សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ
ROYAL UNIVERSITY OF PHNOM PENH

ប្រធានបទនិងរបាយការណ៍ស្រាវជ្រាវ

Implementation of an Image Editor Platform By Using Cloudinary AI

A Final Report
In Partial Fulfilment of the Requirement for the Degree of
Bachelor of Engineering in Information Technology Engineering

ធី ចំរើនពិសិដ្ឋ

Thy Chamroeunpiseth

June 2025

សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ
ROYAL UNIVERSITY OF PHNOM PENH

ប្រធានបទនិងរបាយការណ៍ស្រាវជ្រាវ

Implementation of an Image Editor Platform By Using Cloudinary AI

A Final Report
In Partial Fulfilment of the Requirement for the Degree of
Bachelor of Engineering in Information Technology Engineering

ធី ចំរើនពិសិដ្ឋ

Thy Chamroeunpiseth

Examination committee: Mrs. Soy Sambocheyear
Mr. Sao Kimsong
Mr. Bun Leap
Mr. Yeu Ratha

មិថុនា ២០២៥

មូលន័យសង្ខេប

ប្រធានបទនិងរបាយការណ៍ស្រាវជ្រាវនេះចង់បង្ហាញពីកម្មវិធីកែរូបភាព (Artify) ដែលជាកម្មវិធី ឬ គេហទំព័រដែលផ្អែកលើការប្រើប្រាស់បញ្ញាសិប្បនិម្មិត (AI) ដើម្បីផ្តល់ឱ្យនូវឧបករណ៍ដែលមាន អានុភាព និង វិចារណញ្ញាណ សម្រាប់ការកែលម្អ និង ធ្វើការផ្លាស់ប្តូររូបភាព ។ គេហទំព័រនេះរចនាឡើងសម្រាប់ទាំងអ្នកប្រើប្រាស់ទាំងពីរប្រភេទ គឺអ្នកប្រើប្រាស់ដោយឥតគិតថ្លៃ និង អ្នកប្រើប្រាស់ដោយផ្អែកលើការទិញ ឬ ជាវជាប្រចាំខែ។ គេហទំព័រ នេះផ្តល់ជូននូវមុខងារជាច្រើនដែលជំរុញដោយ AI សម្រាប់ការកាត់ផ្ទៃផ្នែកខាងក្រោយនៃរូបភាព ការលុបវត្ថុរបស់ចេញពីក្នុងរូបភាព ការកែពណ៌វត្ថុនៅក្នុងរូបភាព ការពង្រីកបន្ថែមផ្ទៃរូបភាពដោយប្រើប្រាស់ AI និង ការស្តារគុណភាពរូបភាពដែលបាត់បង់គុណភាពឱ្យត្រឡប់មកវិញ។ សមត្ថភាពទាំងនេះត្រូវបានប្រើ តាមរយៈ APIs របស់ Cloudinary ដែលអនុញ្ញាតឱ្យដំណើរការរូបភាពប្រកបដោយគុណភាព ហើយមានល្បឿនលឿន និង ពេលវេលាជាក់ស្តែង។

ស្ថាបត្យកម្មប្រព័ន្ធធ្វើតាមការរចនាម៉ូឌុល និង អាចធ្វើមាត្រដ្ឋានបានដោយបង្ហាញនូវផ្នែកខាងមុខ (Frontend) Next.js ផ្នែកខាងក្រោយ (Backend) Node.js និងមូលដ្ឋាន ទិន្នន័យ (Database) MySQL សម្រាប់ការផ្ទុកជាប់លាប់។ ហេដ្ឋារចនាសម្ព័ន្ធ webhook របស់ Paddle ត្រូវបានប្រើប្រាស់ដើម្បីគ្រប់គ្រងការចេញវិក្កយបត្រការជាវ និង ការគ្រប់គ្រងការចូលប្រើប្រាស់របស់អ្នកប្រើប្រាស់ដោយធានានូវការធ្វើបច្ចុប្បន្នភាព
តាមពេលវេលាជាក់ស្តែងចំពោះសិទ្ធិរបស់អ្នកប្រើប្រាស់។ អ្នកប្រើអាចដាក់បញ្ចូលរូបភាពដោយមានសុវត្ថិភាព និង អាចមើលឃើញពីការបម្លែងរូបភាពជាមុន មុនពេលទាញយករូបភាពមកប្រើប្រាស់ និង ទាញយកលទ្ធផលដោយផ្តល់នូវបទពិសោធន៍រលូន និងងាយស្រួលប្រើ។

ការធ្វើតេស្តបានបញ្ជាក់ពីការអនុវត្តដែលអាចទុកចិត្តបាននៅទូទាំងសេណារីយ៉ូជាច្រើន ជាមួយនឹងប្រតិបត្តិការមូលដ្ឋានទិន្នន័យមានស្ថេរភាព ការគ្រប់គ្រងការចូលប្រើបានត្រឹមត្រូវ និងការបំប្លែង AI ឆ្លើយតប។ វេទិកានេះក៏បានបង្ហាញពី សមត្ថភាពក្នុងការគ្រប់គ្រងការប្រើប្រាស់ដំណាលគ្នាកម្រិតមធ្យមដោយមិនមានការថយចុះនៃការអនុវត្ត។ គម្រោងក្នុងការកែលម្អបន្ថែមនាពេលខាងមុខរួមមាន ការបន្ថែមមុខងារកែសម្រួលវីដេអូ ពង្រីកសមត្ថភាព AI បន្ថែមដូចជា ការកាត់ផ្ទៃទំហំរូបភាព ការបង្កើន

ទំហំរូបភាព និងការបន្ថែមស្រមោលនៃរូបភាព។ ការអភិវឌ្ឍន៍សេវាកម្ម AI ផ្ទាល់ខ្លួន ដោយប្រើប្រាស់ MinIO ដើម្បី កាត់បន្ថយការពឹងផ្អែកលើ APIs ភាគីទីបី។ សរុបមក គម្រោងនេះបង្ហាញ ពីលទ្ធភាពនៃការផ្តល់នូវការកែសម្រួលរូបភាព AI ដែលមានមូលដ្ឋានលើ cloud-based ដែលអាចធ្វើមាត្រដ្ឋានបានតាមរយៈការជាវដើម្បីប្រើប្រាស់ កម្មវិធីគេហទំព័រ។

ABSTRACT

This report presents the Image Editor Platform (Artify), a web-based application that leverages artificial intelligence to deliver powerful and intuitive tools for image enhancement and transformation. Designed for both free and subscription-based users, the platform offers a range of AI-driven features, including background removal, object removal, object recoloring, generative fill, and image restoration. These capabilities are enabled through Cloudinary's advanced transformation APIs, which support real-time, high-speed, and high-quality image processing.

The system architecture follows a modular and scalable design, featuring a Next.js frontend, Node.js backend, and MySQL database for persistent storage. Paddle's webhook infrastructure is integrated to manage subscription billing and access control, ensuring real-time updates to user privileges. Users can securely upload images, preview transformations live, and download the results, offering a smooth and user-friendly experience.

Testing confirmed reliable performance across multiple scenarios, with stable database operations, accurate access control, and responsive AI transformations. The platform also demonstrated the ability to handle moderate concurrent usage without performance degradation. Planned improvements include support for video editing, enhanced AI tools such as cropping, upscaling, and shadow effects, and the development of a custom AI service using MinIO to reduce dependency on third-party APIs. Overall, this project demonstrates the feasibility of delivering scalable, cloud-based AI image editing through a subscription-enabled web application.

SUPERVISOR'S RESEARCH SUPERVISION STATEMENT

TO WHOM IT MAY CONCERN

Name of program: Bachelor of Engineering in Information Technology Engineering
Name of candidate: Thy Chamroeunpiseth

Title of research report: Implementation of an Image Editing Platform by using Cloudinary AI

This is to certify that the research carried out for the above titled Bachelor's research report was completed by the above-named candidate under my direct supervision. This thesis material has not been used for any other degree. I played the following part in the preparation of this research report:

Supervisor's name: Chhim Bunchhun

Supervisor's signature: ……

Date: ……

CANDIDATE'S STATEMENT

TO WHOM IT MAY CONCERN

This is to certify that the research report that I (Thy Chamroeunpiseth) hereby present entitled Implementation of an Image Editor Platform By Using Cloudinary AI.

for the degree of Bachelor of Engineering at the Royal University of Phnom Penh is entirely my own work and, furthermore, that it has not been used to fulfill the requirements of any other qualification in whole or in part, at this or any other University or equivalent institution.

No reference to, or quotation from, this document may be made without the written approval of the author.

Signed by (Thy Chamroeunpiseth): ……

Date: ……

Sign by Supervisor: Chhim Bunchhun

Supervisor's signature: ……

Date: ……

ACKNOWLEDGEMENTS

To begin, I would like to express my sincere appreciation to everyone who contributed to the success of this project.

First and foremost, I extend my deepest gratitude to my advisor, **Mr. Chhim Bunchhun**, for his guidance, encouragement, and thoughtful feedback throughout the development of this report. His expertise and mentorship were essential in helping me refine my ideas and overcome technical challenges.

I would also like to thank the Department of Information Technology Engineering at the Royal University of Phnom Penh for providing a supportive academic environment and access to the resources necessary for this project. Special thanks go to my lecturers and peers whose insights and discussions helped shape the technical direction of the platform.

Lastly, I am thankful to my family and friends for their continuous motivation, patience, and emotional support. Their belief in my abilities kept me focused and driven. I also acknowledge the developers behind Cloudinary and Paddle for their powerful tools and documentation, which played a crucial role in bringing the Image Editor Platform (Artify) to life.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background and Motivation to the Study

Visual content is becoming a key part of communication, marketing, design, and social media interaction in the digital world we live in. As more and more photographs are made and used, there is a growing need for image editing tools that are quick, easy to use, and smart. In the past, manipulating images to a high standard needed complex tools like Adobe Photoshop and a degree of skill that many people and small organizations may not have. People who want rapid and polished visual material can't get in since they need professional tools and abilities.

Adding artificial intelligence (AI) to image editing has opened up a whole new world of possibilities for making complicated editing tools easier to use. With little input from the user, AI-powered applications may automate difficult tasks like removing backgrounds, finding objects, changing the color of objects, and generative fill. This cuts down on the time, money, and technical know-how needed to get professional-level outcomes.

Cloudinary is one of the best cloud-based media management platforms. It has a lot of AI-powered tools that make smart and automatic image editing available to both developers and end users. Its robust APIs and transformation capabilities make it easy to add complex editing features to both online and mobile apps. This means you don't have to design and manage your own image processing infrastructure.

The need for smart, easy-to-use image-enhancing tools is growing, which is why this project, Implementation of an Image Editing System by Using Cloudinary AI, is happening. It is all about building a platform that employs Cloudinary's AI features to help with basic tasks like removing backgrounds (Background Remove), removing objects (Object Remove), changing the color of objects (Object Recolor), filling in gaps (Generative Fill), and restoring images (Image Restore). The system's goal is to make image editing easier by focusing on three important use cases.

The suggested solution uses cloud-based AI to make the editing process more efficient and make sure that it may grow and be used by everyone. It gives people and companies the capacity to easily maintain and improve their visual content without needing additional software or technical knowledge.

1.2 Problem Statements

1.2.1 Complexity of Traditional Tools

Conventional image editing software such as Adobe Photoshop or GIMP is often complex and requires significant time and training to master. These platforms assume a level of design proficiency, which many casual users or small businesses lack. This creates a barrier to entry for non-technical users and slows down content workflows.

1.2.2 Resource Constraints

Licensing fees for high-end editing software and the cost of hiring professional designers can be prohibitive, especially for startups, freelancers, and small businesses. As a result, many are unable to produce or enhance visual content at a competitive level.

1.2.3 Scalability Issues

Manual editing processes are labor-intensive and do not scale well for businesses dealing with high volumes of visual assets, such as e-commerce platforms or content agencies. Automating repetitive tasks like background removal or object cleanup becomes essential to maintaining efficiency.

1.2.4 Limited Accessibility and Usability

Many available editing solutions are not designed with usability in mind, making them inaccessible to users without technical backgrounds. This lack of inclusivity limits the reach and adoption of existing platforms, particularly in emerging markets.

1.2.5 Lack of Real-Time Editing Capabilities

Several desktop-based editors do not support real-time editing, collaboration, or instant preview, which are increasingly demanded in modern workflows. Delayed feedback cycles lead to inefficiencies and hinder productivity.

1.2.6 Limited AI-Driven Editing Options in Legacy Tools

While AI technology is rapidly evolving, many traditional editors have not yet integrated features like generative fill, automatic object recoloring, or intelligent image

restoration. This limits the creative potential and speed at which content can be developed.

1.3 Aim and Objectives of the Study

The aim of this report is to develop an Image Editor Platform (Artify) that leverages Cloudinary's capabilities to provide accessible, efficient, and high-quality image processing tools for users of all skill levels. The specific objectives are:

- To implement core AI features including background removal, object removal, object recoloring, generative fill, and image restoration.
- To design a secure credit-based payment system for managing access to premium editing functionalities.
- To create a user-friendly and responsive interface suitable for both novice and experienced users.
- To ensure the platform is scalable and optimized for handling large volumes of image processing tasks efficiently.
- To enable cross-platform accessibility, allowing users to interact with the system on both web and mobile devices.
- To integrate secure user authentication and safeguard personal data in compliance with privacy standards.
- To include analytics and reporting features to monitor system usage and support ongoing improvements.
- To build the system with a modular architecture, facilitating the integration of additional AI features in the future.

1.4 Rationale of the Study

In today's digital world, where visuals are so important, being able to make high-quality, interesting images is important in many areas, including e-commerce, marketing, content creation, and personal branding. But traditional image editing programs like Adobe Photoshop or GIMP can be hard to learn and take a lot of time, and not everyone can afford them. These barriers make it harder for people and small businesses to make and improve visual content quickly and easily.

As artificial intelligence (AI) gets better, it becomes possible to change the way we edit images by making complicated tasks easier and more automatic. AI models can

do things like remove backgrounds, find objects, recolor things, and fill in holes with content-aware fill with very little help from the user. This cuts down on the time and skill needed to get professional results by a lot.

Cloudinary is a powerful cloud-based media management platform that has a set of AI-powered image transformation tools that developers can use in their apps without having to build complicated infrastructure from scratch. The proposed Image Editor Platform (Artify) can offer fast, scalable, and easy-to-use editing tools to a wide range of people by using Cloudinary's APIs and pre-trained AI models.

This study is based on the need to make advanced image editing available to everyone by creating a system that is simple to use, affordable, and useful for both people and businesses. The goal of the implementation is to fix problems with image editing accessibility, performance, and scalability. It does this by offering a practical and creative solution that meets the needs of modern digital content.

1.5 Limitations and Scopes

1.5.1 Limitation

While the proposed Image Editor Platform (Artify) offers significant advantages through AI integration, it is subject to certain limitations that may affect its overall performance and user experience:

- The platform’s effectiveness is heavily dependent on the performance of underlying AI models. Results may vary depending on image complexity, resolution, and subject matter.
- Reliance on third-party services such as Cloudinary-AI for image processing and Paddle for subscription management introduces external dependencies. Any downtime, API changes, or limitations from these services can impact system reliability.
- The use of a subscription-based model for accessing premium features may limit accessibility for users who are unable or unwilling to pay for the service.

1.5.2 Scope

The project is designed to deliver a practical and scalable solution for intelligent image editing by focusing on a clearly defined set of features and user needs:

- The platform emphasizes core AI-driven image editing functionalities, including background removal, object removal, object recoloring, generative fill, and image restoration. In addition to these advanced features, the platform also supports essential user interactions such as uploading images, applying transformations, previewing results, and downloading the final output.
- Auth system with usage limits for free/premium users. Free users will get 2 transformations, and premium users have unlimited access.
- It targets a broad user base ranging from individuals to businesses, particularly those involved in e-commerce, digital marketing, content creation, and personal design projects.
- The system aims to provide an intuitive and accessible user experience, eliminating the need for specialized technical skills or professional design software.
- Designed with scalability in mind, the platform is built to accommodate varying workloads and support both casual and high-volume users.

