e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

IMAGINIFY - AI-POWERED IMAGE ENHANCEMENT & TRANSFORMATION SAAS WITH NEXT.JS 14, CLOUDINARY AI, CLERK, AND STRIPE

Tejashwani Pathak*1, Soumya Chandrakar*2, Pragati Sahu*3,
Amita Choudhary*4, Prof. Swati Sahu*5

*1,2,3,4Student, Computer Science And Engineering, Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India.

*5Professor, Computer Science And Engineering, Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India.

ABSTRACT

With their constantly expanding capabilities, computers have taken over the majority of jobs performed by humans in the modern day. Photographs are snapshots of life's events, but environmental elements deteriorate them with time, making it crucial to preserve these priceless memories. Motion blur and misfocus are two issues that arise during image restoration, a procedure that varies in difficulty depending on the degree of degradation. Backlogs result from the time-consuming nature of manual repair. We need to handle a variety of degradations in ancient images, such as noise, blurriness, dust spots, and scratches, in order to improve restoration. Moreover, additional techniques like facial refinement networks might be needed to restore minute facial characteristics. The project's goal is to create an image restoration system that can handle various types of noise and remove it to create clear, restored photos. Online image enhancement has been changed by the introduction of AI into web development. One example of this is the cutting-edge SaaS platform Imaginify. By integrating Clerk, Stripe, Cloudinary AI, Next.js 14, and more, Imaginify provides customers with robust image editing capabilities that are available via online browsers. This article emphasizes the importance of Imaginify in AI-driven picture enhancement by thoroughly examining its architecture, features, market potential, and future prospects. Imaginify is a paradigm shift in online image editing accessibility, with consequences for sectors dependent on compelling digital material. It does this by fusing cutting-edge AI with user-friendly web tech.

Keywords: Artificial Intelligence (AI), Software As A Service (Saas), Image Restoration, Next.Js, React.Js, Tailwind CSS, Clerk Authentication, Stripe Payment Processing, Mongodb Atlas, Cloudinary AI API.

I. INTRODUCTION

Overview:

In the contemporary digital landscape, the demand for high-quality images permeates various online platforms, ranging from social media to e-commerce websites and digital marketing campaigns. However, the process of obtaining and maintaining visually appealing images often poses significant challenges, requiring sophisticated editing tools and expertise that may be inaccessible to many users. Traditional image editing software, characterized by complex workflows and steep learning curves, further compounds these challenges, limiting usability and accessibility.

In response to these obstacles, the emergence of Imaginify represents a pioneering solution in the realm of image enhancement and transformation. Imaginify stands as an AI-Powered Image Enhancement & Transformation Software-as-a-Service (SaaS) platform, meticulously crafted on Next.js 14, with integration of Cloudinary AI, Clerk, and Stripe. This amalgamation of cutting-edge technologies aims to democratize image editing, offering users a user-friendly platform to effortlessly enhance their visual content.

Technical Architecture:

Imaginify is built upon Next.js 14, a popular React framework known for its performance and scalability. Next.js provides server-side rendering capabilities, enabling fast page loads and optimal SEO performance. The frontend of Imaginify is developed using React components, ensuring a responsive and interactive user interface.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

Cloudinary AI serves as the backbone of Imaginify's image processing capabilities. Leveraging state-of-the-art AI algorithms, Cloudinary AI enables functionalities such as image enhancement, background removal, object removal, object recoloring, and generative fill. These features empower users to transform their images with unparalleled precision and efficiency.

Clerk is integrated into Imaginify for user authentication and management. With Clerk, users can securely create accounts, log in, and manage their profiles. Additionally, Clerk's authentication APIs facilitate seamless integration with third-party services, ensuring a smooth user experience.

Stripe powers Imaginify's payment processing and credits system. Users can purchase subscription plans or credits to access premium features and services. Stripe's robust payment infrastructure ensures secure transactions and compliance with industry standards.

Key Features:

Image Enhancement: Enhance image quality and clarity using AI-driven algorithms.

