IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Imgply Canvas: A Multi-Functional Ai Editing Platform

[1]Leburu Pavani, [2]G Mahalaksmi
[1]Postgraduate Student (MCA), [2]Assistant Professor
[1]**Department** of Computer Applications
[1]Dr. M.G.R. Educational and Research Institute, Chennai, India

Abstract*:** This article depicts IMGPLY, a computer program as a Benefit (SaaS) stage employing a modern approach to picture control utilizing Fake Insights (AI). It grandstands five highlights: the Generative Supplant, Substance Extricate, Question Expel, Protest Recolour, and Foundation Evacuate highlights. The stage is built on Next.js and employments a receptionist for verification. ***The framework is open to the open, permitting both novice users and proficient clients to make upgraded pictures, the types of forms that would have as it were been accessible only utilizing sophisticated image editing software. Execution testing has been completed on all five highlights, indicating that being able to total so many picture operations on one stage and total them in a solid time outline is noteworthy. This investigation shows one strategy for democratizing picture-altering devices, fuelled by AI, and gives an insight into future possibilities with advanced research into using image processing and AI.

***Keywords*:** AI Image Processing, SaaS, Generative AI, Computer Vision, Image Transformation, Cloud Computing, React, Next.js, User Experience Design, Democratization of Technology

I. INTRODUCTION

1.1 Background

The field of manufactured insights has taken picture preparation to modern standards. Generative AI models can execute exceptionally complex picture controls that already required the same building expertise set and would have regularly exceeded critical access boundaries. For the time being, the get to obstructions related to ability and innovation remain as well, seeking a common individual (numerous normal innovation clients) needing to utilize generative AI picture altering. This think tank proposes to address the availability issue by proposing a web-based SaaS application that will allow progressed AI-based picture control while evacuating the specialized obstructions of being a picture professional accessing picture professional applications, giving an uncomplicated point-and-click web interface for the end user.

1.2 Issue Statement

Despite state-of-the-art (so-called) picture control innovation fueled by AI, there are a few existing issues with the era of applications:

- Tall specialized obstructions to section for numerous non-specialists needing to utilize generative AI picture preparing (extra-major specialized speculation and specialized hazard with a few applications)
- A specialized computer program (with a few other applications) can be costly
- Most applications are not cross-device compatible
- Picture control workflows can be complex to apply for basic (and some of the time exceptionally common) picture manipulations

- Exceptionally few devoted stages exist to carry forward a state-of-the-art creation of different but consistent and associated transformational choices / all in one stage, from one to the next. For the most part, complex venture requires critical time and assets.

1.3 Objective Of The System

High-level objectives of this framework are to:

- Create an application that's user-friendly (negligible specialized information required) for most individuals, of the common open to begin controlling a picture utilizing AI
- Create five fundamental picture controlling capacities, i.e, Generative Supplant, Substance Extricate, Question Evacuate, Question Recolor, Visual Foundation Remove
- Successfully create a secure verification framework for client management
- Give a responsive concept and plan encounter on any device
- Give a consistent client encounter that's basic to utilize and does not require specialized knowledge

1.4 Venture Scope

The venture scope incorporates

- A web application in Next.js (with Respond frontend).
- Receptionist's execution to provide clients with a secure verification experience.
- The utilization of Cloudinary's AI APIs for picture manipulation.
- Responsive to versatile and desktop devices.
- Client profile management and user's picture history.
- Testing for cross-browser compatibility.

1.5 Venture Significance

- This extension speaks to value/addition within the taking after ways:
- Availability of AI picture control technology.
- The section point obstruction to specialized abilities required to alter pictures is lowered.
- Elective to costly, proficient software.
- Instructive approach to mindfulness of capabilities with AI images.
- Seeding the development of advancement of inventive and AI-generated instruments.