1.6 Structure of Study

Table 1. Structure of Study

Chapter Number	Descriptions
Chapter 1: Introduction	This chapter introduces the study, outlining its objectives, problem statement, and the motivation behind developing the Image Editor Platform. It also presents the project scope and rationale.
Chapter 2: Literature Review	This chapter provides an overview of related work, comparing existing AI-based image editing platforms and tools. It highlights the gaps that the proposed system aims to address.

Chapter 3: Methodology	This chapter describes the methodology used in system development, including the system architecture, system workflow, algorithmic approach, requirements, database design, and third-party API integration (Cloudinary and Paddle).
Chapter 4: Result and Evaluation	This chapter presents the system testing process, test results, and evaluation against the initial objectives, along with observations on system performance.
Chapter 5: Discussion	This chapter analyzes findings, addresses challenges and limitations, interprets results, and discusses lessons learned and possible improvements.
Chapter 6: Conclusion	This chapter summarizes the study, highlights key outcomes, and outlines directions for future work, including AI service expansion and new feature integration.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview of the Research Topic

The integration of artificial intelligence (AI) into image editing has revolutionized the way visual content is created and manipulated. Traditional image editing tools often require significant expertise and time investment. In contrast, AI-powered platforms offer automated solutions for complex tasks such as background removal, object recoloring, and image restoration. Cloudinary, a cloud-based media management platform, provides robust APIs and intelligent transformation tools that facilitate scalable and efficient image editing. This study focuses on implementing an AI-powered image editor platform utilizing Cloudinary's services to enhance accessibility and efficiency in image processing.

2.2 Key Theories and Models

2.2.1 Overview of Theories

The foundation of AI-powered image editing lies in theories of machine learning and computer vision. Machine learning enables systems to learn patterns from data, improving performance over time without explicit programming. Computer vision allows machines to interpret and process visual information, facilitating tasks such as object detection, segmentation, and image classification. These theories underpin the development of AI models capable of performing complex image editing tasks with minimal human intervention.

2.2.2 Overview of Models

The rapid advancement of the Image Editor Platform (Artify) has been driven by key deep learning architectures. These models form the foundation of many other modern image editing platforms, including services like Cloudinary.

Convolutional Neural Networks (CNNs) have been widely adopted for their effectiveness in image classification, segmentation, and object detection tasks. CNNs are known for their ability to capture spatial hierarchies in images, making them ideal for foundational visual processing (Krizhevsky, Sutskever, & Hinton, 2012).

Generative Adversarial Networks (GANs) have revolutionized synthetic image creation. These models work through a generator–discriminator dynamic, enabling

high-quality image synthesis, super-resolution, inpainting, and more (Goodfellow, Bengio, & Courville, 2016).

Transformer-based models, originally developed for natural language processing, are increasingly applied to vision tasks. Vision Transformers (ViTs) interpret images as sequences of patches, allowing them to model long-range dependencies and contextual relationships in visual data. This proves useful in tasks such as image captioning and detailed scene understanding (Dosovitskiy et al., 2021).

Platforms like Cloudinary leverage these model types to deliver advanced capabilities such as generative fill, background replacement, and automated image enhancement at scale (Cloudinary Ltd., 2025a) (Cloudinary Ltd., 2025b).

2.3 Previous Studies and Findings

2.3.1 Overview of Existing Applications

A growing number of AI-powered image editing tools have emerged in recent years, each demonstrating unique strengths in automation, creativity, and user accessibility. This section provides an overview of several prominent applications that exemplify the current capabilities and limitations of AI in image editing.

2.3.1.1 Remove.bg - Specialized AI Background Removal

Remove.bg is an AI-driven platform designed to automate the process of background removal from images. Utilizing advanced machine learning algorithms, it efficiently distinguishes the foreground subject from the background, delivering high-quality results in seconds (Kaleido AI GmbH, 2025).

Key Features:

- Automatic Background Removal: Employs deep learning techniques to separate subjects from backgrounds without manual intervention.
- Batch Processing: Supports bulk image processing, enabling users to remove backgrounds from multiple images simultaneously.
- API Integration: Offers API access for seamless integration into various applications and workflows.
- Editing Tools: Includes basic tools for fine-tuning results, such as adjusting edges and adding new backgrounds.

Use Cases: Widely utilized in e-commerce for product image optimization, marketing for creating promotional materials, and by graphic designers for efficient background editing.

Limitations: While Remove.bg excels in background removal, it offers limited customization and may struggle with complex images requiring precise detailing.

2.3.1.2 Pixelcut - AI Generative Fill for Creative Editing

Pixelcut's Generative Fill tool leverages AI to enable users to fill in images or replace elements using text prompts. This feature allows for intuitive and creative photo editing directly within the browser (Pixelcut Inc., 2025).

Key Features:

- Text-Prompt Editing: Users can describe desired changes, and the AI modifies the image accordingly.
- Object Replacement: Facilitates the replacement of specific elements within an image by highlighting and describing the new object.
- Background Transformation: Enables users to change backgrounds to different scenes or settings using descriptive prompts.

Use Cases: Ideal for content creators, marketers, and individuals seeking to enhance images for social media, product photography, or personal projects.

Limitations: The effectiveness of edits depends on the clarity of text prompts, and results may vary based on image complexity.

2.3.1.3 Imaginify - AI Image Generation and Editing

Imaginify, developed by SlideAI, is an AI-powered platform that allows users to generate and edit images based on textual descriptions. By integrating with Google's Gemini model, it offers a seamless experience for creating and refining images (SlideAI, 2025).

Key Features:

- Image Generation: Users can input detailed descriptions to generate unique images tailored to their specifications.
- Interactive Editing: Provides the ability to refine generated images by describing desired changes, enabling iterative enhancements.
- User-Friendly Interface: Designed for ease of use, allowing users to create and modify images without requiring advanced technical skills.

Use Cases: Suitable for designers, marketers, and individuals looking to create custom visuals for presentations, campaigns, or personal use.

Limitations: The platform's capabilities are influenced by the specificity of user inputs, and achieving desired results may require multiple iterations.

2.3.1.4 Canva - Integrated AI Photo Editor in a Design Suite

Canva combines its widely used, web-based design platform with AI-powered photo editing tools. It caters to users with little to no design experience, enabling the quick creation of visually compelling content for digital and print media (Canva Pty Ltd., 2025).

Key Features:

- Template-Driven Editing: Speeds up content creation using thousands of pre-designed templates for social media, marketing, education, and business needs.
- AI-Powered Tools: Includes Magic Edit (object replacement via text), background remover, Magic Eraser, and auto-enhancement filters.
- Drag-and-Drop Interface: Highly intuitive and user-friendly, making design accessible even to novices.
- Brand Kit & Collaboration: Allows teams to maintain consistent branding and collaborate in real time.
- Cross-Platform Availability: Accessible on web and mobile platforms, supporting on-the-go editing.

Use Cases: Ideal for creating marketing materials, branded content, infographics, presentations, and quick social media graphics. Widely used by educators, marketers, small business owners, and social media managers.

Limitations: While Canva excels in user-friendliness and speed, its AI features are relatively limited in depth compared to specialized image editors. The platform focuses more on design layout than advanced photo manipulation, making it less suitable for professional-grade image restoration or detailed retouching. Additionally, some of its more powerful features are locked behind a premium subscription, which may restrict access for free users.

2.3.2 Comparative Literature Review

Table 2. Comparative Analysis of Literature Review

Platform/ Feature	**Remove.bg**	**Pixelcut Generative Fill**	**Imaginify Image Generation**	**Canva**
Primary Function	Background removal	Object and background replacement via text prompts	Image generation and editing via descriptions	AI editing in design suite
AI Integration	Deep learning for background detection	AI-driven generative fill	Google's Gemini model for image creation	AI integrated in templates and photo tools
User Interaction	Upload image, automatic processing	Upload image, select area, input prompt	Input description, generate/edit image	Drag-and-drop editing with AI enhancements
Customization	Limited to background adjustments	High, based on user prompts	High, iterative editing based on feedback	High, combined with design flexibility
Ideal Use Cases	E-commerce, marketing, photography	Social media content, product photography	Design, marketing campaigns, presentations	Marketing, branding, quick designs
Limitations	Limited customization, struggles with complex images	Dependent on prompt clarity, variable results	Requires specific descriptions, iterative process	Templates may limit creative flexibility

2.3.2 Cloudinary

Cloudinary is a cloud-based media management platform that uses artificial intelligence to automate and enhance image and video workflows. It allows developers

and content creators to store, transform, optimize, and deliver media at scale. By integrating generative AI tools such as background removal, object replacement, and inpainting—Cloudinary simplifies complex editing tasks and accelerates content creation. Its API-first approach and support for real-time media optimization make it especially valuable for e-commerce, publishing, and digital marketing. Cloudinary improves performance and user experience by ensuring that visual assets are always delivered in the right format, size, and resolution for each platform or device (Cloudinary Ltd., 2025c).

2.3.4 Paddle

Paddle is an AI-enhanced revenue platform designed to help SaaS businesses manage billing, payments, taxes, and compliance across global markets. It consolidates subscription management, payment processing, and revenue analytics into a single platform, reducing the need for multiple third-party tools. Paddle's AI capabilities support fraud detection, churn prediction, and intelligent pricing strategies, enabling SaaS companies to scale efficiently while maintaining regulatory compliance. By abstracting away the operational complexity of monetization, Paddle allows businesses to focus on product innovation and customer growth (Paddle.com Market Ltd., 2025a).

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter details the methodology adopted in the development of an Image Editor Platform (Artify) that utilizes Cloudinary's intelligent media transformation capabilities. The methodology outlines the system design, workflows, technologies, and tools used to implement the platform, ensuring that it meets the intended objectives of accessibility, scalability, and intelligent automation. The system architecture is based on a modern full-stack approach, emphasizing modular design and cloud-based processing to handle image editing operations efficiently.

3.2 Methods and System Workflow

The system follows an incremental, component-based development methodology. Initially, functional requirements were identified, which led to designing a modular architecture separating the frontend, backend, AI services, and database management. The workflow begins when a user uploads an image via the frontend interface. The image is then sent to the backend, where it is processed by Cloudinary's AI-powered APIs for specific tasks such as background removal, object recoloring, generative fill, or image restoration.

Upon processing, the modified image is returned to the user interface for preview and download. Simultaneously, user access is managed through a secure subscription mechanism integrated with Paddle, offering monthly and yearly plans. This subscription system determines the availability of premium editing features based on the user's active plan. The platform also maintains a usage log and user data in a structured MySQL database, ensuring accurate reporting and analytics (Oracle Corporation, 2025). This workflow supports a seamless user experience with minimal latency and high reliability.

3.2.1 Architectural Design

The Image Editor Platform adopts a layered and modular architecture designed for scalability, performance, and maintainability. The system consists of the following core components:

- Frontend Layer (Next.js) Built using the React-based Next.js framework, the frontend handles user interaction, routing, and UI logic. It communicates with

backend APIs to manage image uploads, display results, and control access based on subscription status.

- Backend Layer (Node.js + Express) This layer processes API requests from the frontend. It includes routes for image submission, user authentication, subscription verification, and access control. It also handles communication with Cloudinary and Paddle.
- Cloudinary AI Processing Layer This service layer integrates Cloudinary's APIs to execute intelligent image transformations. Each transformation (e.g., background removal or restoration) is performed using predefined presets or dynamic API parameters.
- Paddle Subscription Service Paddle manages subscription billing and access control. It sends webhook notifications to the backend for events such as new subscriptions, plan upgrades, cancellations, or payment failures.
- MySQL Database Layer All structured data, including user profiles, image metadata, and subscription history, is stored in MySQL. The database schema ensures relational consistency and supports analytical queries.

3.2.2 Design Workflow

The system was developed through an agile, sprint-based design approach. The workflow involves the following stages:

1. User Uploads Image: Users log in and upload images through the Next.js interface. Files are temporarily stored in memory or a CDN.
2. Backend Processing: The uploaded image is forwarded to the Node.js backend, which determines the required transformation type and passes the image to Cloudinary.
3. AI-Powered Transformation: Cloudinary performs the requested transformation (e.g., background removal using e_background_removal) and returns a secure URL for the modified image.
4. Frontend Display and Download The frontend receives the processed image and displays it in a preview interface. Users can then download the edited result.
5. Subscription Access Check When premium features are requested, the backend verifies the user's plan via the Paddle API and webhook events. If unauthorized, access is restricted with a prompt to upgrade.

6. Logging and Storage Edited image data and user activity logs are stored in MySQL for future access, analytics, and subscription validation.

3.2.3 Summary of the Methods and Algorithms

The implementation combines asynchronous operations, RESTful architecture, and intelligent automation. Key methods and logic include:

- Image Transformation Algorithms Cloudinary's AI models use deep learning to perform real-time background removal, object inpainting, and photo restoration.
- Access Control Logic A subscription-based access algorithm determines feature availability. It checks Paddle webhook data against the user's role in the MySQL database to enable or restrict features.
- Webhook Event Handlers Paddle webhook events are parsed and validated using HMAC signatures. The backend uses this data to update user access and notify them of changes.
- Data Persistence and Retrieval SQL joins and indexed queries enable fast retrieval of image history, subscription timelines, and usage reports.
- Authentication and Session Management JWT tokens are used for stateless authentication. Middleware validates tokens on each request to secure APIs.

These combined mechanisms enable the system to provide a powerful, scalable, and user-centric image editing experience, with intelligent access control and real-time processing.

3.3 Tools and Technologies Used

To implement this platform effectively, a combination of modern web technologies and cloud services was employed.

3.3.1 Next.js

Next.js, a React-based framework, was used for building the frontend interface (Vercel Inc., 2025). It supports server-side rendering and static site generation, providing fast load times and improved SEO. Next.js also simplifies routing and API integration, making it suitable for dynamic, scalable web applications.

3.3.2 Node.js

Node.js served as the backend runtime environment (OpenJS Foundation, 2025). Its event-driven, non-blocking I/O model allows for high-performance API handling and real-time processing. Express.js, a minimal Node.js web framework, was used to structure RESTful API routes for image upload, processing, user authentication, and subscription management.

3.3.3 Cloudinary

Cloudinary, a cloud-based media management platform, was the core engine for image editing. It offers robust AI tools such as automatic background removal, object detection, and intelligent image enhancement through its API. Its scalable infrastructure and advanced transformation features allow for seamless integration and real-time processing of images.

3.3.4 Paddle

Paddle was integrated for subscription billing and payment management. It supports flexible billing cycles, secure payment processing, and webhooks that enable real-time updates for subscription status, cancellations, or renewals. This integration ensures the platform can manage user access to premium features efficiently.

3.3.5 MySQL

MySQL, a reliable relational database, was used to manage structured data such as user profiles, subscription records, and image editing history. It provides a robust querying system and strong data consistency guarantees, which are crucial for analytics and billing logic.

3.4 Requirements Analysis

The development of the Image Editor Platform (Artify) was driven by a comprehensive analysis of both functional and non-functional requirements. These requirements ensure the system meets user needs while maintaining high performance, security, and scalability.

3.4.1 Functional Requirements

Functional requirements define the specific behaviors and functionalities the system must support. The platform includes the following core functions:

- User Authentication: Users must be able to register, log in, and manage their profiles securely.
- Image Upload: Users should be able to upload images from their local device for editing.
- Image Editor Platform Tools: The system must allow users to apply AI-based editing functions, including background removal, object recoloring, generative fill, and image restoration.
- Preview and Download: Users should be able to preview edited images in real time and download the final output.
- Subscription Management: The system should allow users to subscribe to monthly or yearly plans through a secure billing gateway.
- Paddle Webhook Integration: The system must handle real-time updates on subscription status—such as activation, cancellation, or renewal—through Paddle webhooks. These webhooks trigger backend updates to ensure users' access rights are always in sync with their subscription status (Paddle.com Market Ltd., 2025b).
- Feature Access Control: The platform must restrict access to premium features based on the user's active subscription, as verified through webhook data.
- Usage Logging: The system must track user activities such as image uploads and edits for analytics and support purposes.

