Yusuf_Enes_Aras.docx

by Yusuf Aras

Submission date: 17-Apr-2024 07:56AM (UTC+0100)
Submission ID: 229648349
File name: Yusuf_Enes_Aras.docx (7.59M)
Word count: 651
Character count: 4447

YUSUF_A AI Image Generator:

Name: Yusuf Aras
NO: 97394922
Module: Full Stack Application Development 23-24

Introduction:

AI image generation is an innovative field where artificial intelligence models are used to create images based on textual descriptions. Inspired by the advancements in AI and the potential for creative applications, the Yusuf_A AI Image Generator was developed. This project draws inspiration from pioneering tools like Midjourney and OpenAI's DALL-E, which have demonstrated the capability of generating high-quality images from textual prompts.

Project Goals:

The primary objective of the project is to build a user-friendly image generation tool utilizing the MERN stack (MongoDB, Express.js, React.js, Node.js) along with OpenAI's DALL-E model. The aim is to empower users to effortlessly generate images from textual descriptions, facilitating creative expression and exploration.

Technologies Used:

- MERN Stack: Utilized for full-stack web development, enabling seamless interaction between frontend and backend components.
- Tailwind CSS: Employed for efficient and responsive UI design, facilitating rapid development.
- OpenAI's DALL-E model: Integrated to generate images based on textual descriptions, leveraging state-of-the-art AI capabilities.
- Cloudinary: Utilized for image storage and management, ensuring scalability and efficient handling of media assets.

Development Process:

- Frontend Development: Developed the user interface using React.js and enhanced styling with Tailwind CSS to ensure a modern and intuitive user experience.

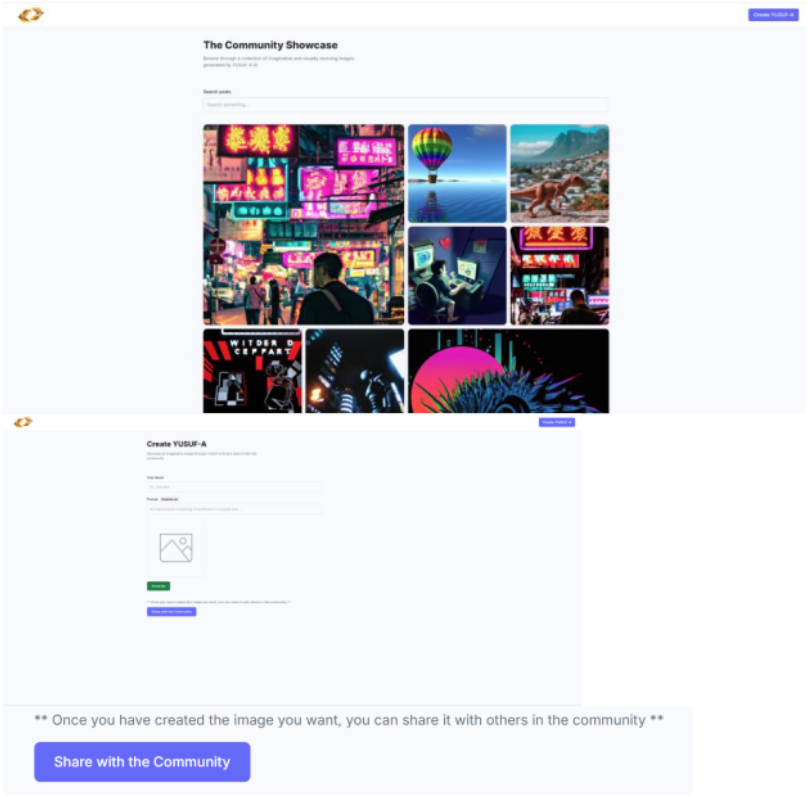
- Backend Development: Implemented backend functionality using Node.js and Express.js to handle API requests and orchestrate interactions with the AI model.
- AI Integration: Integrated OpenAI's DALL-E model into the backend to generate images based on user input.
- Testing and Deployment: Conducted thorough testing to ensure the functionality and reliability of the application. Deployed the application using a chosen deployment platform, ensuring accessibility to users.