Background Removal: Easily remove backgrounds from images for professional results.

Object Removal: Erase unwanted objects from images with precision and accuracy.

Object Recoloring: Change the color of objects in images to match specific preferences or branding requirements.

Generative Fill: Automatically fill in missing parts of images for seamless results.

User Authentication: Securely authenticate users and manage user profiles with Clerk.

Payment Processing: Facilitate secure transactions and credits management using Stripe.

II. METHODOLOGY

Before commencing our project, we embarked on an exploration to identify technologies that aligned with our project objectives. Our research led us to a compelling article1 highlighting Next.js as a pivotal technology suitable for our endeavor. This article elucidated how Next.js could enhance scalability and performance, thereby extending our project's capabilities. This discovery influenced our methodology and guided our decision to incorporate Next.js into our project's architecture.

Next.js presents several advantages that position it as an optimal choice for your project:

- Superior Performance: Next.js delivers exceptional performance in terms of load times, ensuring a smooth and responsive user experience.
- Enhanced SEO: With support for server-side rendering, Next.js significantly boosts website SEO, enhancing visibility and reach.
- Accelerated Development: Next.js streamlines development processes, facilitating faster time to market, enabling rapid iteration and product improvement based on user feedback.
- Design Flexibility: Next.js empowers the creation of entire pages, not just components, providing greater flexibility in designing your website's user interface.
- Robust Community Support: Benefit from Next.js' extensive and active developer community, ensuring access to resources and assistance in troubleshooting any encountered issues.

Next.js and React.js:

Imaginify leverages cutting-edge technologies such as Next.js and React.js to construct its frontend infrastructure. Next.js, a widely adopted open-source framework, offers indispensable features like server-side rendering and static site generation, essential for crafting high-performing websites. Meanwhile, React.js, a renowned JavaScript library, facilitates the development of reusable UI components, thereby amplifying code efficiency and maintainability.

Tailwind CSS:

Tailwind CSS, a utility-first CSS framework, is employed for styling Imaginify's pages. This framework offers a plethora of low-level utility classes, empowering developers to fashion entirely bespoke designs directly within their HTML, streamlining the styling process.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

Shadcn for Styling:- Shadcn is a set of elegantly styled, easily obtainable, and adaptable React components that you can use with Next.js to create cutting-edge web applications. You may quickly and simply develop visually appealing and useful user interfaces using Shadcn.

Shadcn allows you to build your own component library. It provides a lot of components out of the box for dashboards, cards, tasks, forms and many more. Shadcn uses Tailwind CSS behind the scenes, so you can naturally style them. As of right now, it supports Next.js, Gatsby, Remix, Astro, Laravel, and Vite.

Login and Registration:

The project's login and registration features are executed through Clerk, a service dedicated to user management and authentication. Leveraging the Clerk SDK, we've integrated pre-built SignIn and SignUp components to construct our login and registration pages.

Clerk's incorporation offers a robust and secure solution for user management, seamlessly integrating with popular web frameworks like Next.js. This simplifies the integration of authentication into web projects. For security, Clerk employs modern protocols and advanced features such as secure password hashing, Cross-Site Request Forgery (CSRF) protection, and multi-factor authentication, ensuring the system's resilience.

The pre-built SignIn and SignUp components significantly reduce development time, encompassing tasks from data collection to validation and error handling. The SignIn component, imported from @clerk/nextjs, facilitates the creation of the login page, providing a comprehensive sign-in interface. Similarly, the SignUp component facilitates the registration process, offering form fields, validation, and error handling.

Both components are enclosed within an AuthLayout component, furnishing a tidy and focused layout for the login and registration pages. This layout can be further customized using CSS and tailwindcss to meet specific requirements.

Storing Clerk API keys in a .env file enhances security, safeguarding sensitive information. These keys authenticate our application with Clerk, enabling user management functionalities.

This approach ensures a secure and user-friendly experience, allowing us to concentrate on our application's unique features while entrusting user authentication to a dependable third-party service.