3.4.2 Non-Functional Requirements

Non-functional requirements place more emphasis on the system's quality characteristics than on particular actions. The following are the main non-functional requirements:

- Performance: To guarantee a seamless user experience, the system should react to requests for image processing with the least amount of latency possible.
- Scalability: Without sacrificing performance, the architecture must be able to handle growing user counts and request processing.

- Security: All user data, including payment and authentication details, must be sent and kept in a secure manner.
- Usability: Even non-technical users should be able to edit images with ease thanks to an intuitive and user-friendly interface.
- Reliability: Even with heavy usage, the platform should be able to gracefully handle errors and maintain uptime.
- Maintainability: To accommodate upcoming improvements or debugging, the system needs to be modular and thoroughly documented.

3.5 Use Case Diagram

The Image Editor Platform's use case model explains how the user interacts with the main features of the system. A user first registers or logs into their account when they access the platform. The user can upload an image to the system for editing after authenticating. Depending on their subscription level, they can use a variety of image editing tools after uploading, including generative fill, object recoloring, background removal, and image restoration.

The system shows a real-time preview of the edited image after the editing process is complete. The user can download the finished product if they are happy with the outcome. Through the integrated billing system, users can sign up for a monthly or annual plan to access premium features. Additionally, there are subscription management features that let users change, renew, or cancel their plans whenever they want.

Paddle webhooks, which provide real-time notifications regarding modifications to the user's subscription status, are used by the system to manage subscription updates in the background. The backend modifies access permissions in accordance with these notifications. Users can also examine the activity logs and editing history that are kept in the database. An interactive, automated, and customized user experience is supported by this extensive collection of use cases.

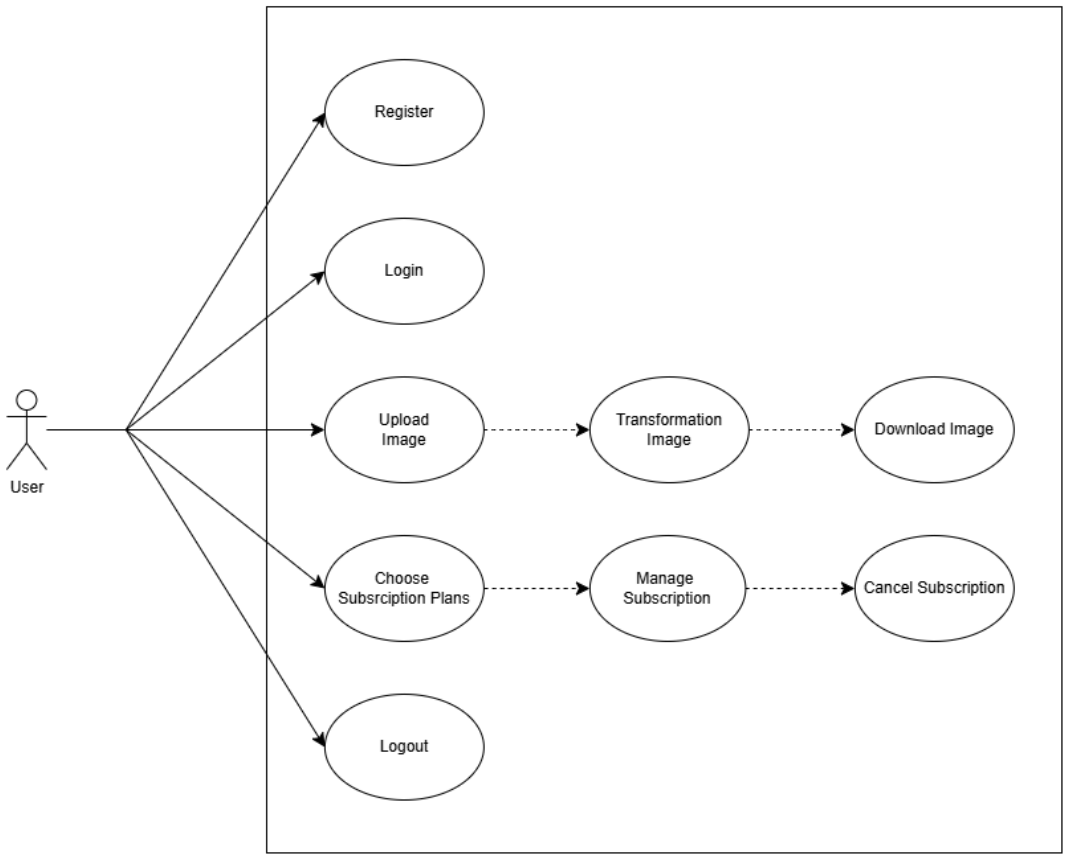


Figure 1. Use Case Diagram

3.6 User Flow Diagram

The user flow begins with authentication, where users are prompted to register if they haven't already. Once authenticated, users land on the homepage and are checked for an active subscription. Subscribed users gain full access to all transformation tools, including image upload, Cloudinary processing, and saving results. Unsubscribed users are given the option to subscribe or proceed with a limited two-image transformation trial. If the trial limit is exceeded, they are redirected to subscribe. Subscribed users can also manage or cancel their subscriptions as needed. The flow ensures a clear distinction between free and premium access, guiding users through a logical path based on their subscription status.

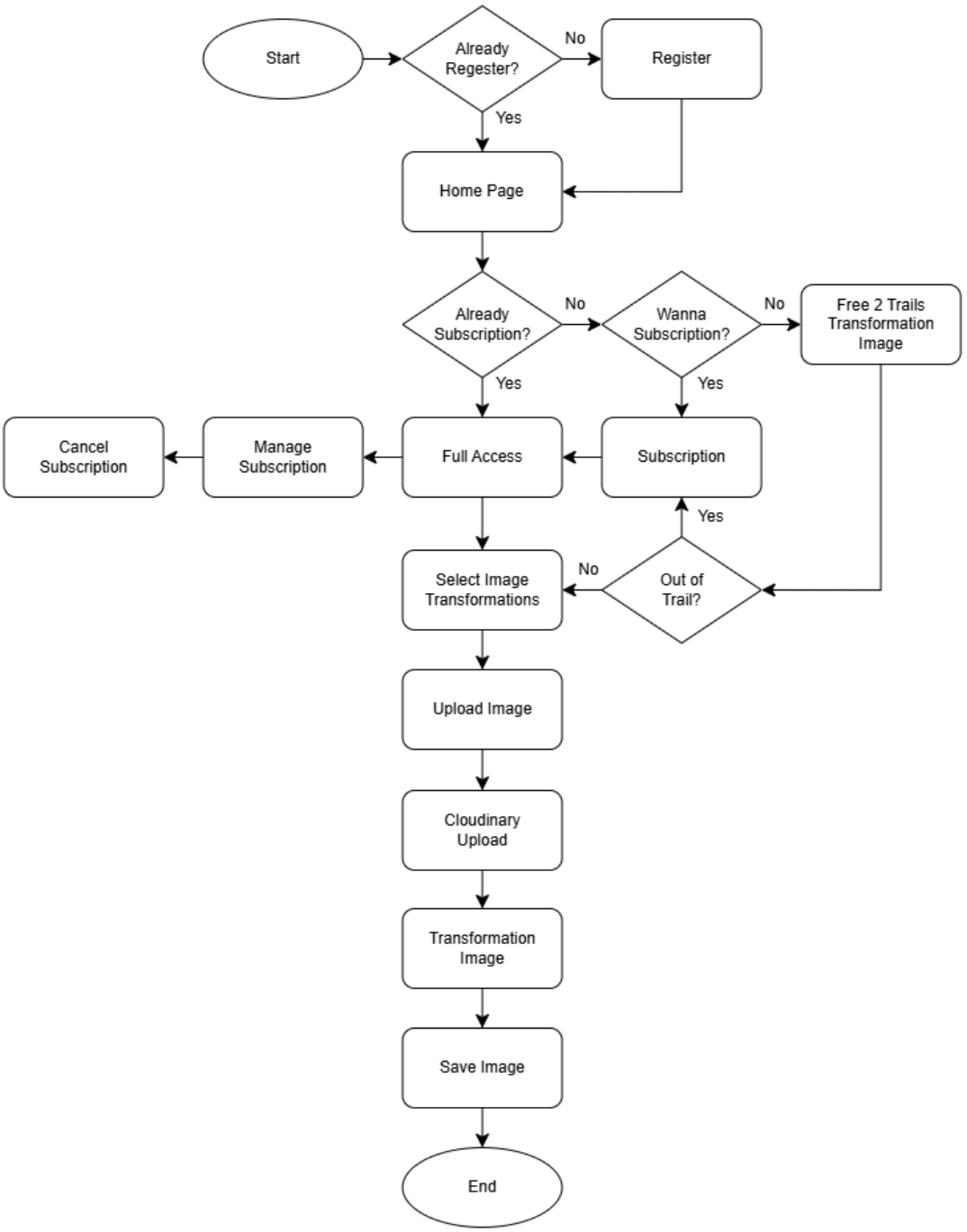


Figure 2. User Flow Diagram

3.7 System Architecture

The platform's system architecture is made to facilitate real-time performance, scalability, and modularity. The frontend, backend, external AI services, payment processing layer, and database make up its multi-layered architecture.

Next.js is used to build the user interface at the frontend, offering a smooth and responsive environment for previewing, editing, and uploading images. This interface

manages user sessions and starts image processing operations by interacting with the backend via RESTful APIs.

Node.js and Express are used in the backend's implementation to handle application logic, handle API requests, and transfer data between the database, AI services, and user interface. It is in charge of managing webhook processing, user authentication, subscription verification, and edit task coordination.

The external AI service in charge of carrying out the image transformation tasks is called Cloudinary. The backend sends the image to Cloudinary's API when a user chooses an editing option, where it carries out tasks like object detection, generative fill, and background removal. The altered image is sent back to the frontend for user preview and download after processing is finished.

Paddle is used to manage access control and subscription billing. The system manages payments when users sign up and starts webhooks to notify the backend of any modifications to the subscription status. These webhook events regulate access to premium features according to the user's plan and automatically update the user's subscription record.

Lastly, all system data is centrally stored in the MySQL database. It keeps track of webhook logs, editing histories, user profiles, and subscription information. To guarantee system integrity and user accountability, the database facilitates accurate reporting, billing validation, and audit tracking.

This architectural design guarantees that the platform can provide real-time, high-quality image editing services while managing user data and subscriptions in an effective and secure manner.

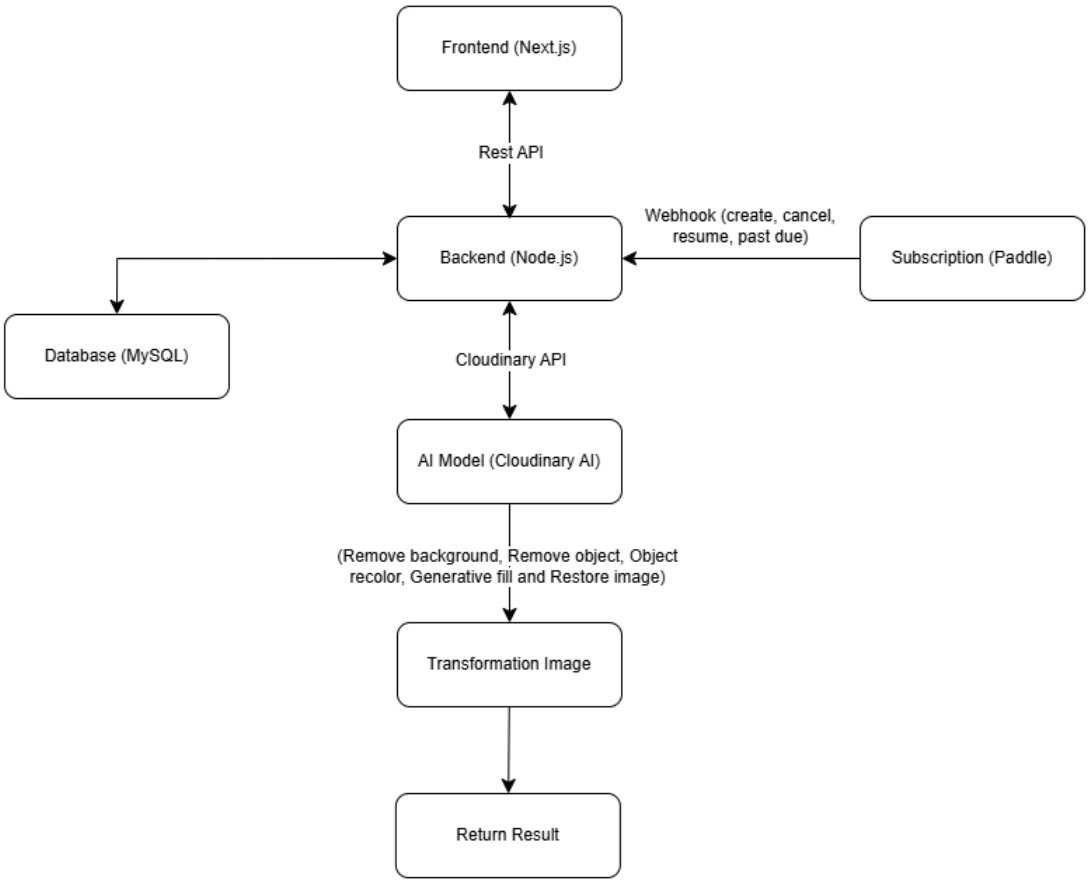


Figure 3. System Architecture

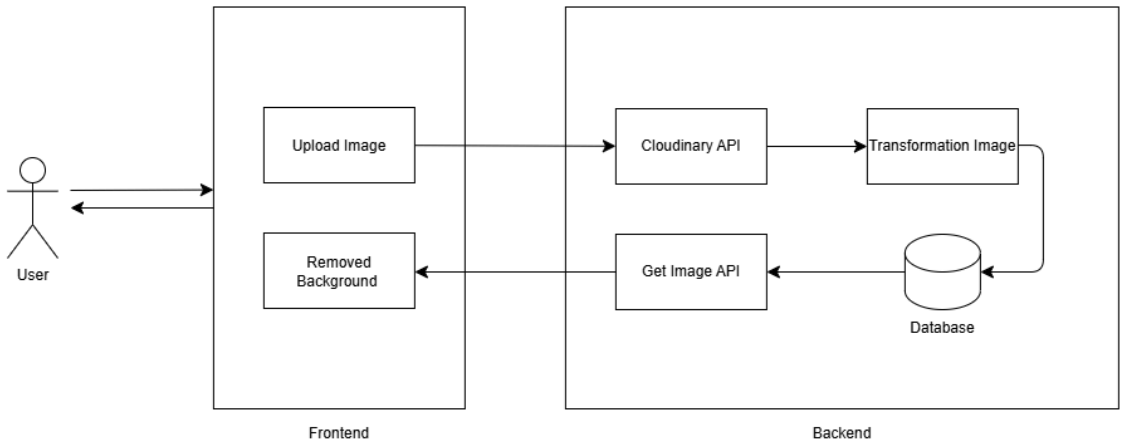


Figure 4. Image Transformation Form Diagram

3.8 Design Principles

To guarantee a reliable and future-proof solution, the system design complies with contemporary software engineering principles. During architectural planning, important concepts like scalability and modularity were given top priority.

3.8.1 Modularity

By breaking the system down into distinct parts, such as the frontend, backend, database management, AI service integration, and third-party payment gateway (Paddle), modularity was accomplished. Every module is in charge of a particular function and communicates with the others via clear APIs. In addition to improving maintainability, this division of responsibilities enables developers to work on various system components separately without creating dependencies or conflicts.

3.8.2 Scalability

A stateless backend architecture and cloud-native tools were used to integrate scalability into the system. Paddle offers scalable subscription management, while Cloudinary manages image processing on a large scale. Concurrent connections and asynchronous requests are supported by the Node.js-built backend, which makes it appropriate for heavy traffic. By utilizing MySQL configurations to enable sharding or replication, the database design additionally facilitates horizontal scaling.