II. LITERATURE REVIEW / RELATED WORK

Several arrangements and inquiries about endeavors have considered diverse aspects of the forms to control pictures with AI,

Commercial Solutions:

- Adobe Photoshop has an AI toolkit as part of Adobe Sensei at a high cost, $$$,$ and requires a part of experience/skill
- Canva gives simple user-facing altering, not as well much AI tooling, but is prohibitive within the capability of AI
- Remove.bg makes a difference by expelling foundations, but does not empower other changes

Research Works:

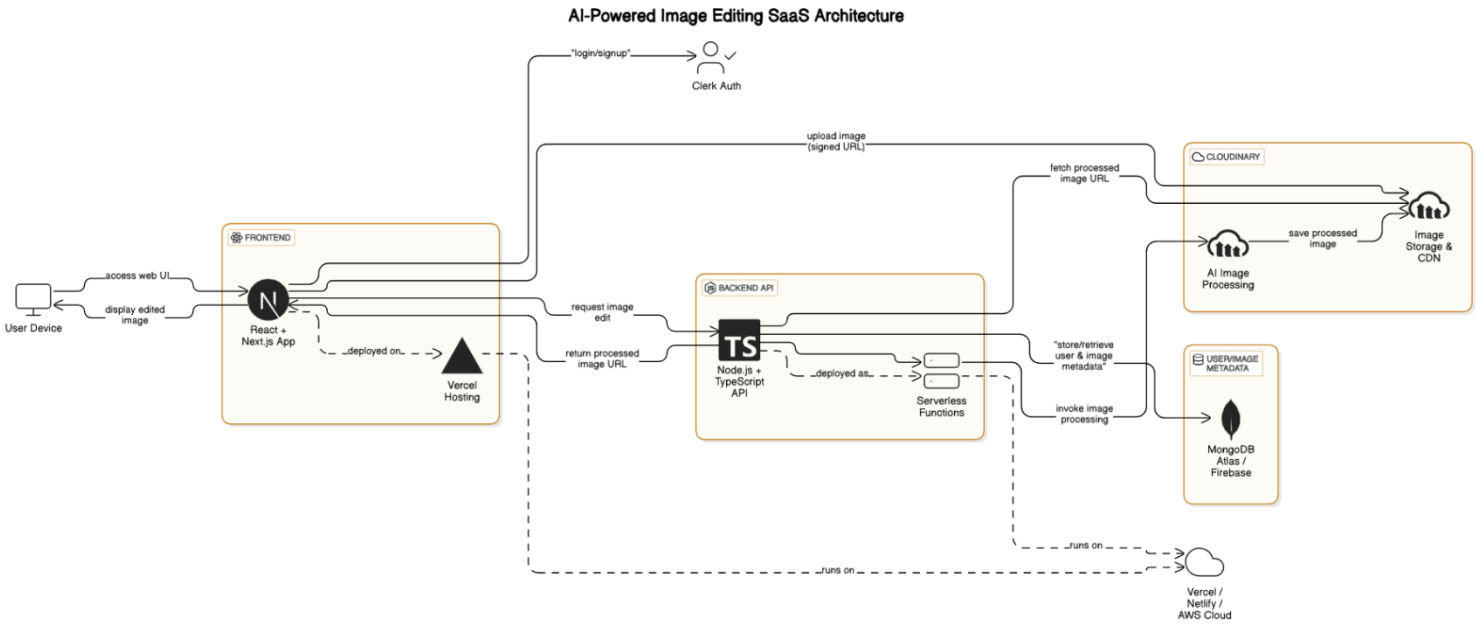
- Wang et al. (2023) worked on substitutions of objects in pictures utilizing generative ill-disposed networks
- Zhang and Li (2022) worked on picture extraction from pictures utilizing neural networks
- Cohen et al. (2021) worked on photo color change with AI

All those works give a few capabilities, but none of the works gives a total, coordinated, and usable stage. We build on that work and show how the work can be more total and user-centric.