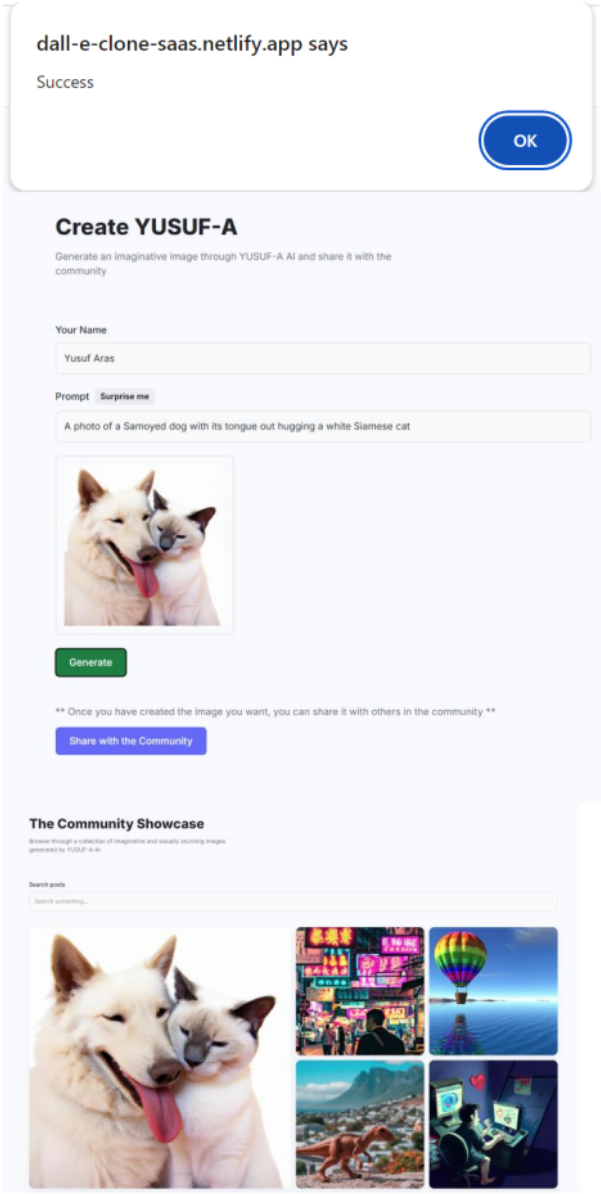
Testing and Deployment:

- Functionality Testing: Comprehensive testing was conducted to validate the functionality of the image generation tool, ensuring accurate rendering of images from textual descriptions.
- Deployment: The application was deployed using a cloud-based platform, ensuring scalability and availability to users.

Results and Discussion:

- User Interface:




dall-e-clone-saas.netlify.app says
Success
OK
Create YUSUF-A
Generate an imaginative image through YUSUF-A AI and share it with the community
Your Name
Yusuf Aras
Prompt
Surprise me
A photo of a Samoyed dog with its tongue out hugging a white Siamese cat
Generate
** Once you have created the image you want, you can share it with others in the community **
Share with the Community
The Community Showcase
Search posts

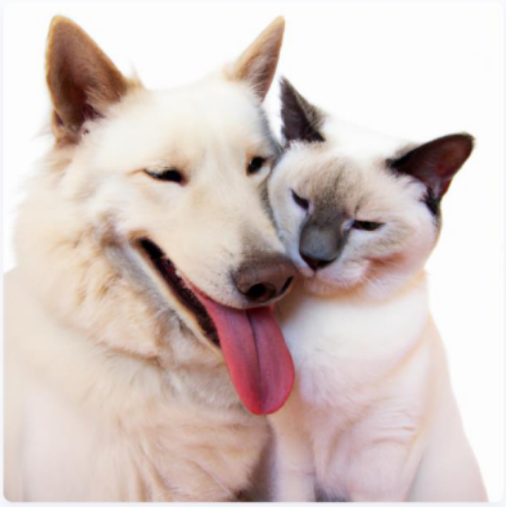
The Community Showcase

Browse through a collection of imaginative and visually stunning images generated by YUSUF-A AI

Search posts

Samoyed dog with its tongue

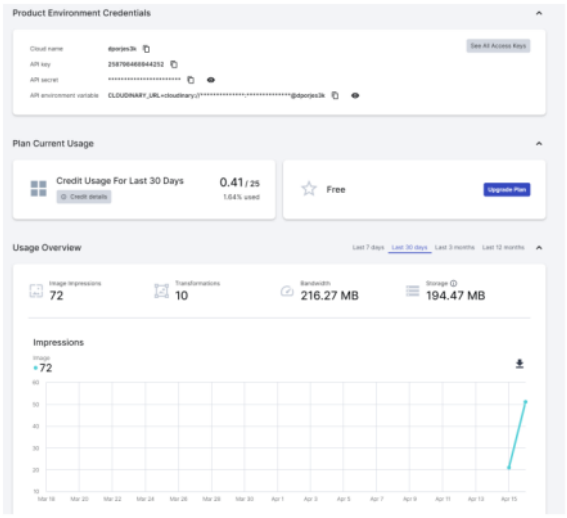
Showing Resuls for **Samoyed dog with its tongue**:

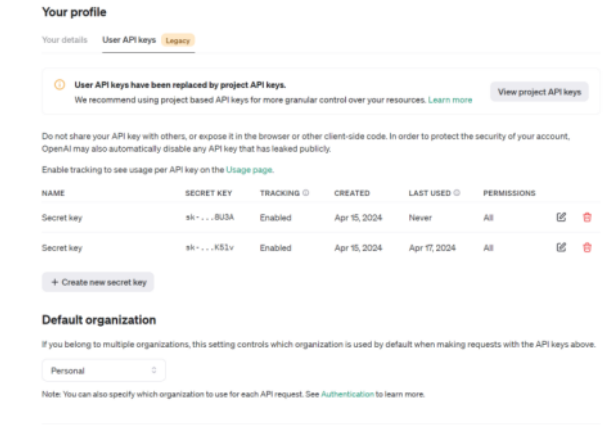


test

posts

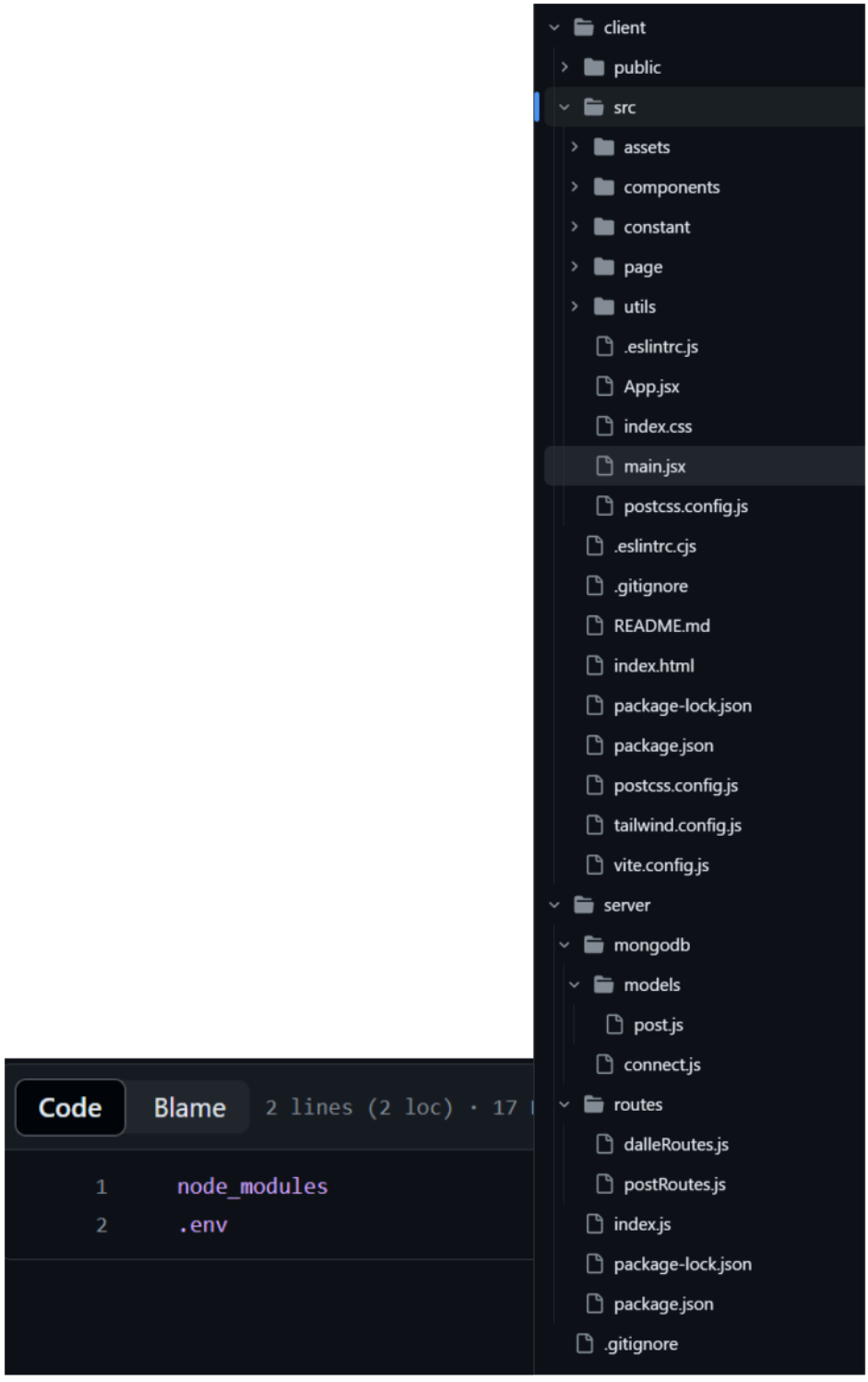
test.posts


Product Environment Credentials
Cloud name
API key
API secret
API environment variable
See All Access Keys
Plan Current Usage
Credit Usage For Last 30 Days
0.41 / 25
1.64% used
Credit details
Free
Upgrade Plan
Usage Overview
Last 7 days
Last 30 days
Last 3 months
Last 12 months
Image Impressions
72
Transformations
10
Bandwidth
216.27 MB
Storage
194.47 MB
Impressions
72