3.9 Database Design

The database was created with the purpose of effectively storing and retrieving structured data, such as payment histories, user profiles, and picture data. Because of its strength and ease of enforcing relationships between various data entities, a relational schema was chosen.

3.9.1 Schema Design

The three main tables in the platform's database are Users, Payments, and Images. A char(36) UUID is used as the primary key in each table to guarantee distinct and reliable identification across dispersed systems.

- Credentials and profile information are kept in the Users table.
- Each user's subscription information, including billing cycle and status, is tracked in the Payments table.
- Every image that users upload or edit is recorded in the Images table.

Standard creation and update timestamps are included in every table to make auditing and activity tracking easier.

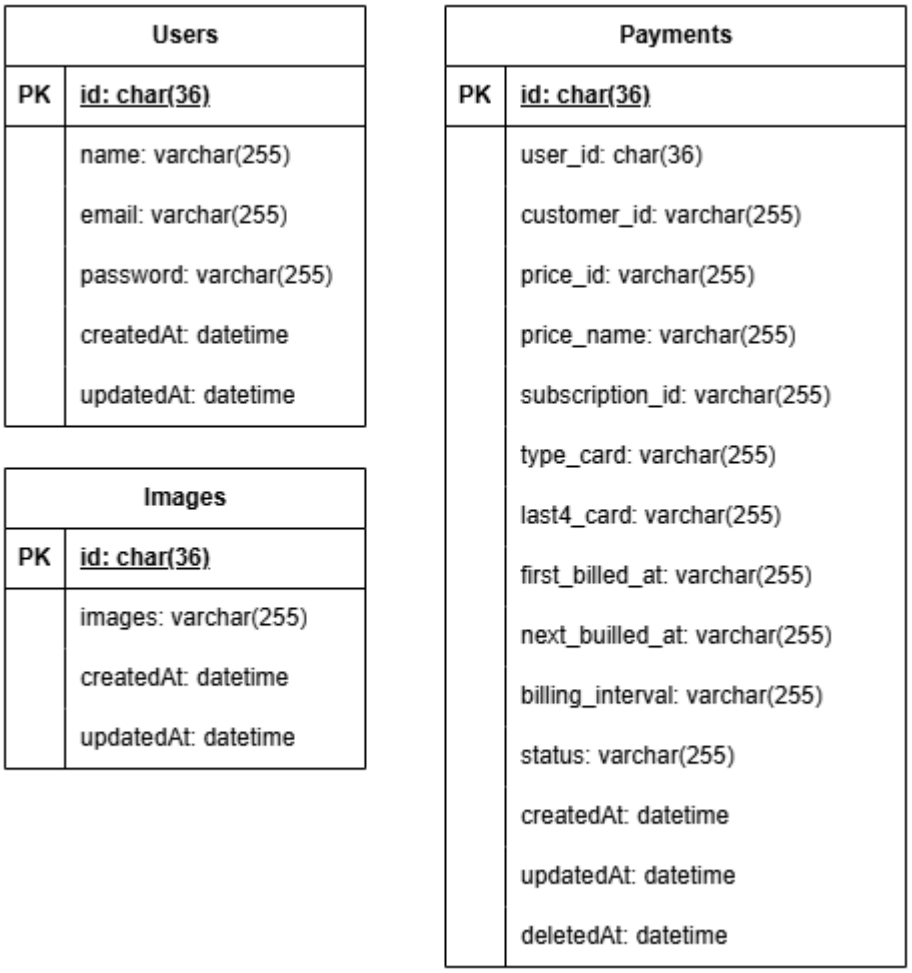


Figure 5. Database Schema Design

3.9.2 Data Relationship

The relationship model used by the database is one-to-many. Multiple payment entries can be linked to a single user, making it possible to track billing events, plan modifications, and recurring subscriptions. Similar to this, users have the ability to upload and manage numerous images, each of which is saved and associated with a unique user

ID. This relational integrity is established in the Payments and Images tables using the foreign key user_id.

Normalized data storage is supported by this relational design, which also reduces redundancy and guarantees consistency throughout platform operations.

3.10 Development Process

After gathering requirements, each module was designed and implemented as part of an incremental and iterative approach to system development. In the first stage, Next.js and Node.js were used to set up the frontend and backend environments, respectively. Later, Paddle was set up for subscription billing, and Cloudinary was integrated to manage image transformations.

Before being incorporated into the entire system, each component was created and tested separately. Git and GitHub were used to maintain version control, facilitating change tracking and collaborative development. To guarantee code stability prior to deployment, testing and updates were carried out continuously in both local and staging environments.

3.11 Frameworks Implementations

- To effectively build and implement the platform, a number of frameworks and services were used:
- The frontend was built using Next.js, which provides smooth API routing, server-side rendering (SSR), and static site generation (SSG). Within this framework, React components were used to create previews, image upload forms, and user authentication.
- The backend infrastructure was made up of Express.js and Node.js. To manage user sessions, handle image uploads, call Cloudinary APIs, and get webhook data from Paddle, RESTful API endpoints were developed. Error management and API request authentication were handled by middleware.
- To manage AI-powered image editing, Cloudinary was incorporated. In order to specify the desired transformation, like background removal or object recoloring, the system sends uploaded images to Cloudinary's API. The user interface receives a modified image URL from the API response.

- Paddle was used to control access that was based on subscriptions. Webhook events, such as the creation, cancellation, or renewal of a subscription, are detected by the backend and recorded in the MySQL database. Depending on their subscription status, these events decide which features a user can access.
- Structured data was stored using MySQL. To effectively preserve data, tables for users, payments, and picture history were made, and queries were refined to guarantee integrity and performance.

3.11.1 Cloudinary Transformations

Cloudinary is used by the Image Editor Platform to control image delivery, transformation, and storage. Through the use of transformation expressions supplied as URL parameters or API calls, Cloudinary provides robust server-side image manipulation capabilities. Enabling features like object recoloring, background removal, and generative editing straight from the platform requires this functionality.

During deployment, the platform ensures security and flexibility by connecting to Cloudinary with credentials stored in environment variables. The official Cloudinary Node.js SDK, which supports advanced media queries and uploading, is used for integration.

A significant portion of the implementation, namely the section that retrieves and filters images using transformation expressions, is included below:

```
cloudinary.config({
  cloud_name: process.env.NEXT_PUBLIC_CLOUDINARY_CLOUD_NAME,
  api_key: process.env.CLOUDINARY_API_KEY,
  api_secret: process.env.CLOUDINARY_API_SECRET,
  secure: true,
})

let expression = 'folder=imaginify';

if (searchQuery) {
  expression += ` AND ${searchQuery}`
}

const { resources } = await cloudinary.search
  .expression(expression)
  .execute();

const resourceIds = resources.map((resource: any) => resource.public_id);
```

Figure 6. Cloudinary Transformations Integration for AI-Powered Image Editing

3.11.2 Paddle Webhook Integration

The Image Editor Platform incorporates Paddle's webhook system to automate and manage user subscriptions. Real-time subscription events like creation, cancellation, and completed transactions are monitored by Paddle webhooks. This guarantees smooth synchronization between the platform's access rights and the user's payment status.

These webhook events are handled in the backend by the PaymentController class using Express and Node.js. The Getwebhook method watches for Paddle's HTTP POST requests. Upon receiving a webhook payload, it performs logic based on the event_type and pertinent subscription information that are extracted from the request body.

For instance:

- When a **subscription.created** event is triggered, the system logs the subscription details, such as the customer ID, price ID, billing interval, and timestamps for billing.
- If a **transaction.completed** event is received and the payment status is **captured**, the subscription is confirmed, and the system stores the billing period.
- In the case of **subscription.canceled**, the same data structure is used to mark the subscription as canceled in the database.

The webhook processing ensures time-zone consistency using the dayjs library with UTC plugins and adjusts for latency with a 55-minute offset to align with Paddle's scheduled events.

The following code snippet illustrates the webhook handler:

```
// ========== GET WEBHOOK ========== //
async Getwebhook(req, res) {
try {
  dayjs.extend(utc);
  const payload = req.body;
  const data = payload.data;
  const eventType = payload.event_type;
    You, last week • completed manage subscription
  const firstItem = data.items?.[0];
  const price = firstItem?.price;
  const payment = data.payments?.[0];
  const card = payment?.method_details?.card;
```

```
  if (eventType === 'subscription.created') {
    await IPaymentRepository.add({
      id: uuidv4(),
      user_id: data.custom_data?.user_id,
      customer_id: data.customer_id,
      price_id: price?.id,
      price_name: price?.name,
      subscription_id: data.id,
      type_card: card?.type,
      last4_card: card?.last4,
      first_billed_at: dayjs.utc(data.current_billing_period?.starts_at).add(55, 'minute').format('YYYY-MM-DD HH:mm:ss'),
      next_billed_at: dayjs.utc(data.current_billing_period?.ends_at).add(55, 'minute').format('YYYY-MM-DD HH:mm:ss'),
      billing_interval: price?.billing_cycle?.interval || 'month',
      status: price?.status,
    });
  }
```

```
  if (eventType === 'transaction.completed' && payment?.status === 'captured') {
    await IPaymentRepository.add({
      id: uuidv4(),
      user_id: data.custom_data?.user_id,
      customer_id: data.customer_id,
      price_id: price?.id,
      price_name: price?.name,
      subscription_id: data.subscription_id,
      type_card: card?.type,
      last4_card: card?.last4,
      first_billed_at: dayjs.utc(data.billing_period?.starts_at).add(55, 'minute').format('YYYY-MM-DD HH:mm:ss'),
      next_billed_at: dayjs.utc(data.billing_period?.ends_at).add(55, 'minute').format('YYYY-MM-DD HH:mm:ss'),
      billing_interval: price?.billing_cycle?.interval || 'month',
      status: price?.status,
    });
  }

  if (eventType === 'subscription.canceled') {
    await IPaymentRepository.add({
      id: uuidv4(),
      user_id: data.custom_data?.user_id,
      customer_id: data.customer_id,
      price_id: price?.id,
      price_name: price?.name,
      subscription_id: data.id,
      type_card: card?.type,
      last4_card: card?.last4,
      first_billed_at: dayjs.utc(data.current_billing_period?.starts_at).add(55, 'minute').format('YYYY-MM-DD HH:mm:ss'),
      next_billed_at: dayjs.utc(data.current_billing_period?.ends_at).add(55, 'minute').format('YYYY-MM-DD HH:mm:ss'),
      billing_interval: price?.billing_cycle?.interval || 'month',
      status: price?.status,
    });
  }
```

```
    return res.status(200).json({ message: 'Webhook received', data: eventType });
  } catch (error) {
    console.error('Webhook error:', error);
    return res.status(500).json({ message: 'Webhook processing failed', error });
  }
}
```

Figure 7. Paddle Webhook Integration Code for Subscription Event Handling

Only authorized subscription events can cause changes thanks to the webhook mechanism's efficiency and security. Secure API keys kept in environment variables are used for authorization, and Paddle's server communicates over HTTPS.

In order to maintain the platform's responsiveness to real-time billing events and accurately and consistently enforce the subscription model, this integration is essential.

3.12 Testing and Debugging

Testing was conducted at multiple stages to validate the functionality, performance, and security of the system. Both manual and automated testing strategies were applied.

- Unit testing was performed for backend routes and services to ensure isolated components worked as expected.
- Integration testing checked the interaction between the frontend, backend, and third-party APIs (e.g., Cloudinary and Paddle).
- End-to-end (E2E) testing involved real user scenarios, including signing up, uploading an image, subscribing to a plan, and downloading the processed image.

During the debugging phase, common issues such as incorrect API responses, permission errors, or asynchronous timing problems were resolved using tools like Postman, Chrome DevTools, and backend logging with console.log and debug packages. Errors triggered during image processing or webhook handling were captured and logged for further analysis.

All identified bugs were tracked and resolved iteratively to ensure a robust and reliable final system.

CHAPTER 4

RESULTS AND EVALUATION

4.1 Test Result

Table 3. Test Result

Test Case	Expected Outcome	Observed Result	Status
1	Upload image via frontend	Image uploaded and displayed in the interface	Success
2	Apply background removal using Cloudinary API	Background successfully removed	Success
3	Apply object recoloring	Object recolored with the input color only	Success
4	Use generative fill on missing area	Missing part of image realistically filled	Success
5	Attempt premium feature on free plan	Prompt displayed to upgrade subscription	Success
6	Upgrade subscription via Paddle	Subscription changed and reflected in UI	Success
7	Receive webhook from Paddle on subscription cancellation	Subscription status updated and access restricted	Success
8	Try concurrent uploads (2 simultaneous uploads)	All uploads processed without error	Success
9	Test image download after editing	Edited image downloaded in correct format	Success

4.2 Snapshots and Captions

This section provides a series of snapshots demonstrating the user interface and the results of different image editing features offered by the Artify platform. Each figure is accompanied by a caption that describes the functionality presented and the transformation applied.

4.2.1 Webpage

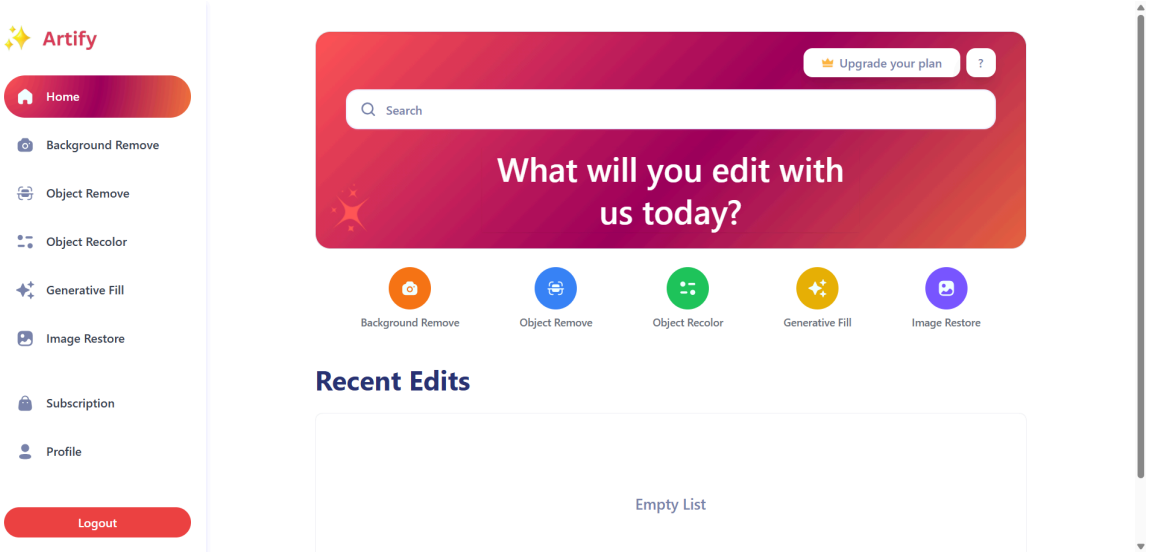
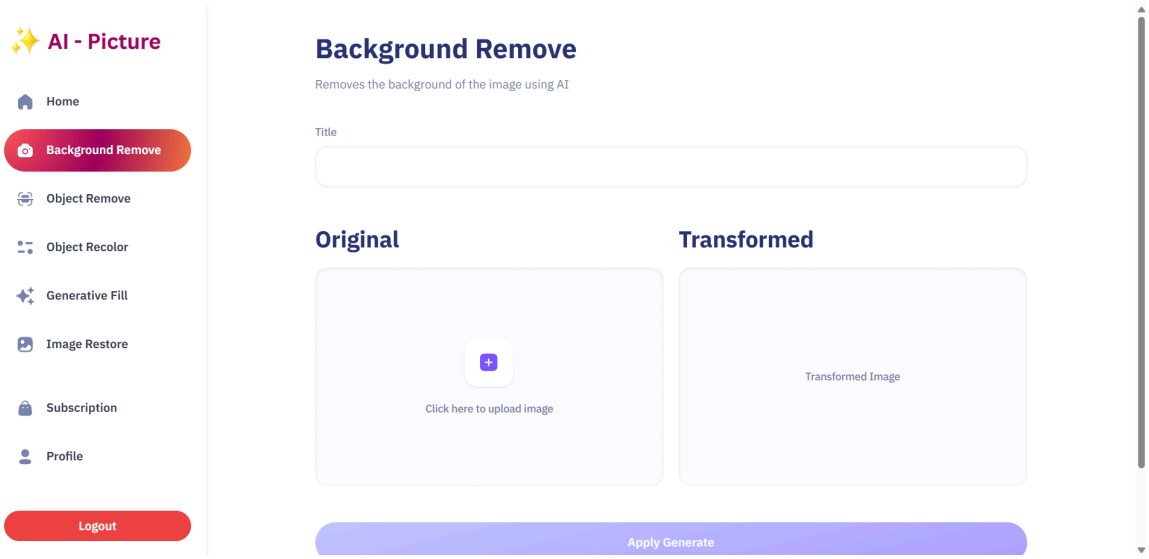


Figure 8. Artify Platform Home Page

The landing page of the Artify platform showcases a simple and user-friendly layout with navigation options for logging in, uploading images, and accessing available tools.



