | 10 |
| 7 | Conclusion | 11 |
| 8 | Reference’s | 12 |

|  |  |  |
| --- | --- | --- |
| S. No | Topic | Page No. |

INTRODUCTION

In recent years, the rapid advancement of artificial intelligence has enabled innovative ways to enhance user creativity, especially in the realm of digital art and design. AI-driven image generation is one such area where users can input simple text descriptions to generate visually stunning and unique images. This project leverages this AI capability by integrating OpenAI's ChatGPT API within a MERN (MongoDB, Express.js, React.js, Node.js) stack to build a dynamic AI image generator.

The primary goal of this project is to create an interactive web application where users can input a text description and receive a set of AI-generated images. The system generates creative image prompts using ChatGPT, which are then passed to an AI image generation API. The top three images, based on predefined size specifications, are saved into a MongoDB database along with the user's input prompt and other metadata.

By utilizing the MERN stack, the project ensures smooth communication between the front-end and back-end. React.js handles the user interface, allowing for an intuitive and responsive user experience. Express.js and Node.js power the server-side operations, managing the API calls, processing data, and interacting with MongoDB to store the generated content.

LITERATURE REVIEW

The integration of AI in creative processes, particularly image generation, has gained significant traction with the advent of models like Generative Adversarial Networks (GANs) and diffusion models. These innovations, such as OpenAI's \*DALL-E\*, enable AI to generate images from textual descriptions, pushing the boundaries of machine-generated art. AI-generated image systems rely on understanding language and visual translation, a process enhanced by transformer models like OpenAI’s ChatGPT, which generates rich and complex prompts for image creation.

ChatGPT, built on the GPT-3 and GPT-4 architecture, excels at generating detailed and imaginative prompts from user input, as demonstrated by research on zero-shot learning. By chaining language models with image generation systems, multi-step AI processing improves output quality. This project leverages ChatGPT for prompt generation to enhance the accuracy and creativity of AI-generated images.

The project employs the MERN (MongoDB, Express.js, React.js, Node.js) stack for full-stack development. MongoDB’s scalability is ideal for storing AI-generated images and metadata, while Express.js and Node.js provide efficient back-end operations. React.js ensures a responsive and interactive front-end user experience. Integrating AI into full-stack applications, using modular approaches and REST APIs, allows for seamless communication between the front-end, back-end, and AI services, supporting scalability and flexibility in web applications.

In summary, this project combines advancements in AI-driven image generation and full-stack web development to create a dynamic, interactive application, offering users a creative, AI-enhanced platform for generating and storing digital art.

RESEARCH OBJECTIVE

The primary objective of this research is to develop an advanced full-stack web application that seamlessly integrates AI-driven image generation using the MERN (MongoDB, Express.js, React.js, Node.js) development stack. Specifically, the project aims to:

1. **Integrate OpenAI's ChatGPT API** to dynamically generate creative and descriptive text prompts from user input, which will serve as the basis for generating unique digital images.
2. **Leverage an AI-based image generation API** to fetch and render a collection of images based on the prompts, ensuring the top three images—selected according to predefined size and quality criteria—are stored for further use.
3. **Implement a MongoDB database** to efficiently store the generated images, along with associated metadata such as the user’s search prompt, timestamps, and image properties. This will ensure scalability, quick retrieval, and proper organization of data.
4. **Create an interactive and responsive user interface** using React.js to allow users to easily input prompts and view generated images in real-time, ensuring a seamless and intuitive user experience.
5. **Enhance back-end processing** with Node.js and Express.js to efficiently handle API calls, manage data flow between the AI systems and the database, and facilitate real-time interactions between the front-end and back-end components.

In addition, the project aims to evaluate the potential of combining cutting-edge AI technologies with full-stack web development frameworks to offer users a novel, AI-enhanced platform for creative tasks. It will examine how well AI can be used to automate the creative process, how efficient and scalable the system is, and how effectively AI-generated content can be managed within a modern web application infrastructure. Through this research, the project will contribute to the growing intersection of AI and web development, showcasing the practical applications of AI in digital art and interactive media platforms.

TECHNOLOGIES USED

**Front-end:**

1. **React.js**: A JavaScript library used for building the user interface. React enables the creation of a dynamic, responsive, and interactive front-end that allows users to input prompts, view AI-generated images, and interact with the system in real-time.
2. **CSS (Cascading Style Sheets)**: Used for styling the user interface components, ensuring a visually appealing and user-friendly design. CSS enhances the responsiveness and layout of the web application.
3. **Axios**: A promise-based HTTP client used in React to make asynchronous API requests to the back-end, such as sending user input to generate prompts and retrieving the top AI-generated images.
4. **JavaScript/JSX**: Used to manage the logic and structure of the React components, providing the interactivity and flow of the application.

**Back-end:**

1. **Node.js**: A JavaScript runtime environment used for building the server-side logic of the application. Node.js powers the back-end, handling API requests and enabling efficient data processing between the front-end, ChatGPT API, and image generation service.
2. **Express.js**: A minimal and flexible Node.js web application framework used for creating the server’s API endpoints. Express is responsible for handling routing, API calls, and interactions between the front-end, ChatGPT, image generation API, and MongoDB database.
3. **RESTful API**: The architectural style used for structuring the communication between the front-end, back-end, and external AI services (ChatGPT and image generation APIs). REST APIs ensure that data requests and responses are efficiently handled.
4. **ChatGPT API (OpenAI)**: A language model API used to generate detailed prompts based on user input. The back-end communicates with ChatGPT to produce creative text prompts, which are then passed to the image generation API.
5. **AI Image Generation API**: An external service used to generate images from the prompts created by ChatGPT. The back-end fetches multiple images from this API, selects the top three based on size criteria, and returns them to the front-end while storing them in the database.

**Database:**

1. **MongoDB**: A NoSQL database used for storing AI-generated images, user inputs, and associated metadata. MongoDB is chosen for its flexibility in handling large and varied data structures, as well as its scalability for future growth. The database efficiently stores and retrieves images, along with information like the search prompts and timestamps.