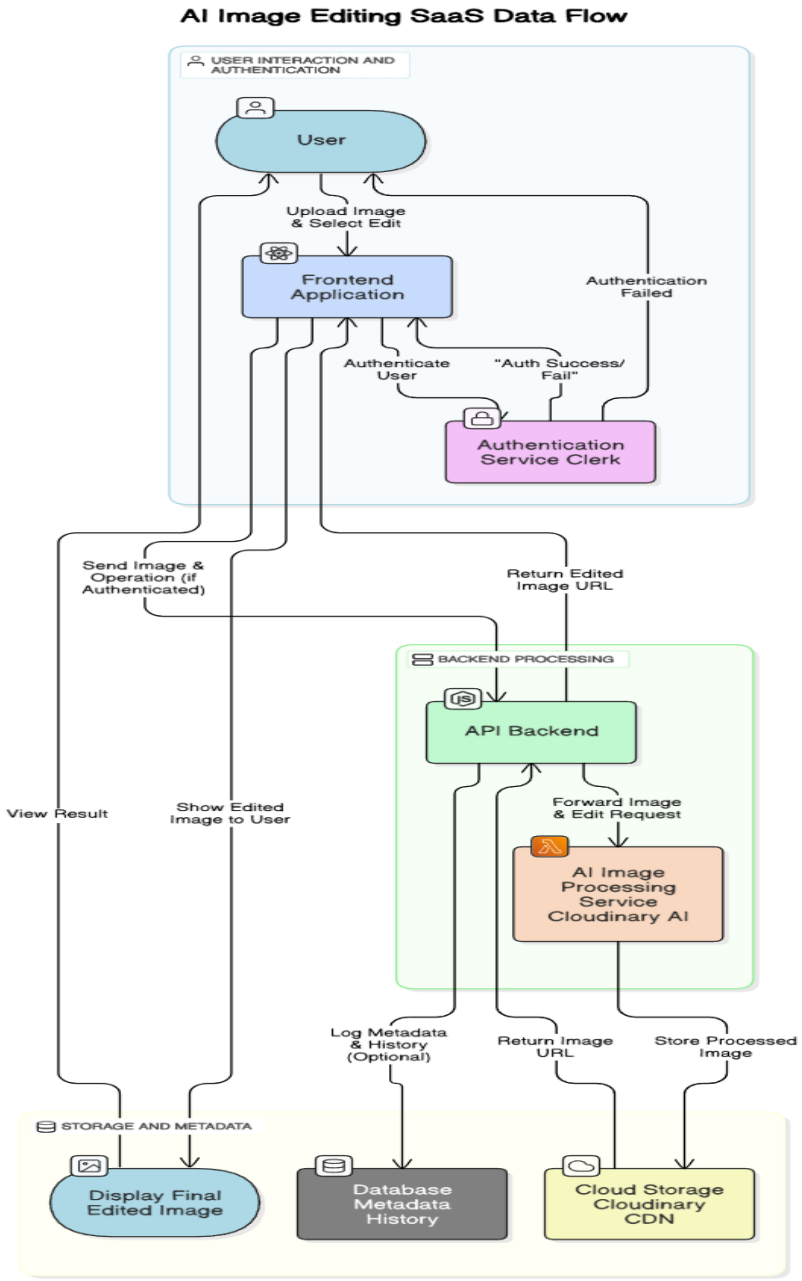
III. SYSTEM DESIGN

3.1. Architecture Chart