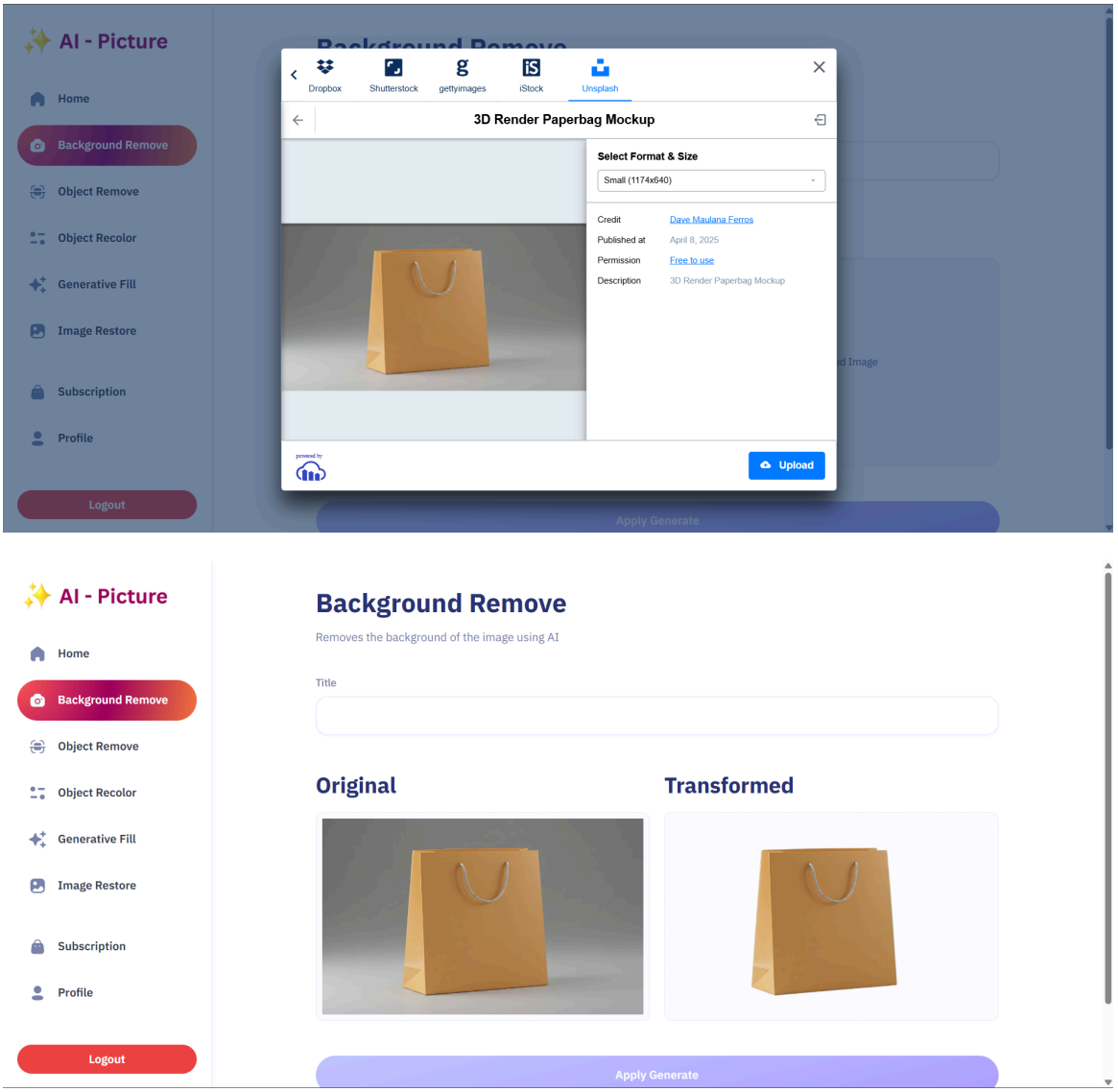


Figure 9. Background Removal Interface and Result Output

This snapshot displays the UI for the background removal feature. It shows the original uploaded image alongside the processed result where the background has been cleanly removed using Cloudinary's background removal transformation.

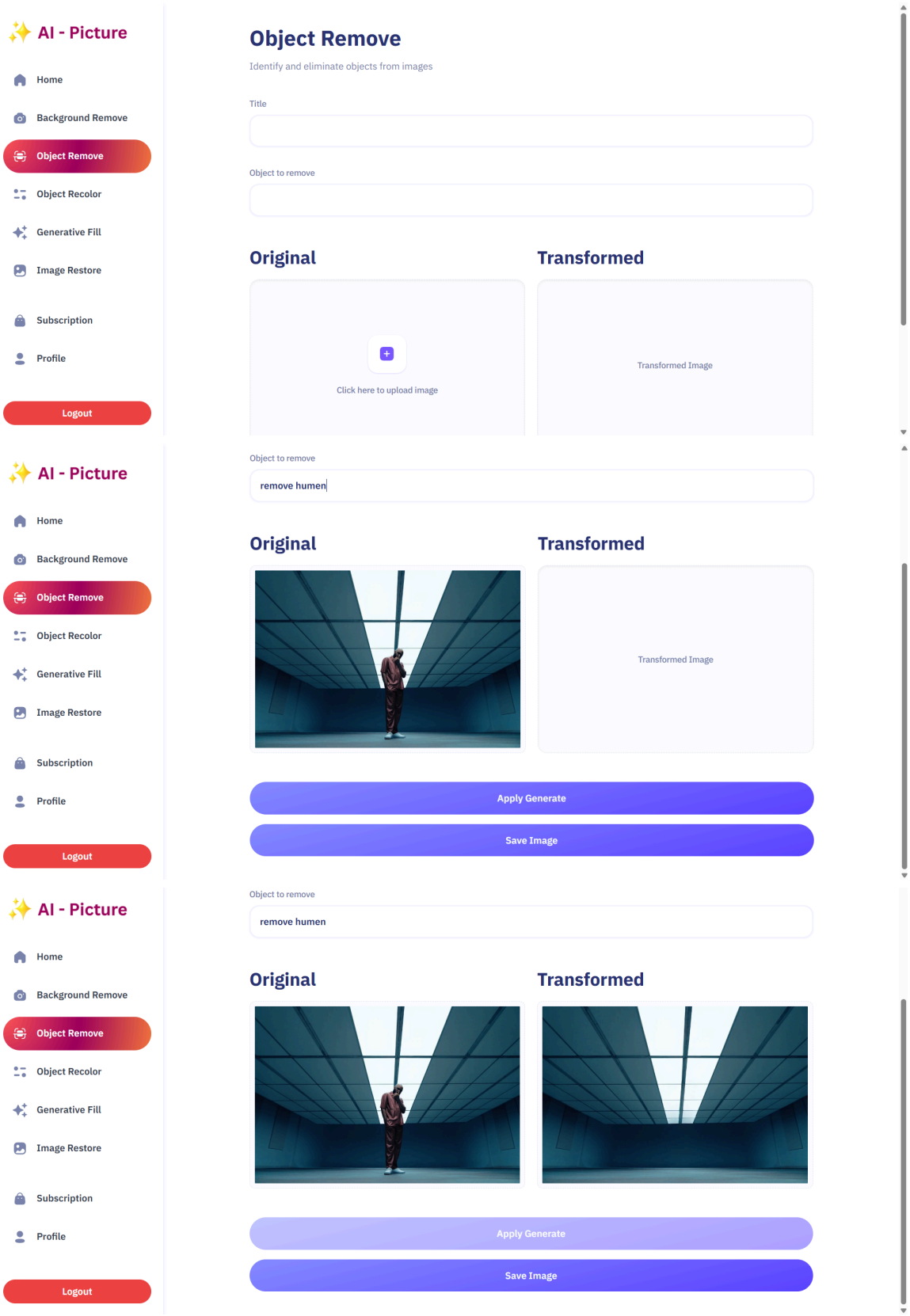


Figure 10. Object Removal Interface and Result Output

The figure shows how a user selects an unwanted object in the image. The result demonstrates successful object removal and inpainting using Cloudinary's generative fill API.

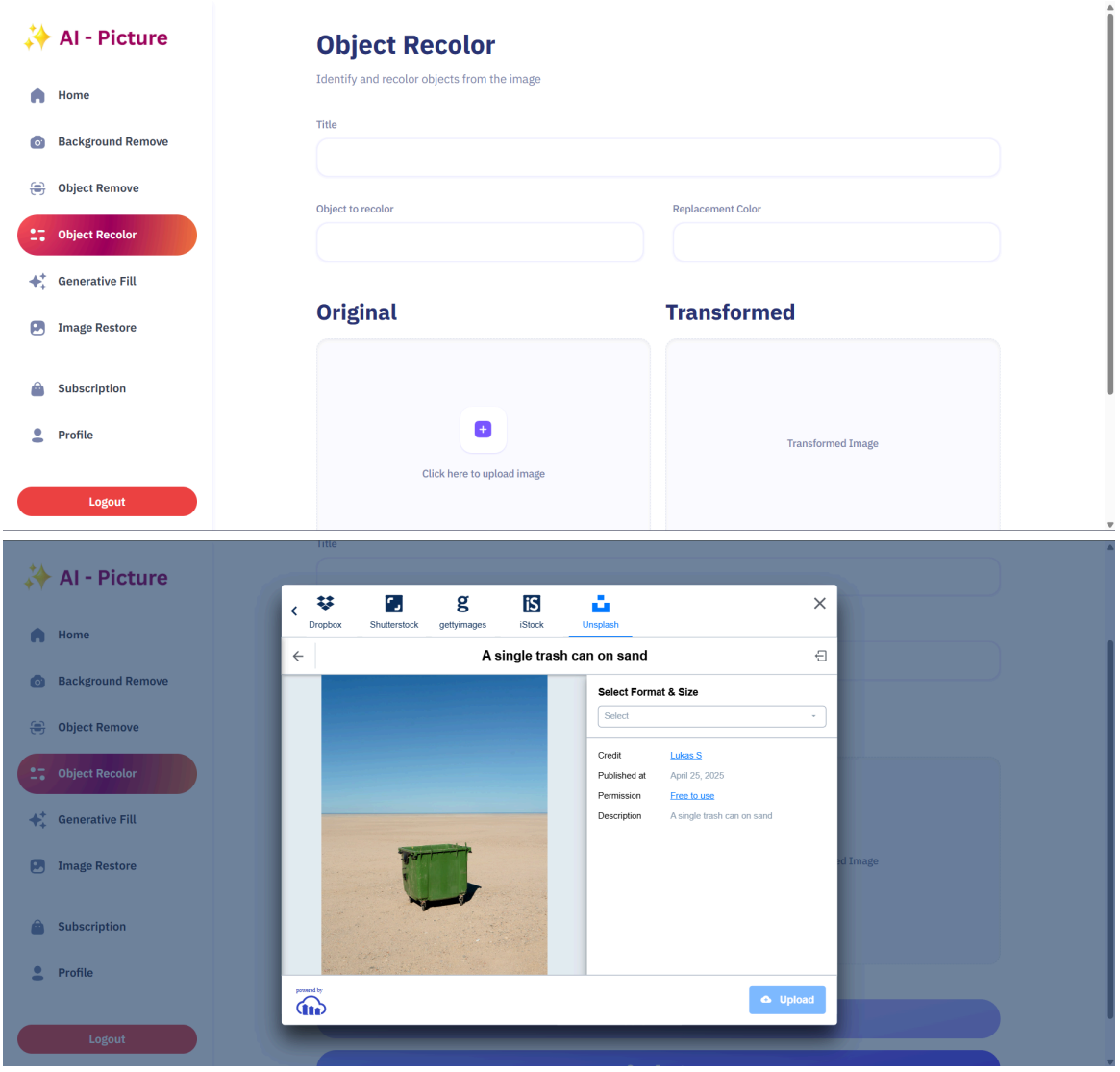
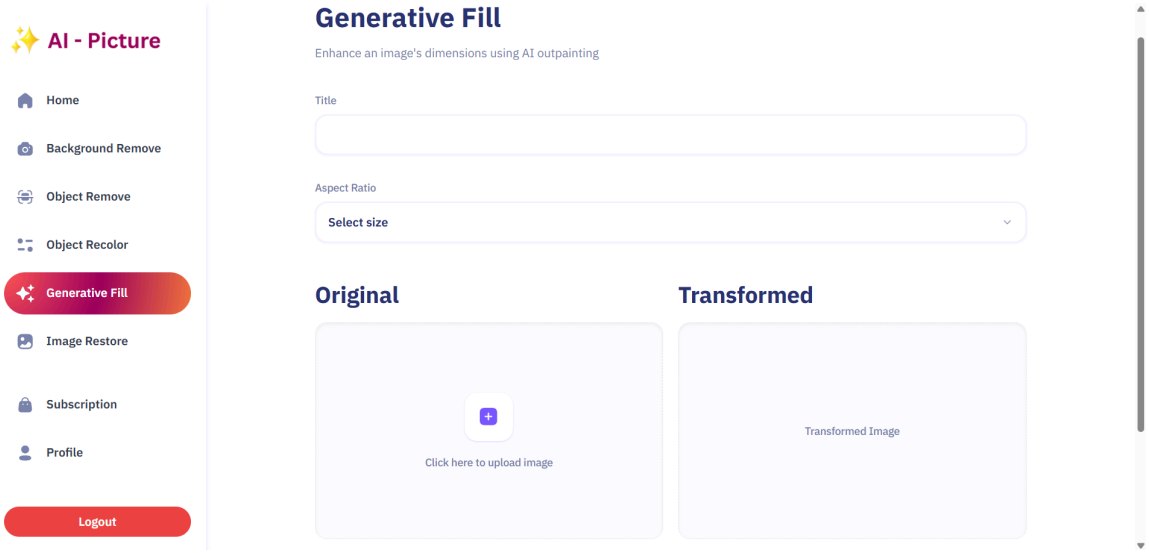


Figure 11. Object Recolor Interface and Result Output

This page allows users to select specific objects to recolor. The result shows a modified object with a new color applied using segmentation and Cloudinary's color manipulation tools.