Implementation of MongoDB Atlas:

The implementation of MongoDB Atlas within the Imaginify project represents a crucial component in its architecture, enriching its capabilities for data storage, management, and scalability. As an AI-Powered Image Enhancement & Transformation SaaS built on Next.js 14, integrated with Cloudinary AI, Clerk, and Stripe, Imaginify relies on MongoDB Atlas to handle the persistent storage of diverse data types integral to its operation.

Data Storage and Management:

MongoDB Atlas serves as the primary database solution for Imaginify, offering a flexible and scalable NoSQL database environment. It efficiently stores various types of data, including user profiles, image metadata, subscription details, and transaction records. MongoDB's document-oriented model allows for the storage of complex data structures, aligning seamlessly with the project's requirements.

Integration with Next.js:

MongoDB Atlas integrates seamlessly with the Next.js frontend, providing a robust backend data store for the application. Through the use of Mongoose or the MongoDB Node.js driver, Imaginify's Next.js components can communicate with the MongoDB Atlas database, enabling smooth data retrieval, manipulation, and storage operations.

Scalability and Performance:

MongoDB Atlas's cloud-based architecture ensures scalability and performance, crucial for a SaaS application like Imaginify. As the user base and data volume grow, MongoDB Atlas can dynamically scale to accommodate increased demand, ensuring optimal performance and responsiveness for users accessing the platform.

Data Security:

MongoDB Atlas prioritizes data security, offering robust encryption mechanisms and access controls to safeguard sensitive information stored within the database. With features such as Transport Layer Security

(TLS) encryption and Role-Based Access Control (RBAC), MongoDB Atlas ensures the confidentiality and integrity of data stored within Imaginify.

Real-time Data Processing:

MongoDB Atlas enables real-time data processing capabilities, allowing Imaginify to deliver dynamic and personalized user experiences. By leveraging MongoDB's aggregation pipeline and change streams, the application can perform complex data manipulations and trigger actions in response to user interactions or system events.

In conclusion, the implementation of MongoDB Atlas within the Imaginify project empowers the platform with robust data storage, management, scalability, and security capabilities. As a foundational component of its architecture, MongoDB Atlas plays a pivotal role in enabling Imaginify to deliver a seamless and feature-rich experience for users engaging with AI-powered image enhancement and transformation functionalities.

Integrating Cloudinary with Next.js:

Incorporating Cloudinary with Next.js in the Imaginify project marks a significant stride towards realizing its ambition of an AI-powered image enhancement and transformation SaaS platform. Cloudinary's extensive array of image management and manipulation tools, alongside Next.js's robust frontend development capabilities, synergistically enhance the project's functionality and user experience.

Cloudinary's AI-driven features, including image enhancement, background removal, object removal, object recoloring, and generative fill, seamlessly align with Imaginify's objectives. By integrating Cloudinary AI with Next.js, users gain access to advanced image editing functionalities directly within the web application, obviating the necessity for intricate third-party software or manual editing processes.

Furthermore, Cloudinary's smooth integration with Next.js expedites the development cycle, allowing developers to seamlessly incorporate image management and manipulation features. Complemented by Next.js's server-side rendering capacities, Cloudinary ensures swift and responsive image loading across various devices and screen dimensions.

The amalgamation of Cloudinary with Next.js also facilitates dynamic image transformations, empowering users to tailor images according to their preferences and requirements. With Cloudinary's URL-based transformation capabilities, developers can programmatically adjust image properties such as size, format, quality, and effects, enabling real-time customization of images.

Moreover, Cloudinary's extensive set of APIs and SDKs simplifies the implementation of image-related functionalities within the Imaginify platform. Leveraging Cloudinary's APIs, developers can efficiently perform tasks such as uploading, transforming, and delivering images, while Next.js seamlessly handles frontend presentation and interaction logic.

In essence, the integration of Cloudinary with Next.js in the Imaginify project elevates the platform's image editing capabilities, performance, and user experience. By harnessing Cloudinary's AI-driven features and Next.js's frontend development tools, Imaginify emerges as a sophisticated SaaS solution for AI-powered image enhancement and transformation, poised to revolutionize users' interaction with digital images.