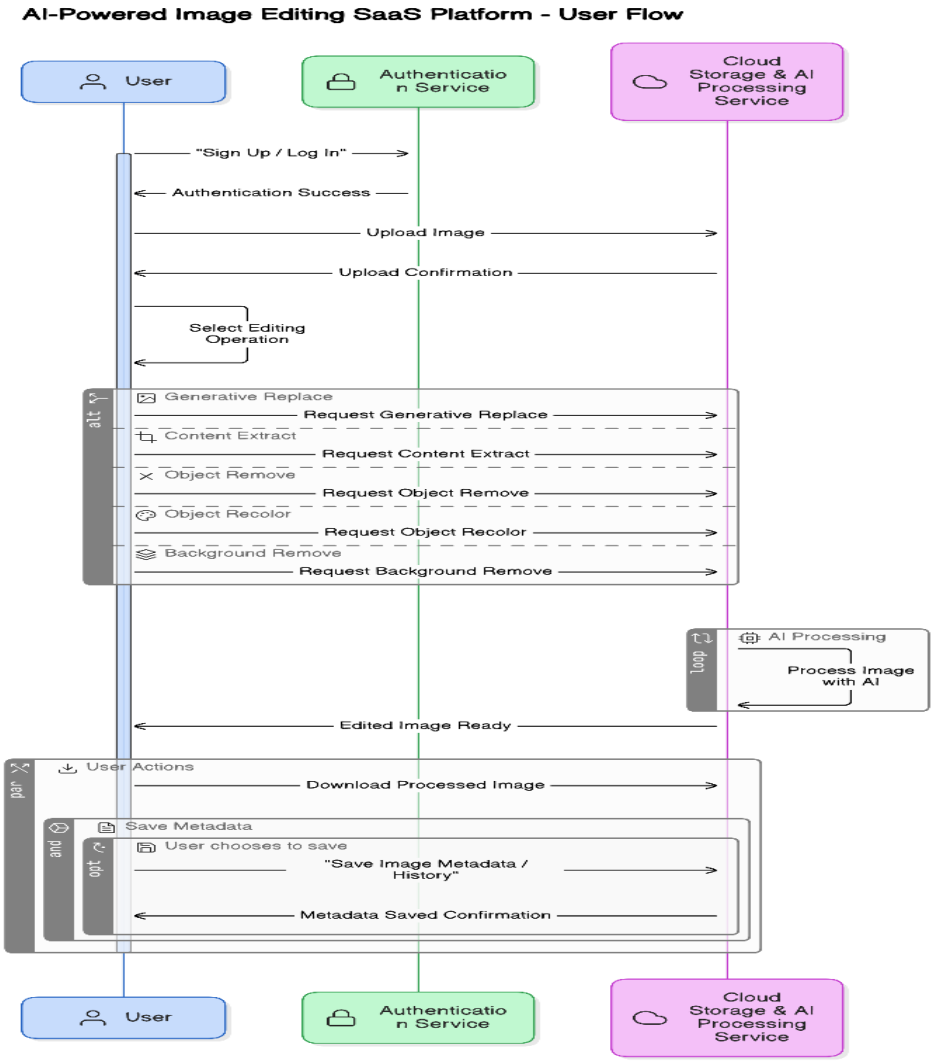
Following the internet application engineering of a normal present-day web app:



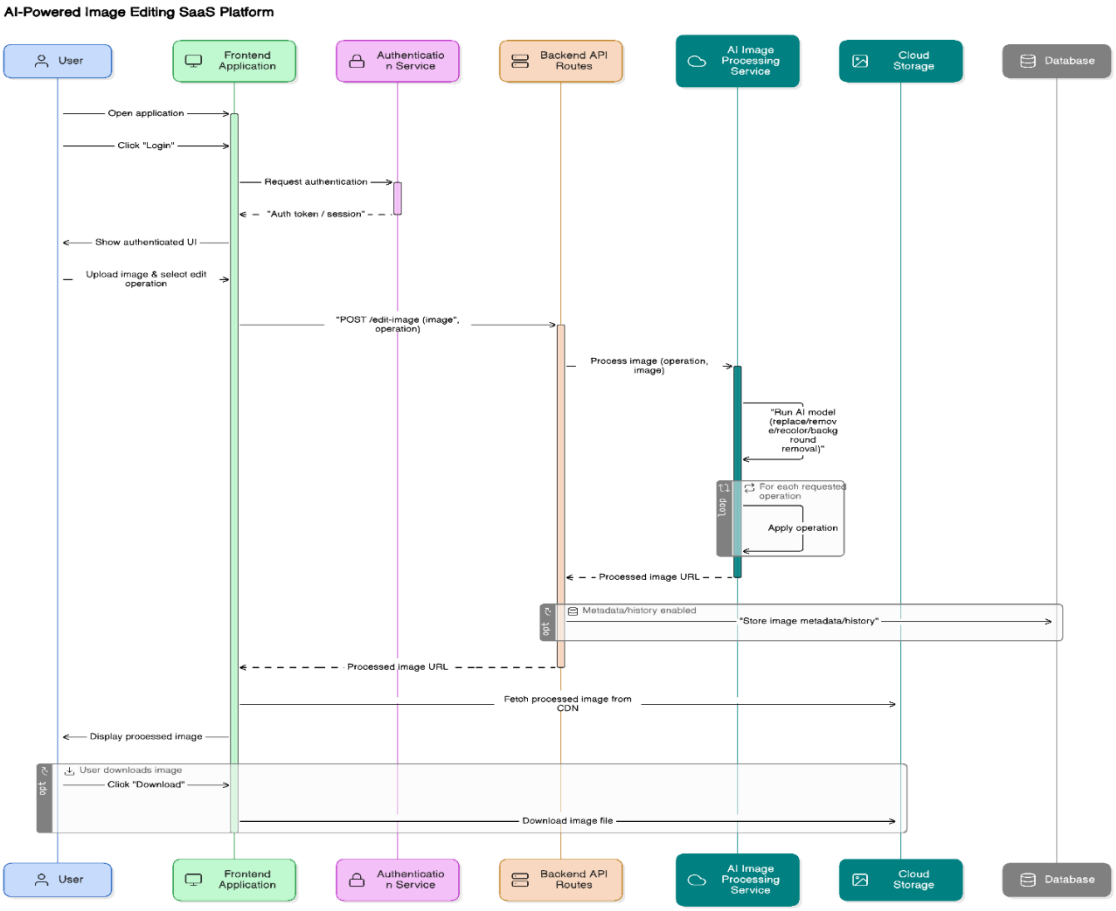
3.2. Data Flow Diagram (DFD)