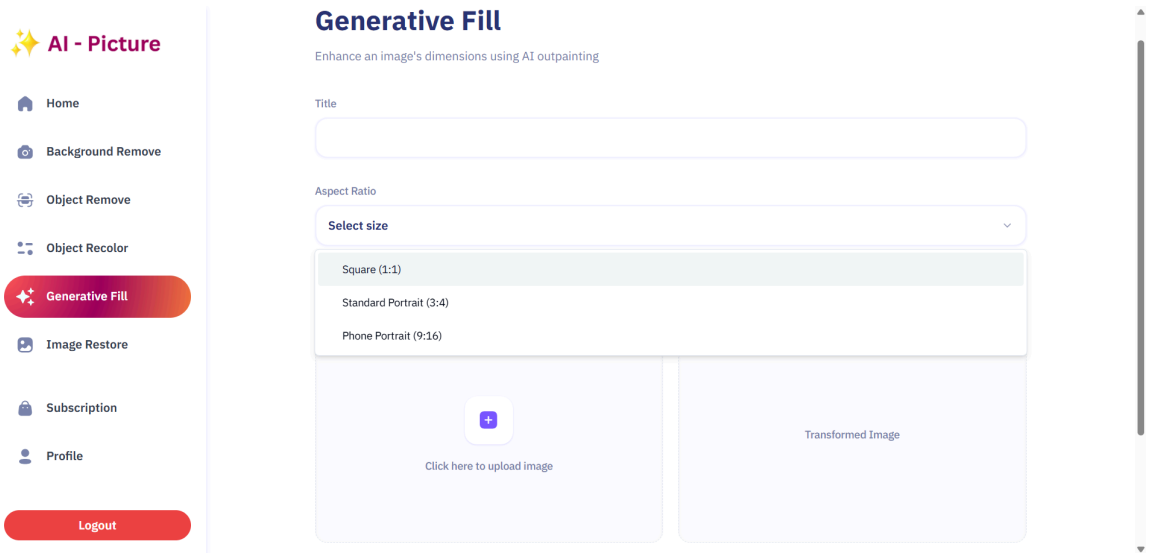
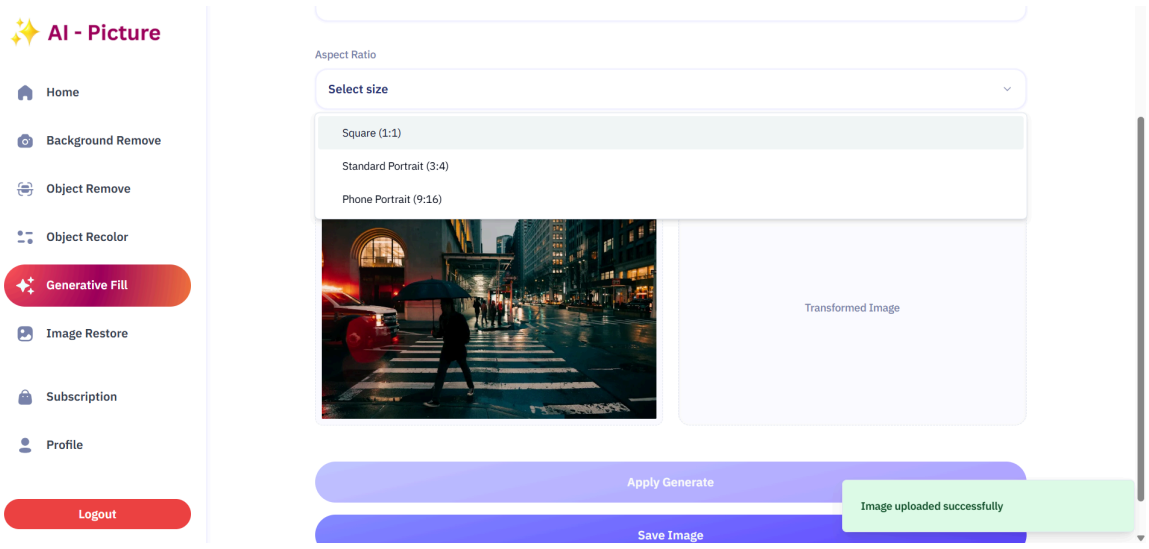


Figure 12. Generative Fill UI with Aspect Ratio Selection

Demonstrates the UI for generative fill. Users can define a masked area and aspect ratio. The result shows the AI-generated image content filling the blank space using a prompt-based generation.



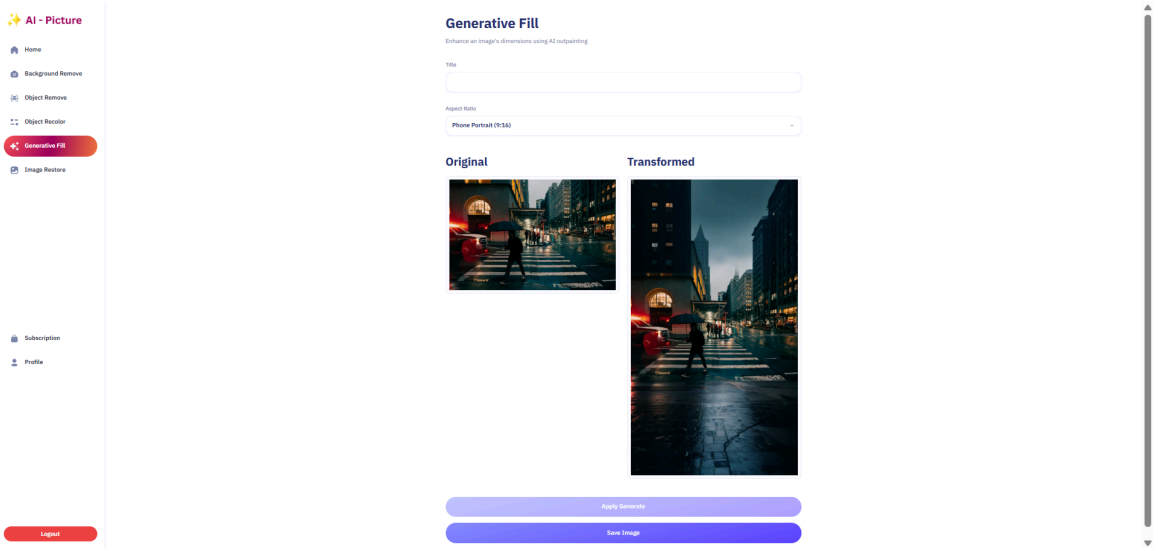
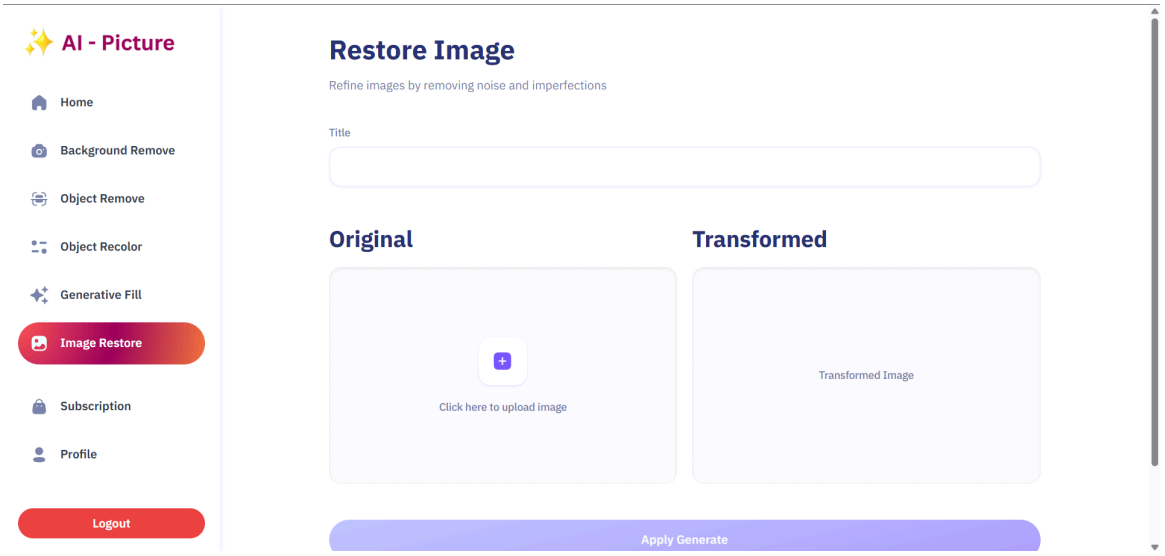


Figure 13. Image Upload and Generative Fill Result Output

This figure combines the image upload process and the final output after applying the generative fill feature.



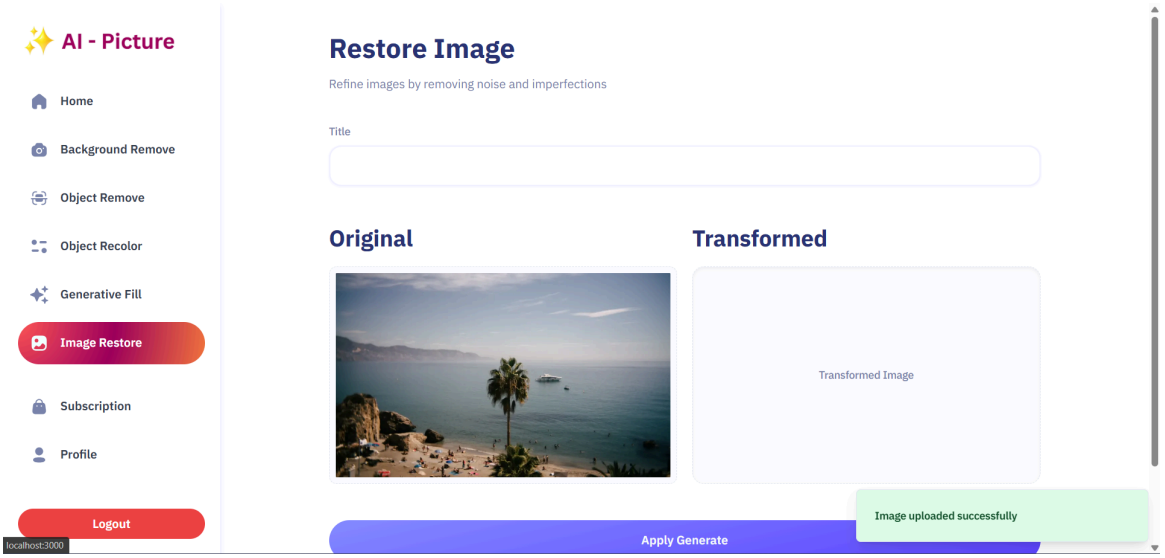
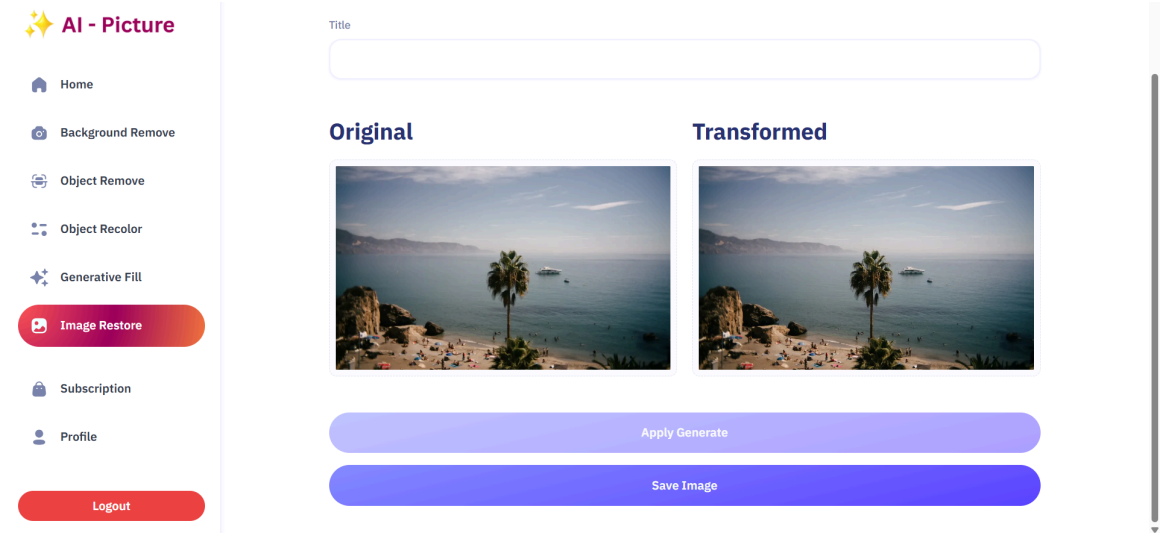


Figure 14. Image Restoration Page Before Enhancement

Displays the restoration interface with the uploaded damaged or low-resolution image ready for enhancement.



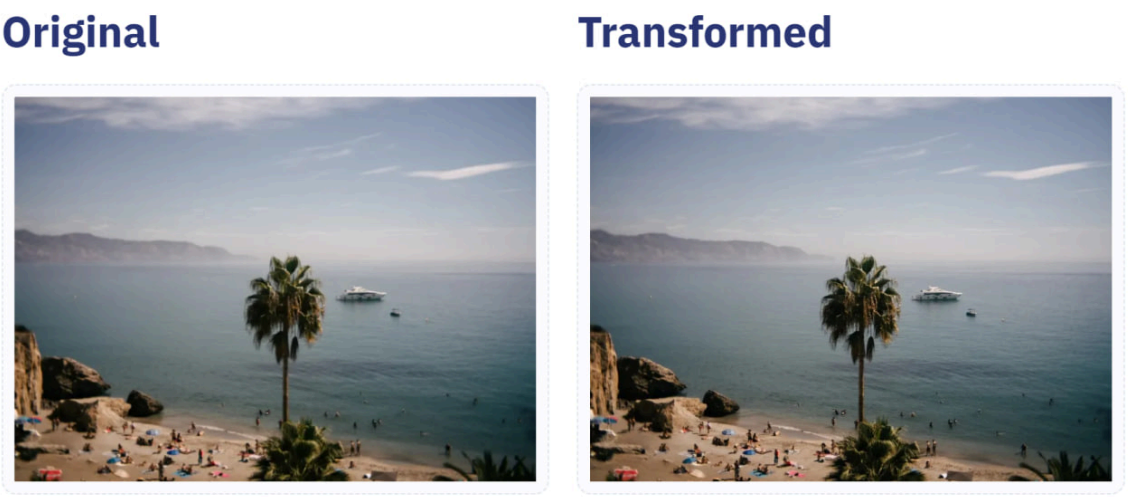


Figure 15. Final Restored Image with Preview Display

Shows the restored image with improved clarity and sharpness, completed using Cloudinary's enhancement pipeline.

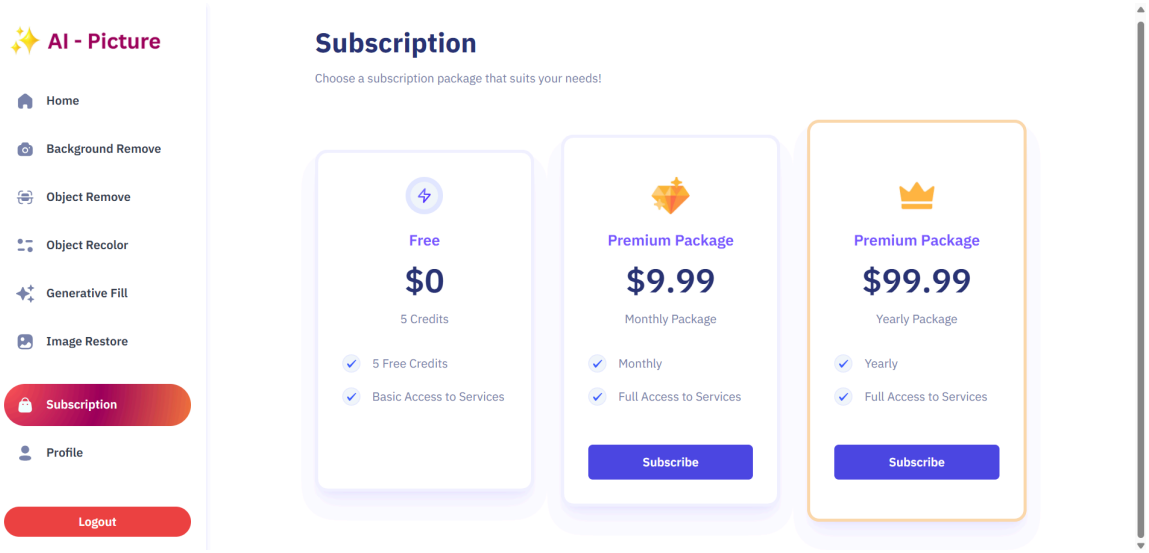


Figure 16. Available Subscription Plans (Free, Monthly, Yearly)

Illustrates the available subscription plans for free, monthly, and yearly access. Each plan outlines the feature limits and benefits.