Implementation of Stripe:

Stripe is seamlessly integrated into our project in the following manner:

User Authentication: The authentication process relies on the auth function from Clerk to authenticate the user and retrieve their current details.

User Subscription Retrieval: Prisma is utilized to fetch the user's subscription details from the database.

Stripe Session Creation: Based on the user's subscription status and Stripe customer ID, a Stripe billing portal session is initiated for existing subscribers. For new users without a subscription, a Stripe checkout session is generated for subscribing to the product.

Stripe webhooks serve as a means for Stripe to communicate with our server regarding various events occurring in the Stripe account, such as payments, subscriptions, and disputes. When an event is triggered, Stripe sends a POST request to our server containing a JSON payload, including an Event object.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

The POST function, an asynchronous function, is responsible for handling incoming webhook events from Stripe. It parses the request to extract the body and Stripe-Signature header, verifies the event's authenticity using the constructEvent function from the Stripe library, and proceeds to handle the event accordingly. For instance, if the event type is checkout.session.completed, subscription details are retrieved from Stripe and a new subscription is created in the database using Prisma. Similarly, if the event type is invoice.payment_succeeded, the existing subscription in the database is updated with details from Stripe.

Stripe stands out as a premier online payment processing platform renowned for its versatility, resilience, and user-friendly integration capabilities1. It proves especially advantageous for subscription management, owing to several key factors:

Diverse Pricing Models: Stripe accommodates a range of pricing structures, including usage-based, tiered, and flat-fee plus overage models. This adaptability empowers businesses to swiftly tailor their pricing strategies to meet evolving user demands.

Automated Invoicing: Simplifying billing management, Stripe automates the generation of invoices for each billing cycle. This automation reduces manual workload and streamlines the invoicing process.

Enhanced Security Measures: Stripe prioritizes robust security protocols, encompassing secure password hashing and multi-factor authentication. These measures ensure the integrity of customer data and transactions, instilling trust and confidence among users.

Customer Self-Service Features: Through its customer portal, Stripe offers customers self-service options for managing their subscriptions. This provision enhances the user experience by granting customers greater control and autonomy over their accounts.

Effortless Integration: With seamless integration options across various web frameworks, Stripe simplifies the process of incorporating payment processing functionality into projects. This facilitates smooth and efficient implementation, saving developers time and effort.

In case of errors during these operations, they are logged, and an appropriate status code is returned.

The implementation of Stripe in our project provides a robust and secure method for managing subscriptions. By leveraging Stripe's features and integrating with our backend, we ensure real-time synchronization between our application and Stripe account, enabling timely responses to important events.

Customer Support:

The CrispProvider component plays a central role in integrating the Crisp Chat service into the IntellAI project. Crisp is a platform facilitating customer messaging, enabling businesses to engage with their customers directly through their website or mobile application.

Component Structure:

The CrispProvider component, functioning as a functional component, encapsulates the CrispChat component. Presumably, the CrispChat component serves as a wrapper around the Crisp Chat widget, which presents the actual chat interface to users.

Functionality:

Upon rendering the CrispProvider component, the CrispChat component is included in the component tree. Consequently, the Crisp Chat widget becomes embedded within the page, enabling users to engage with the customer support team directly from the application interface.

Use Case:

Typically, this component is positioned at a high level within the application, such as within the main layout or App component. This strategic placement ensures the presence of the Crisp Chat widget across all pages of the application, ensuring consistent accessibility to customer support services.

In conclusion, the CrispProvider component assumes a simple yet pivotal role in the project, seamlessly integrating the Crisp Chat service into the application interface and facilitating direct communication between users and the customer support team.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

Deployment to Vercel Network

React and Node.js are combined by Next.js to provide a great developer experience. With its help, we may develop dynamic front-end programs that enable progressive static regeneration, server-side rendering, and static page generation. It can also be used to create API tiers that our front-end apps may utilize, and we can even expose these endpoints to other apps. Next.js page components allow JavaScript to be written isomorphically, which offers a great deal of freedom.