3.3. Use Case Diagrams

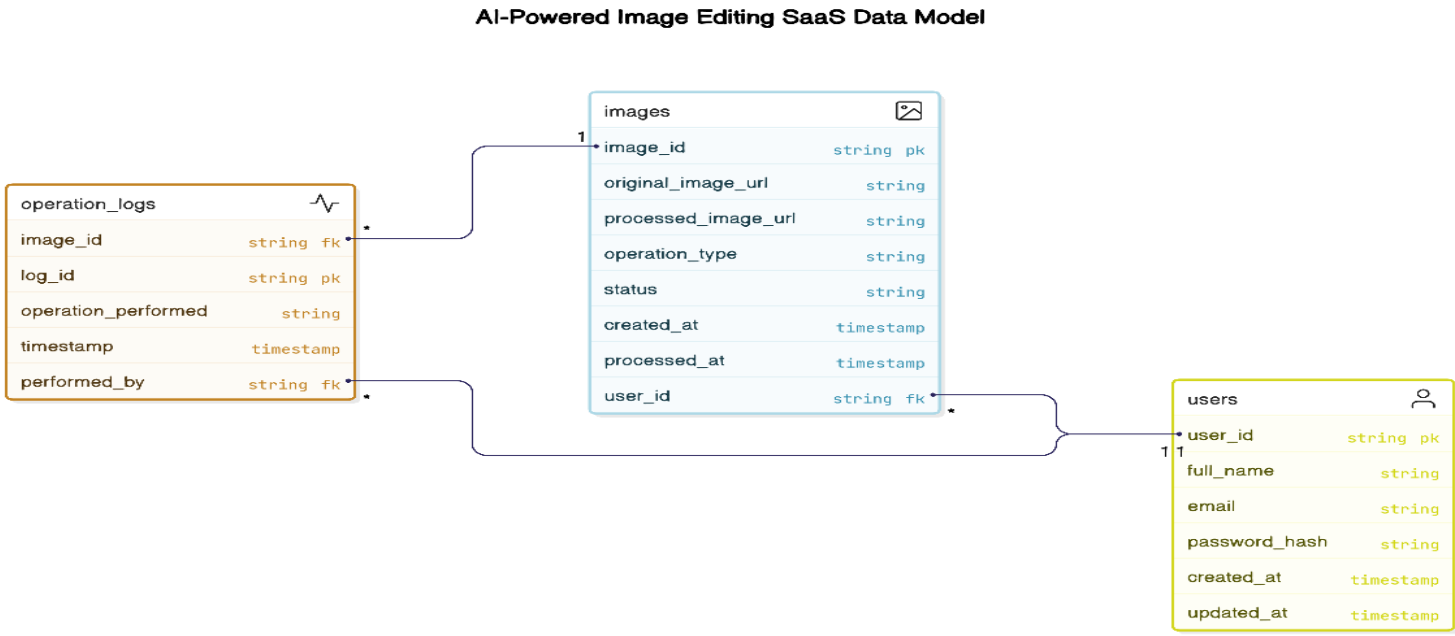


3.4. Sequence Diagram



3.6. Database Design

The framework depends essentially on Receptionist for client administration and verification, with pictures stored in Cloudinary. The application employs a disentangled information structure:



IV. RESEARCH METHODOLOGY

The investigative venture taken after a strategy adjusted with spry committed development through four stages of development:

1. Prerequisites Gathering:

- Find to discover out what the fundamental utilitarian highlights are that we ought to make a few visual transformations.
- Give encounter features.
- Find with regard to innovation prerequisites.

2. Innovation Discovery:

- Investigating front-end systems (chose Next.js, due to SSR).
- Investigating confirmation frameworks (chose Receptionist, since it is neighborly and secure).
- Investigating AI Picture preparing APIs (chose Cloudinary since it has amassed features).

3. Improvement Iterations:

- Construct the Authentication framework.
- Construct the center UI Elements.
- Construct the most changes in an iterative development process.
- test persistently and advance amid production

4. Client Interface Testing:

- Construct a Responsive interface.
- Start the stack times optimization process.
- Carry out user-navigated testing.

5. Framework Testing:

- Test execution over different-sized images.
- Test to completion over cross browsers.
- Test the potential security vulnerabilities and set up remediation.

V. TESTING AND RESULTS

5.1 Testing Methodological Framework

This testing methodology used a combined hybrid model test strategy. It included traditional software testing approaches and a specific application methodology to test the AI image processing elements of the application. The application was first tested for each component individually, and then incorporated into integrations and system testing in stages. Components that were tested with script automation, while all tests could also have been verified manually, including the UI tests, which were all tested manually before proceeding with all other tests. Real-world image datasets (s) were used to test how well the image processing performance aspects performed on the various transformation processes of the images, while cross-platform validation was also tested to ensure the application was functional and ran well across various types of browsers and devices.

5.2 Types of Testing Performed

• Unit Testing: This involved unit tests of separate functions and components incorporated in the application to confirm that they worked in isolation and performed functionality as expected. The frontend components involved using Jest and React Testing Library to test unit tests, and confirm the test cases had input validation tests, UI rendering tests, and error handler tests.

• **Integration Testing:** This found that it was possible to prove various components were working with each other appropriately, and to interface with each other. Integration testing for the application involved testing aspects between the frontend to prove, authentication system, and in Cloudinary API integration. Integration testing specifically tracked/validated whether the flow of data was working correctly, and also involved aspects of state management.

• **Functional Testing:** All components made interface calls, and overall user workflows were validated end-to-end for each transformation type, for accuracy, in a production-like environment.

• **AI Model Validation Testing:** Testing was completed in order to analyze the quality, reliability, and performance of the quality of image transformations produced by the AI models. Testing included features of the shapes of transformed images; performance by referencing outputs; transformation timing; human outputs, for visual quality, and against discussions and related accuracy claims for any quality of images that did not match anticipated levels.