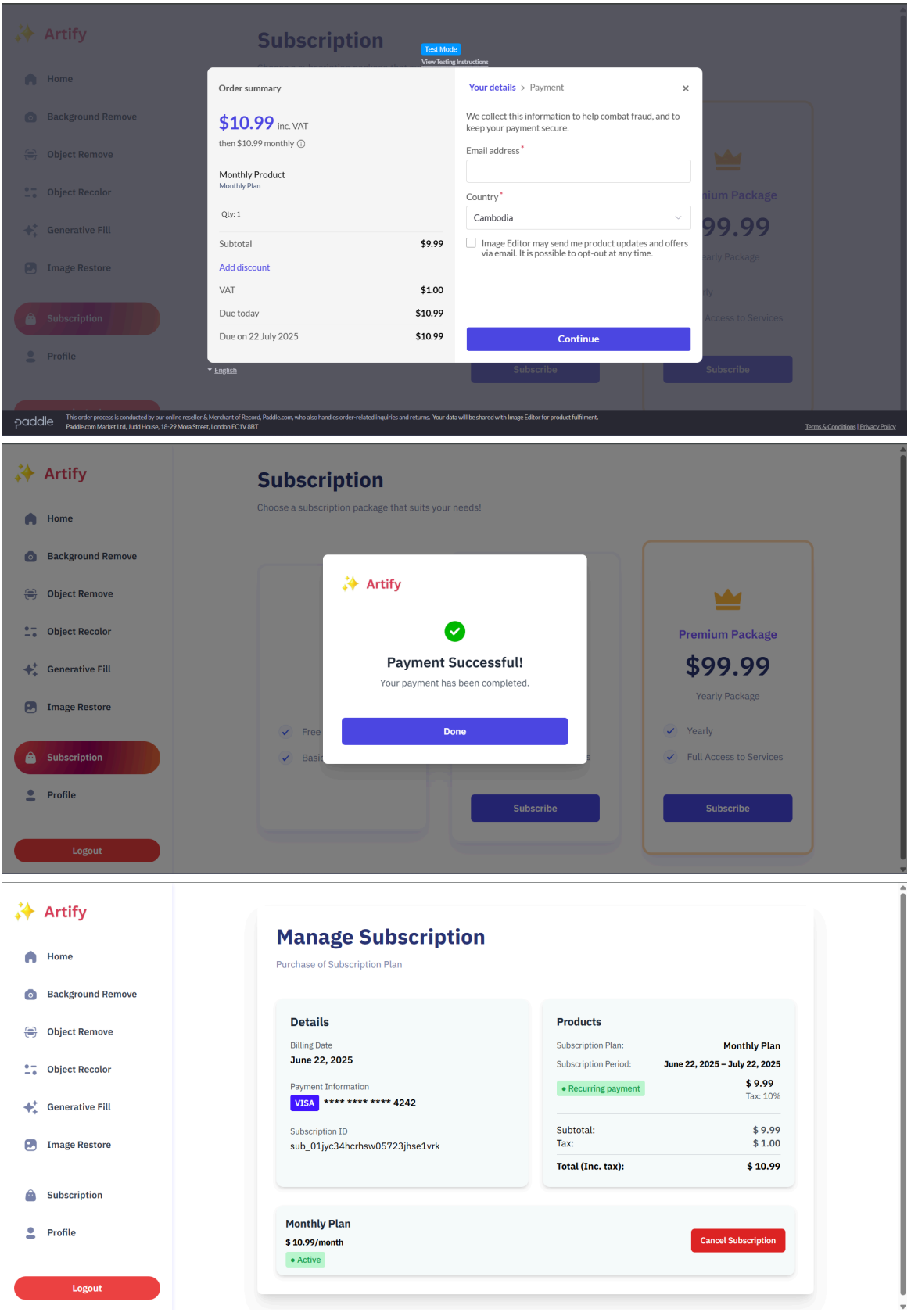


Figure 17. Payment Confirmation and Subscription Management

Captured after a successful subscription transaction. Users can review their plan, view status, and manage billing preferences.

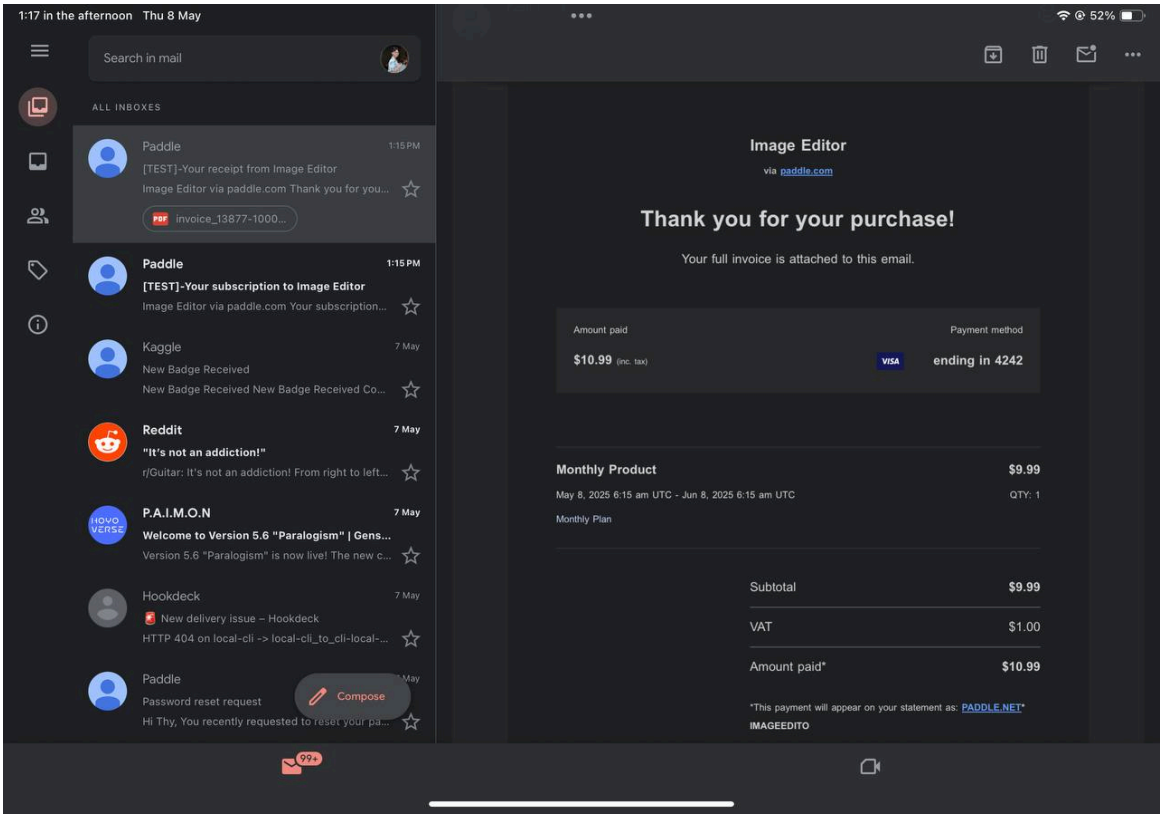


Figure 18. Subscription Confirmation Email from Paddle

Example of the confirmation email received after subscription, sent automatically by Paddle as part of the payment workflow.

4.3 Testing and Evaluation

Testing was performed across unit, integration, and user levels to validate the reliability and performance of each module.

4.3.1 Test Result

Individual components (image upload, background removal, subscription status check) were tested using mocked API responses and dummy data. Test cases verified:

- Image validation and size check
- Subscription level access
- Paddle webhook responses

4.3.2 Test Result Integration Testing

- Ensured end-to-end flow from image upload to download and billing activation.

- Tested with different subscription states (free, monthly, yearly) to evaluate access logic.
- Verified backend API integration with Cloudinary and Paddle using production keys in a staging environment.

4.3.2 User Testing

Five test users interacted with the deployed app and provided usability feedback. Feedback included:

- Interface was intuitive
- Background removal was fast (~1 to 2s)
- Subscription upgrade was seamless via Paddle
- Desire for image history panel (future enhancement)

4.4 Design Evaluation

Design criteria like modularity, scalability, and user experience were used to assess the system architecture, user interface, and data handling processes.

Table 4. Design Evaluation

Design Aspect	Evaluation
Modularity	Maintainability and future extensibility are guaranteed by the division of the frontend (Next.js), backend (Node.js), and APIs.
Scalability	Managing heavier workloads without re-architecting core logic is made possible by the use of Cloudinary and Paddle.
Security	Secure routes and token validation are used by API endpoints; Paddle securely manages user roles and payments.
Performance	Editing time per image is less than two seconds on average. Real-time results are provided by Cloudinary's CDN and transformations.
User Interface	The user experience was improved by minimalist design, responsive layout, and

	simple interaction flow.

4.5 Observations

In every test scenario, the system operated dependably. Overall responsiveness was enhanced by Cloudinary's CDN and transformation APIs, which produced quick results with little delay. Real-time subscription status updates without any access control errors were guaranteed by Paddle's webhook integration. During concurrent uploads, the backend's performance remained consistent, suggesting that it is ready for moderate production use. MySQL ensured accurate records by consistently logging user actions without losing any data. The majority of user reviews praised the platform's usability, though some suggested including a "transformation history" feature for convenience.

CHAPTER 5

DISCUSSION

5.1 Challenges and Findings

A number of architectural and technical difficulties arose during the creation of the Image Editor Platform (Artify). Careful webhook handling was necessary for integration with third-party services like Cloudinary and Paddle in order to guarantee precise user state synchronization. Another significant challenge was maintaining low-latency performance while managing real-time AI image processing (e.g., generative fill, background removal). In spite of these difficulties, the system showed responsiveness and stability in scenarios involving concurrent usage.

The platform's effectiveness in offering quick image changes without compromising output quality is one of the main conclusions. While Cloudinary's infrastructure greatly increased image delivery speeds, the modular backend architecture helped ensure smooth integration and scalability. Additionally, Paddle's subscription system managed plan entitlements and access control with success under test conditions.

5.2 Interpretation of Results

The platform achieved its functional goals, which included user-friendly interfaces, fast AI-powered transformations, and safe subscription management. Testing results validated real-time user role changes, smooth data logging, and accurate transformation task execution, as illustrated in Chapter 7. The platform's usability was confirmed by the observed user feedback, which showed that most users found the system to be effective and intuitive. Requests for features like "transformation history," however, point to possible areas where the user experience could be improved.

5.3 Limitations

Notwithstanding its effective deployment, the system has a number of drawbacks. One significant limitation is the AI model's performance; given Cloudinary's current AI capabilities, some complex image types, like those with overlapping objects or fine details, may produce inconsistent transformation results. The platform's reliance on third-party APIs, specifically Cloudinary and Paddle, is another drawback. Because any service outages, API deprecations, or modifications from these providers could have an

immediate effect on platform functionality, this dependence poses possible risks. Additionally, users on the free tier have a limited experience and may be less engaged because the subscription model limits access to advanced editing features. The system has not yet been tested in high-traffic, continuous production settings, where more performance or scalability bottlenecks might appear, even though testing showed that the backend could manage concurrent usage efficiently.

5.4 Lessons Learned and Future Improvements

The advantages of using a modular architecture and utilizing outside services to expedite the implementation of intricate features were brought to light by the creation of the Image Editor Platform. The effort needed for AI-based editing was greatly decreased by Cloudinary's extensive collection of image transformation tools, which proved to be both strong and developer-friendly. Similarly, Paddle's webhook-based subscription system made it easier to manage plans and control access, which allowed for the seamless integration of premium features.

A number of improvements are anticipated in the future to enhance both functionality and user experience. The inclusion of a transformation history feature, which would enable users to review and redownload their earlier edits, is one important area. Using offline queuing and retry logic to handle unsuccessful requests during connectivity problems or external API outages is another way to improve system resilience. In order to support more varied tasks like image captioning, resolution upscaling, and artistic stylization, the platform might also profit from incorporating more AI models. Lastly, to help users have a more knowledgeable and engaging experience, an advanced analytics dashboard will be created to provide information about their usage patterns, favorite tools, and storage usage.

5.5 Comparative Analysis

Table 5. Comparative Analysis

Platform/ Feature	**Image Editor Platform (Artify)**	**Remove.bg**	**Pixelcut Generative Fill**	**Imaginify Image Generation**
Primary Function	Comprehensive Image Editing	Background removal	Generative Fill via Text	Image Generation & Editing

AI Capabilities	Multi-Tool AI Editing	Deep Learning	Text-to-Image Editing	Text-to-Image Generation
User Interaction	Upload & Select Tools	Upload & Auto Process	Upload, Select, Prompt	Text Input & Refinement
Customization Level	High	Low	High	High
Ideal Use Cases	Restoration, Enhancement	E-commerce, Photography	Social Media, Marketing	Design, Presentations
Limitations	Performance varies with complexity	Limited to background removal	Prompt clarity dependent	Requires specific descriptions

This analysis shows that Artify offers a more comprehensive and flexible toolset compared to some niche competitors, positioning it as a robust choice for creative professionals and casual users alike.

CHAPTER 6

CONCLUSION

6.1 Summary

The potential of incorporating AI technologies into a cloud-based image editing solution has been effectively illustrated by the Image Editor Platform (Artify). For those who need generative fill, restoration, recoloring, and image background removal, the system provides a smooth experience by utilizing Cloudinary for AI-powered changes and Paddle for subscription management. The platform supports scalability and performance under concurrent tasks in addition to emphasizing ease of use. The overall findings confirm that Artify is a useful tool for artistic and expert image editing, despite issues with third-party dependencies and AI limitations.

6.2 Future Works

Building on the current achievements, several improvements and expansions are planned for future development. These include adding new editing features such as image cropping, upscaling, drop shadow, and advanced blur effects. A generative replace tool is also proposed to allow targeted object substitution using AI. Enhanced image enhancement options and deeper integration of text-to-image and image-to-image models will further improve output quality and customization. The platform will also explore support for video content, allowing users to apply similar AI transformations to video frames.

To reduce reliance on external services and gain greater control over AI capabilities, a major future direction involves developing a custom AI service architecture using **MinIO** for scalable object storage (MinIO, Inc., 2025). This transition will enable more secure, flexible, and cost-efficient AI processing directly within the platform's infrastructure**.**

6.3 Final Remark

In conclusion, Artify represents a forward-looking approach to intelligent media editing. It bridges the gap between traditional image processing and modern AI capabilities by offering a flexible, accessible, and powerful editing platform. With continuous improvements, greater independence through in-house AI infrastructure, and

expansion into multimedia, Artify is well-positioned to evolve into a comprehensive creative ecosystem that empowers users with efficiency, control, and innovation.