Applications written in Next.js can be installed on the Vercel network, which distributes server tasks to "the edge". In order to achieve extremely quick response times and minimal latency in production, it employs the Edge Runtime, which is leaner than Node.js. Vercel is also very good at managing static pages and server-side produced content, and it can handle a very high volume of concurrent requests. This provides our team with the confidence to develop and deliver scalable products with extremely high availability, minimal configuration, and maintenance requirements.

Vercel offers several advantages for online deployment, including:

Ease of Use: Renowned for its user-friendly interface, Vercel simplifies the deployment process. Once configured and connected to a Git repository, deploying a web application becomes as straightforward as making a Git commit and push.

Continuous Deployment: Vercel boasts a robust continuous deployment (CD) pipeline. It automates the creation of distinct builds for every commit to your Git repository, encompassing production, staging, or "preview" commits.

Preview Deployments: With each Git commit, Vercel automatically generates builds of your project, providing unique URLs for accessing individual builds. This transforms each Git repository commit into a fully functional "preview" deployment that team members can access, utilize, and review.

Step-by-Step Deployment: Comprehensive articles offer detailed guides on deploying applications via Vercel, from setting up your Vercel account to configuring deployment settings.

Integration with Git: Seamlessly integrating with Git, Vercel transforms each repository commit into a "preview" deployment, facilitating efficient collaboration and testing.

Support and Community: Backed by a supportive community and responsive support team, Vercel ensures users receive prompt assistance with any challenges they encounter.

III. RESULT

Here are some results from our website:

Fig 1: Landing Page

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

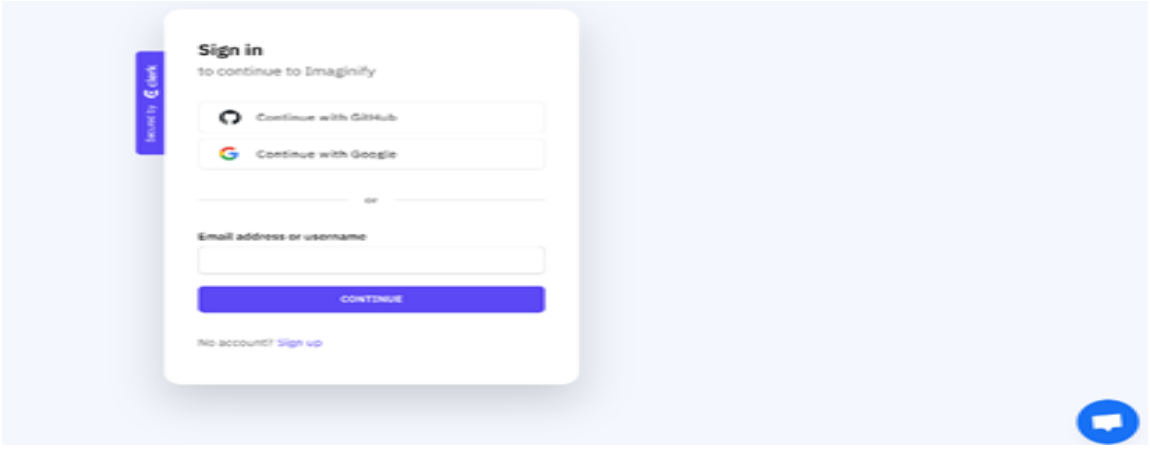

Fig 2: Signup Page

Clerk helps to establish a secured authentication with seamless user interface.