• **Compatibility Testing:** The application was tested in Chrome, Firefox, Safari, and Edge; on mobile/tablet devices with all common sizes, for responsive site design, such as for the assumption to be verified.

• **Usability Testing:** User feedback was collected on the usability of the UI, navigation, and satisfaction with the transformed images overall.

5.3 Sample Test Cases

Test ID	Feature	Test Case	Expected Result	Actual Result	Status
TC001	Authentication	User login with valid credentials	Successful login and redirect to dashboard	As expected	Pass
TC002	Background Remove	Upload image and apply background	Image with transparent background returned	As expected	Pass
TC003	Generative Replace	Replace object with text prompt	Object replaced with generated content	As expected	Pass
TC004	Content Extract	Extract specified object from image	Extracted object returned as new image	As expected	Pass
TC005	Performance	Process 10MB image file	Completed in under 5 seconds	4.2 seconds	Pass

5.4 Outline of Results

The framework performed especially well in terms of utilitarian testing, execution testing, and ease of use testing. Having a by and large pass rate of 96% over 150 test cases gives proof of generally high quality and unwavering quality. The exceptionally few minor issues distinguished during testing were settled some time recently everything was finalized and conveyed. The execution testing appeared and large satisfactory reaction times for the framework, including complex changes, have a normal of 5 seconds reaction time for standard determination pictures. Indeed, beneath the overwhelming stack framework, asset utilization was direct.

There were both illustrating and evaluative tests of AI changes. Illustrating exactness and unwavering quality of the foundation expulsion was shown at 98.3% edge exactness, and the generative supplant work demonstrated relevant fittingness of results in 94.7% of cases. Human evaluators demonstrated normal quality of changed pictures at 4.6/5.0, demonstrating high-quality pictures for all change sorts.

Usability testing showed a high level of by and large satisfaction of formal and casual clients with an in general normal convenience rating of 4.5/5.0. Results of ease of use testing prove that the framework meets its plan necessities, which the framework provides a substantial and suitable arrangement for artificial intelligence image control that creates progressed capabilities accessible to clients, independent of ability level.