Your profile
Your details
User API keys
Legacy
User API keys have been replaced by project API keys.
We recommend using project based API keys for more granular control over your resources. Learn more
View project API keys
Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that has leaked publicly.
Enable tracking to see usage per API key on the Usage page.
NAME
SECRET KEY
TRACKING
CREATED
LAST USED
PERMISSIONS
Secret key
sk-...BU3A
Enabled
Apr 15, 2024
Never
All
Secret key
sk-...K51v
Enabled
Apr 15, 2024
Apr 17, 2024
All
+ Create new secret key
Default organization
If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.
Personal
Note: You can also specify which organization to use for each API request. See Authentication to learn more.


Usage
$0.54


client
server
.gitignore


client
public
src
assets
components
constant
page
utils
.eslintrc.js
App.jsx
index.css
main.jsx
postcss.config.js
.eslintrc.cjs
.gitignore
README.md
index.html
package-lock.json
package.json
postcss.config.js
tailwind.config.js
vite.config.js
server
mongodb
models
post.js
connect.js
routes
dalleRoutes.js
postRoutes.js
index.js
package-lock.json
package.json
.gitignore
Code
Blame
2 lines (2 loc) · 17
1 node_modules
2 .env

- Project Success: The project successfully achieved its goals of developing a user-friendly image generation tool. Users can effortlessly create images from textual descriptions, fostering creativity and exploration.
- https://dall-e-clone-saas.netlify.app/ (try now)

Challenges:

- Integration Complexity: Integrating the DALL-E model into the backend posed a significant challenge due to its complexity and the need for efficient communication between the frontend and backend.
  Solution: To address this challenge, we carefully studied the documentation provided by OpenAI and leveraged community resources and forums for guidance. By breaking down the integration process into smaller tasks and collaborating with team members, we were able to successfully integrate the model and establish seamless communication between frontend and backend components.
- Performance Optimization: Another challenge was optimizing the performance of the application, especially when handling multiple image generation requests simultaneously, to ensure a smooth user experience.
  Solution: We conducted thorough performance testing and profiling to identify bottlenecks and areas for optimization. By implementing caching mechanisms, optimizing API endpoints, and fine-tuning server configurations, we were able to significantly improve the application's performance and responsiveness.
- Deployment Scalability: Deploying the application on a scalable platform while ensuring consistent performance and reliability posed a challenge, especially considering the computational requirements of the DALL-E model.
  Solution: We opted for a cloud-based deployment platform that offered scalable infrastructure and resources. By leveraging containerization technologies like Docker and orchestrating deployment with Kubernetes, we achieved horizontal scalability and efficient resource utilization, ensuring optimal performance under varying workloads.
-

Conclusion:

The Yusuf_A AI Image Generator project provided valuable insights into the integration of AI technologies into web applications. Key learnings include effective utilization of MERN stack components, seamless integration of AI models, and responsive UI design. Future improvements may include enhanced AI capabilities, additional features for image customization, and integration with social media platforms.

References:

- https://github.com/yusufaras104/dall.e “code and assets”
- https://react.dev/
- https://platform.openai.com/docs/api-reference
- https://tailwindcss.com/
- https://cloudinary.com/
- https://nodejs.org/docs/latest/api/
- https://www.mongodb.com/docs/
- https://expressjs.com/
- https://docs.netlify.com/

Yusuf_Enes_Aras.docx

ORIGINALITY REPORT

7%	3%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of College Cork Student Paper	2%
2	Submitted to University of Florida Student Paper	2%
3	jobs.vault.com Internet Source	2%
4	"Diffusions in Architecture", Wiley, 2024 Publication	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Yusuf_Enes_Aras.docx

GRADEMARK REPORT

FINAL GRADE

72/100

GENERAL COMMENTS

The report is well-written, but there is a noticeable presence of AI-generated content. Additionally, the report lacks essential system requirements and fails to adequately caption and explain images, which could hinder understanding.

Feedback on the presentation:

The website presented is intriguing and demonstrates a considerable level of complexity. Effective utilization of HTML, CSS, and React is evident in the frontend design. However, it is noted that the authentication functionality is not fully operational. While APIs are employed for AI integration, there is a notable absence of Python code implementation and the use of Python frameworks, as expected. This shortfall may impact the project's ability to fully leverage AI capabilities.

Despite the impressive presentation of the website and the proficient use of frontend technologies, the incomplete authentication and AI integration aspects suggest areas for improvement in functionality and integration.

Project Presentation Score: 78

Report Score: 54

Overall Score: 72