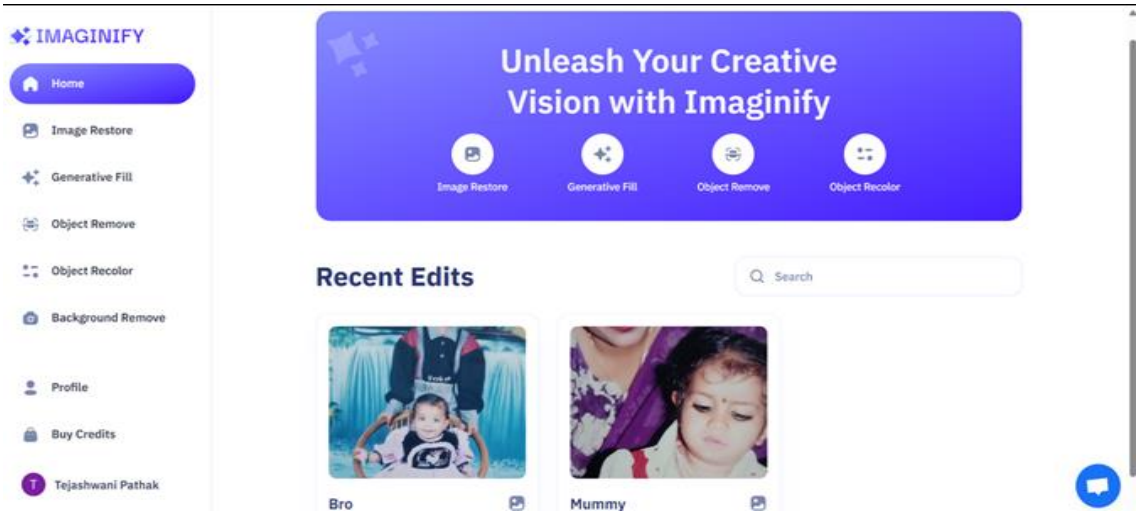

Fig 3: Dashboard Page

Fig 4: Result of Image Restoration

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

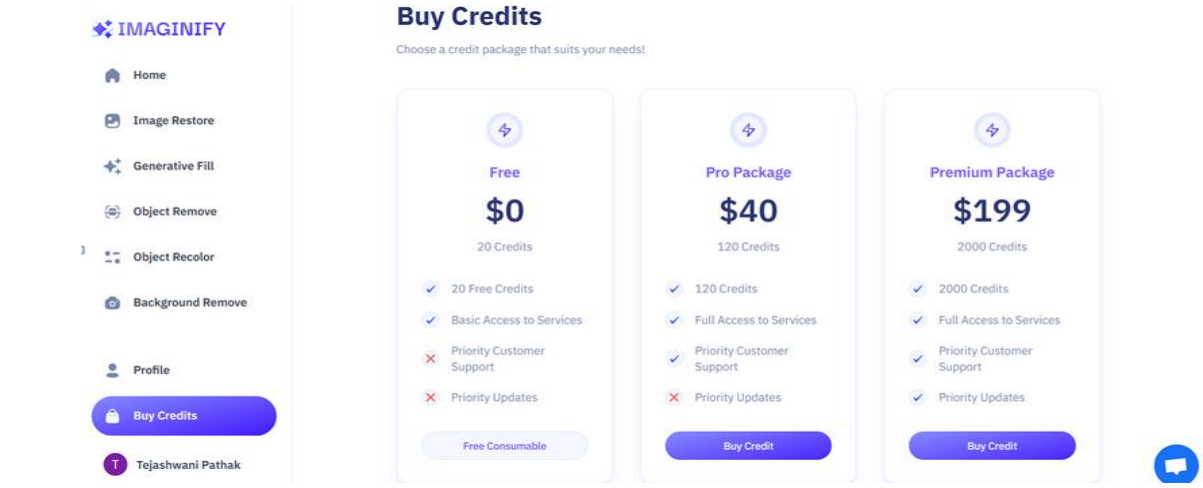

Fig 5: Buy Credits Page

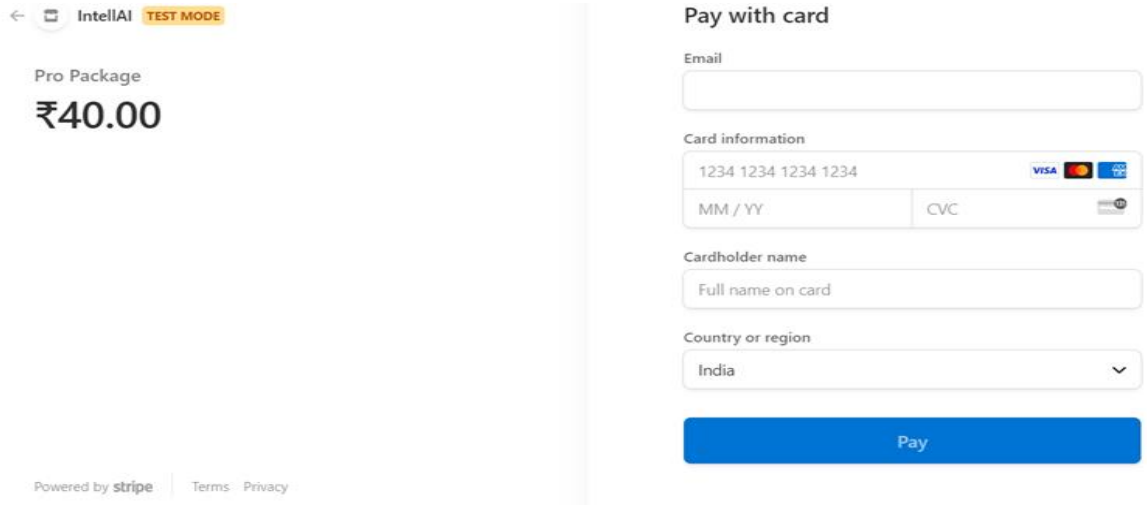

Fig 6: Stripe Checkout Page

IV. FUTURE SCOPE

In order to better serve its consumers' changing demands, Imaginify plans to provide more features and functionalities in the future. Future improvements could include support for video processing, the addition of more AI algorithms for complex picture editing jobs, and team-based editing workflows through collaborative tools. Imaginify also intends to investigate joint ventures with agencies, e-commerce sites, and content creators in order to grow its user base and penetrate new areas.

V. CONCULSION

Imaginify, which provides customers with a strong yet user-friendly platform driven by AI technologies, signifies a paradigm shift in the field of image enhancement and transformation. With its cutting-edge features, sturdy technical framework, and smooth integration with Clerk, Stripe, Cloudinary AI, Next.js 14, Imaginify has the ability to completely change how photos are altered and modified online. With the growing need for superior visual content, Imaginify is positioned to take center stage in the digital imaging sector.

VI. REFERENCES