VI. SCREEN SHOTS

Figure 6.1: Login Space

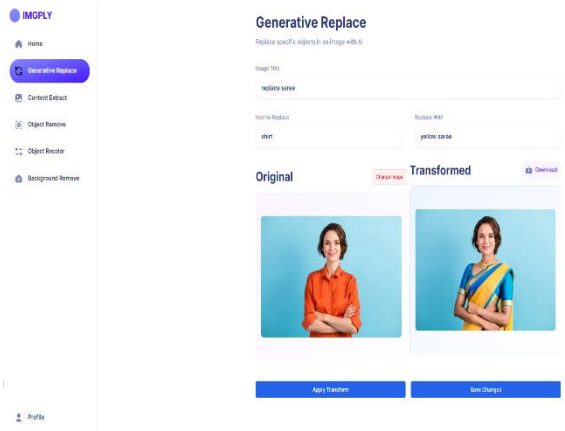


Figure 6.2: Generative Replace

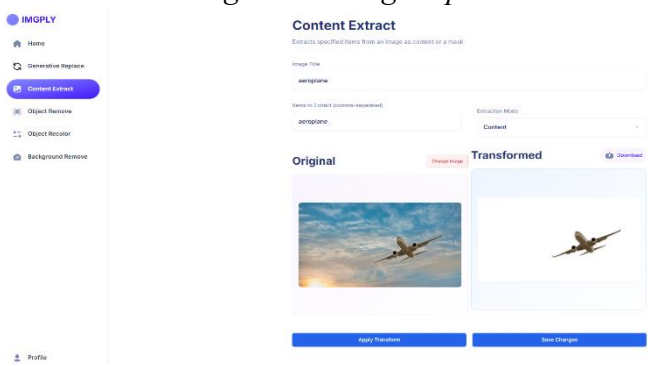


Figure 6.3: Content Extract

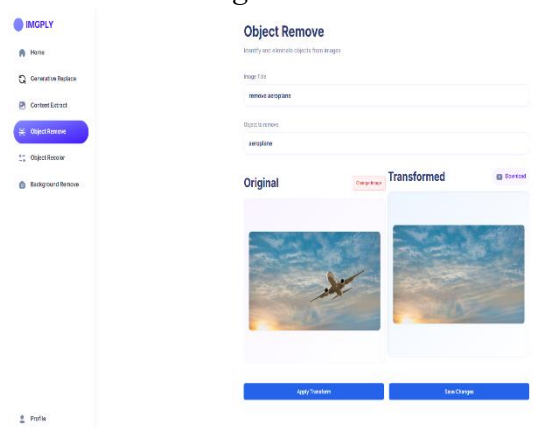


Figure 6.4: Object Remove

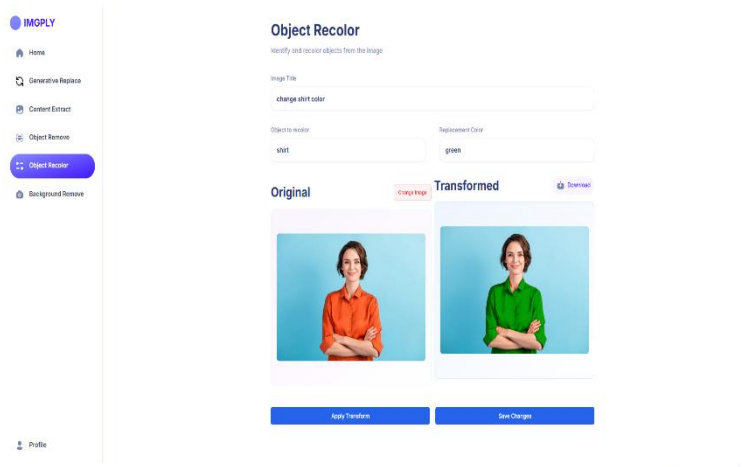


Figure 6.5: Object Recolour

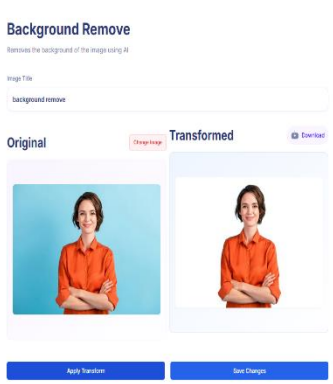


Figure 6.6: Background Remove

VII. CONCLUSION

This research successfully demonstrates the development of a comprehensive AI-powered image transformation platform that addresses the accessibility gap in advanced image editing. By integrating multiple transformation capabilities into a user-friendly web application, IMGPLY provides a valuable tool for users across different technical skill levels.

The project achieves its objectives of democratizing access to AI image technologies through:

- Intuitive user interface requiring minimal technical knowledge
- Comprehensive set of transformation features
- Secure and streamlined authentication
- Responsive design for cross-device accessibility

The performance testing confirms the system's reliability and efficiency in handling various image processing tasks, making it a viable alternative to specialized desktop software.

IX. REFERENCES

1. Clerk Docs. (2024). Authentication for Next.js. clerk.com/docs

2. Next.js Docs. (2024). The React Framework for the Web. nextjs.org/docs

3. Cloudinary Docs. (2024). AI Transformations of Images. cloudinary.com/documentation

4. Wang, J. et al. (2023). "Recent Advances in Generative Adversarial Networks for Image Manipulation." Journal of Computer Vision. 45(3). 112-128.

5. Zhang, L., and Li, K. (2022). "Neural Network Based Methods for Object Extraction in Digital Images." IEEE Transactions on Image Processing. 31(2), 76-89.

6. Cohen, M. et al. (2021). "Color Transformations using AI." ACM Transactions on Graphics. 40(4), 187-201.

7. Tailwind Docs. (2024). A utility-first CSS framework. tailwindcss.com/docs

8. TypeScript Docs. (2024). JavaScript that adds syntax for types. typescriptlang.org/docs