REFERENCES

Canva Pty Ltd. (2025). *Canva AI photo editor* [Internet]. Retrieved May 30, 2025, from https://www.canva.com/features/ai-photo-editing/

Cloudinary Ltd. (2025a). *Cloudinary* [Internet]. Retrieved May 28, 2025, from https://console.cloudinary.com

Cloudinary Ltd. (2025b). *Generative AI for complex image editing* [Internet]. Retrieved May 30, 2025, from https://cloudinary.com/blog/generative-ai-for-complex-image-editing-at-scale-in-cloudinarys-latest-innovations

Cloudinary Ltd. (2025c). *Generative AI transformations* [Internet]. Retrieved May 30, 2025, from https://cloudinary.com/documentation/generative_ai_transformations

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*. Retrieved May 30, 2025, from https://internationalconferenceonlearningrepresentations

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved May 30, 2025, from https://www.deeplearningbook.org/

Kaleido AI GmbH. (2025). *Remove.bg* [Internet]. Retrieved May 30, 2025, from https://www.remove.bg

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*. Retrieved May 30, 2025, from https://imagenetclassificationwithdeepconvolutionalneuralnetworks

MinIO, Inc. (2025). *MinIO object storage for AI workloads* [Internet]. Retrieved May 31, 2025, from https://min.io/

OpenJS Foundation. (2025). *Node.js* [Internet]. Retrieved May 28, 2025, from https://nodejs.org

Oracle Corporation. (2025). *MySQL* [Internet]. Retrieved May 28, 2025, from https://www.mysql.com

Paddle.com Market Ltd. (2025a). *Paddle* [Internet]. Retrieved May 28, 2025, from https://www.paddle.com

Paddle.com Market Ltd. (2025b). *Webhooks - Paddle developer* [Internet]. Retrieved May 28, 2025, from https://developer.paddle.com/webhooks/overview

Pixelcut Inc. (2025). *Pixelcut: AI generative fill* [Internet]. Retrieved May 30, 2025, from https://www.pixelcut.ai/generative-fill

SlideAI. (2025). *Imaginify: AI image generator & editor* [Internet]. Retrieved May 30, 2025, from https://imaginify.slideai.net/

Vercel Inc. (2025). *Next.js* [Internet]. Retrieved May 28, 2025, from https://nextjs.org

APPENDICES

Appendix A: Image Upload and Transformation Results (Before and After)

This appendix presents a collection of screenshots that demonstrate the before and after results of AI-powered image transformations performed using the Artify platform. Each image has been processed using Cloudinary's transformation APIs, showcasing the core functionalities implemented.

Background Removal

The original image included a subject with a cluttered background. After processing, the background was automatically removed, leaving only the main subject cleanly isolated.

Figure 19. Background Removal Before and After

The figure shows an original image with a background and the result after AI-powered background removal using Cloudinary.

Object Removal

The initial image featured an unwanted object in the scene. After transformation, the object was seamlessly removed, and the surrounding area was realistically filled using AI-based inpainting.

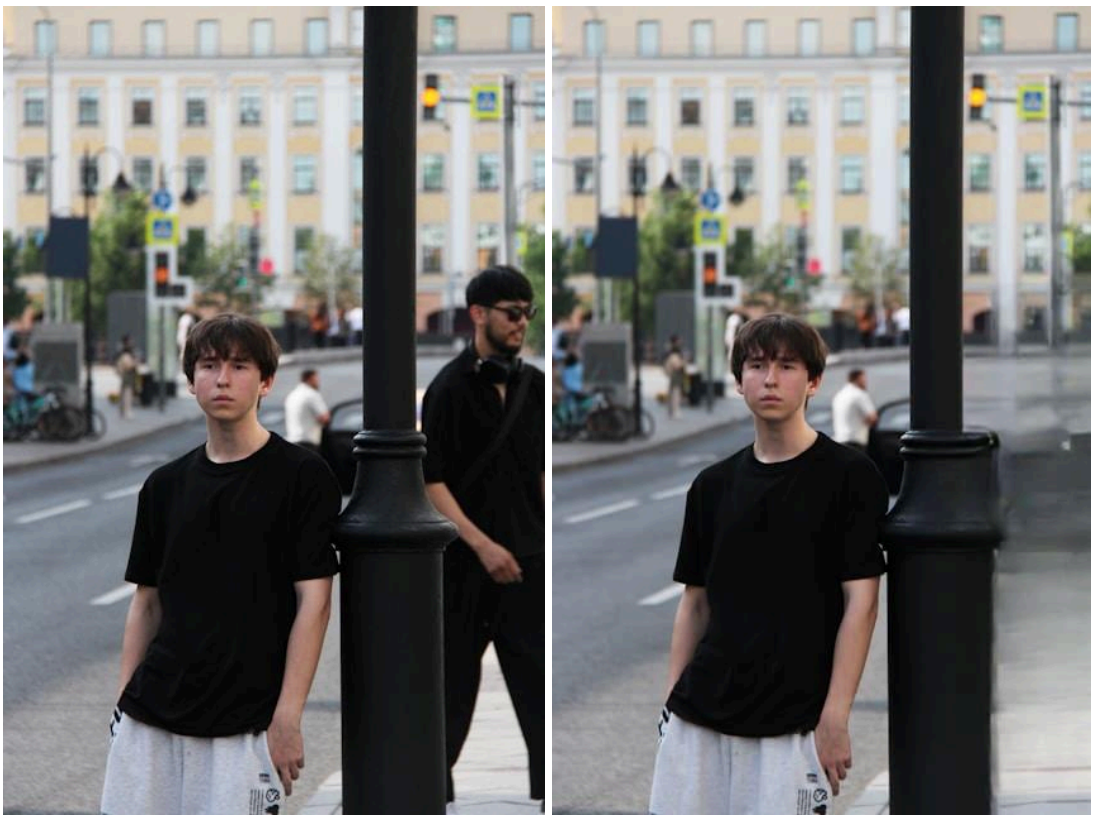


Figure 20. Object Removal Before and After

This figure illustrates how an unwanted object is eliminated from the scene and the empty space is filled naturally using AI inpainting.

Object Recoloring

In the original version, an object appeared in its default color. After applying the object recoloring tool, the color was changed to the user-specified hue while preserving the object's shape and texture.

Figure 21. Object Recoloring Before and After

Displayed here is the transformation of an object’s original color into a new shade, guided by user input and executed via Cloudinary's recoloring API.

Generative Fill

The input image had a blank or cropped region. Once processed, the system generated new content to fill the missing area in a visually coherent and context-aware manner.

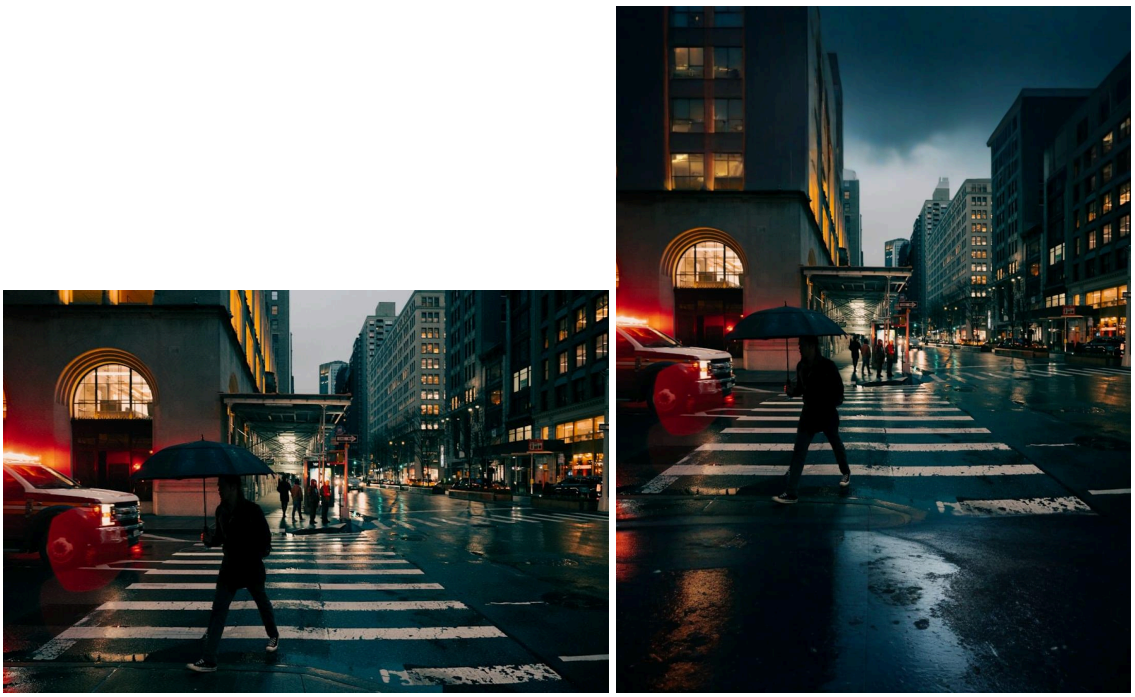


Figure 22. Generative Fill Before and After

The figure depicts how missing or blank regions in the image are filled with new, AI-generated content based on surrounding context.

Image Restoration

The input image had a blank or cropped region. Once processed, the system generated new content to fill the missing area in a visually coherent and context-aware manner.

Figure 23. Image Restore Before and After

This figure presents the improvement of a low-quality or damaged image after restoration, showing enhanced clarity and sharpness.

Appendix B: Failed Transformations from Incorrect Prompts

Misleading Object Removal Prompt

The input image contained no animals, but the prompt "remove animal" was used. The AI mistakenly removed the human subject instead, demonstrating a limitation in prompt interpretation and semantic recognition.

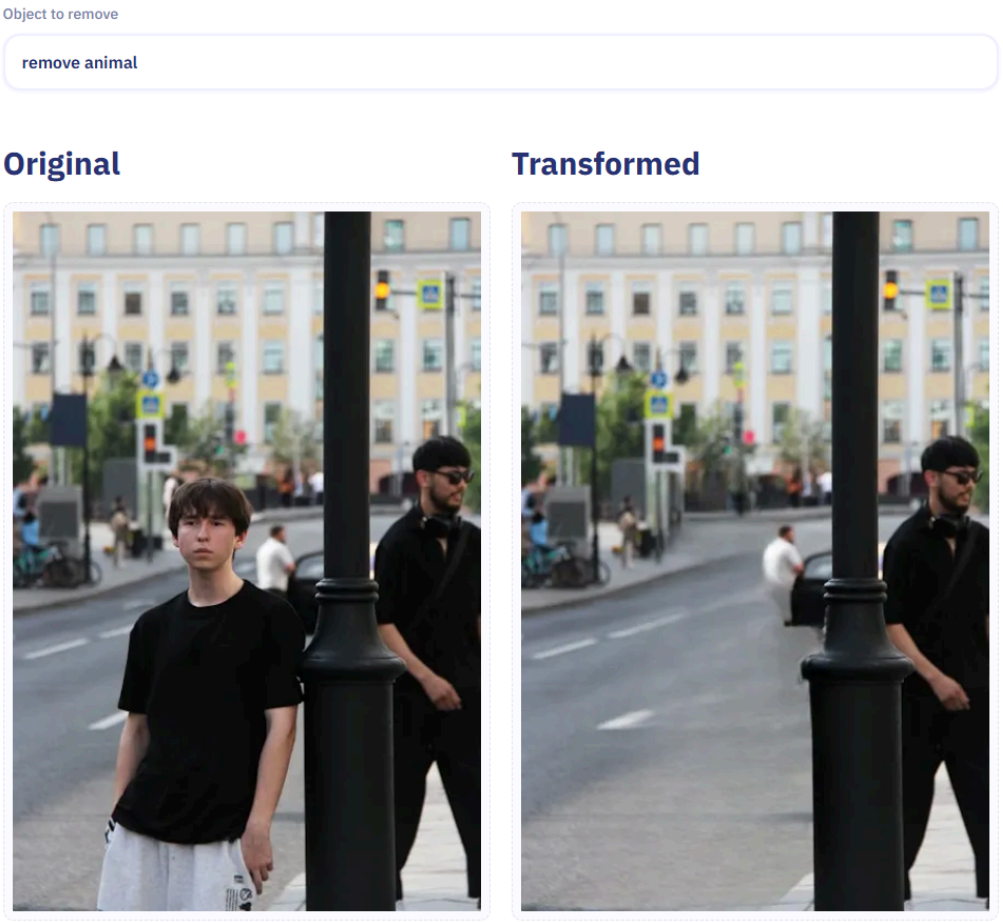


Figure 24. Incorrect Object Removal Caused by Misleading Prompt

The prompt "remove animal" was used; however, the image did not contain any animals. As a result, the system mistakenly removed the human subject instead, highlighting a limitation in semantic understanding and prompt validation.

Incomplete Object Recoloring Prompt

In this case, the user prompted the system to "recolor red shirt". While the AI correctly identified the color red in the image, it failed to execute the transformation and returned the original image unaltered.

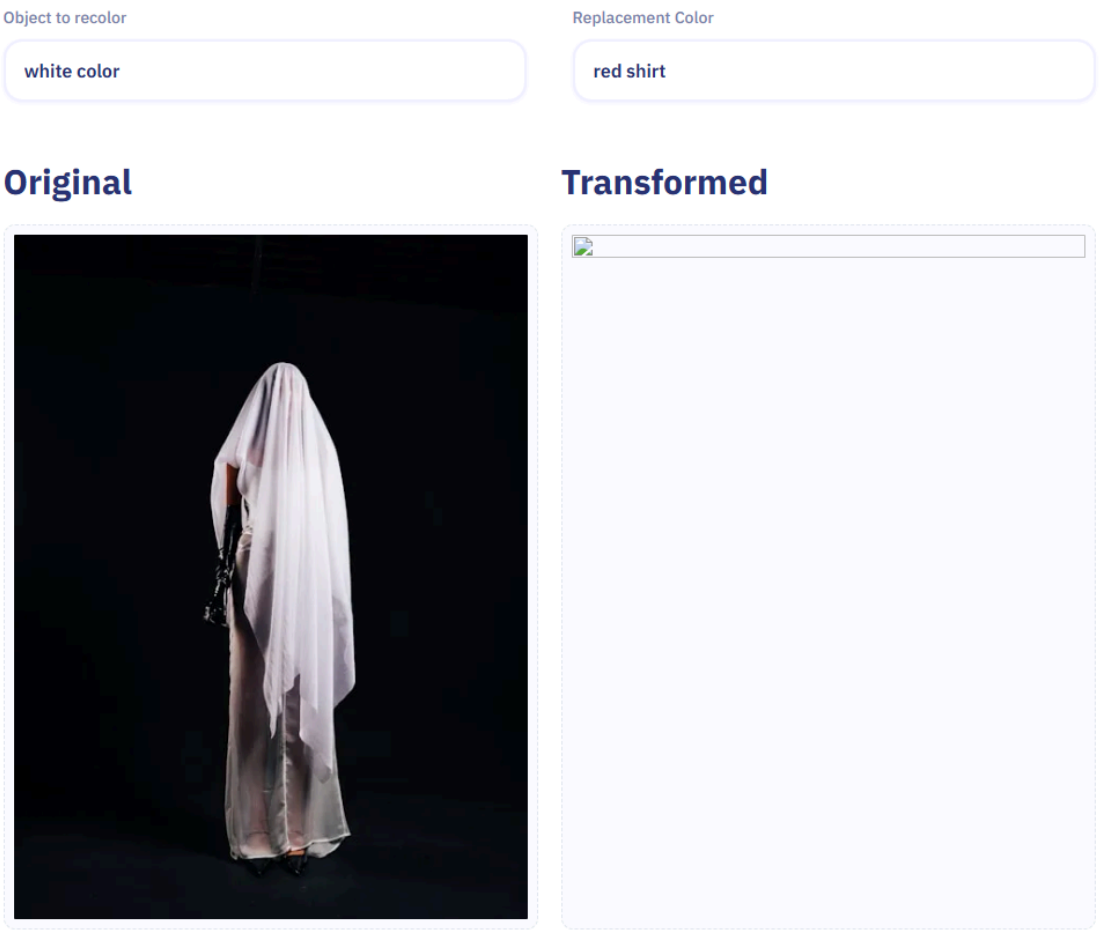


Figure 25. Failed Object Recoloring Due to Prompt Misinterpretation

Despite recognizing the red shirt, the system did not apply any recoloring, highlighting a weakness in connecting object detection with action execution.

Appendix C: Cloudinary Method

This appendix presents the core methods and AI models used by Cloudinary to enable advanced image transformations in the Artify platform. Each transformation is powered by specific machine learning techniques optimized for cloud-based real-time editing.

Table 6. Cloudinary Method

Feature	**Method**	**Model Type**
Remove Background	Semantic segment	U-net, Mask R-CNN

Remove Object	Inpainting	LaMa, Contextual Attention GANs
Recolor Object	Segmentation + Color transform	Custom pixel replacement / ML Segmentation
Generative Fill	Diffusion model (e.g., SD)	Stable Diffusion, GANs
Restore Image	Super-resolution / denoising	ESRGAN, DeblurGAN, Autoenciders