[1] Lydia Hallie, (2022), "Upgrading next for instant performance improvements" Vercel official blogpost;
[2] Adam Wathan ,December 19, 2023, "Tailwind CSS v3.4: Dynamic viewport units, :has() support, balanced headlines, subgrid, and more"
[3] Durgaprasad Budhwani, Dec 15, 2023, "Unlocking the Power of Retrieval-Augmented Generation with MongoDB Atlas Vector Search, Prisma, and OpenAI Embeddings"

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:05/May-2024 Impact Factor- 7.868 www.irjmets.com

[4] Malith parishan, (02 August 2021) "10 Ways to Improve Your Next.js App Performance", Medium Article;
[5] "Use Clerk with Next.js", official clerk website
[6] Imran Alam (02 March 2019) "Best practices to increase the speed of next.js apps", Stack Over flow discussions blog post;
[7] Seah Conolly (27 Dec 2018), "Performance difference between next.js and react.js"; Log Rocket Series of articles;
[8] Christian Vega (27 Feb, 2017), "Client-side vs. server-side rendering: why it's not all black and white", Free Code Camp;
[9] Andrei Neogie, John Smilga (2020), "A Modern way of loading heavy scripting files", Google developers Network";
[10] Chad Murobayasi (2021), "How to handle routing in next.js and similar frameworks", Level up coding articles;
[11] Astha Guptha (26 Apr 2021), "Into the world of server-side rendering with next.js", Hash node blog post;
[12] Acauan Matias (Jan 2018), "Stripe: a Born Global Payment Processor", ResearchGate