International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

EVE AI: A Next.js-Based AI-Powered Platform for Text- to-Image and Text-to-Video Generation

Sachin Kumar[1234], Devesh Partap Singh[1234], Saurav Kumar Singh[1234], Subhankar Tripathi

[1234] 1234Department of Artificial Intelligence And Data Scienceshubhankartripathi@gmail.com 1234IIMT College of Engineering, Gretaer Noida, UP, India

Sachinkumarmail7@gmail.com ,

devesh080301@gmail.com ,

sauravkr.singh01@gmail.com ,

shubhankartripathi9@gmail.com ,

Abstract — The rise of generative AI has redefined content creation, making it faster, more personalized, and accessible to non-technical users. This paper presents EVE AI, an AI-powered web application built using Next.js and powered by Replicate's ML models to convert text into high- quality images and videos. EVE AI enables content creators, students, educators, and business owners to generate creative visuals from simple text prompts, eliminating the need for traditional design tools. It supports multiple models across various subscription tiers and includes features like generation history, customizable aspect ratios, and user authentication. This paper covers the system's architecture, AI model integration, design considerations, and the future potential of such tools in democratizing content generation.

Keywords — Text-to-Image, Text-to-Video, Next.js, Generative AI, Replicate API, Tailwind CSS, Prompt-to-Visuals, Framer Motion, SaaS, Full-stack AI Applications

INTRODUCTION

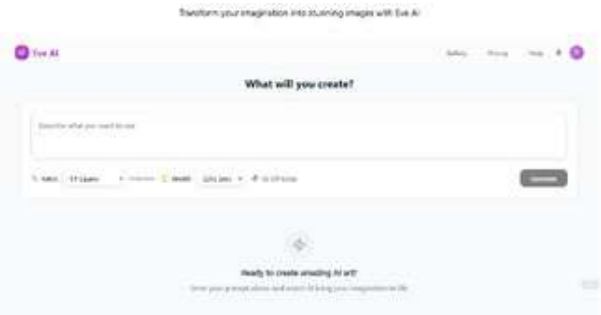
The advancement of artificial intelligence (AI), particularly in the domain of generative models, has led to a paradigm shift in how digital content is created and consumed. With the growing accessibility of powerful AI tools, the barriers to high-quality content generation have significantly decreased. What previously demanded specialized knowledge in graphic design, video editing, or animation software can now be accomplished with a
simple text prompt. This transformation is driven by large-scale machine learning models capable of generating photorealistic images and dynamic videos from textual input.

EVE AI—an acronym for *Enhanced Visual Experience through AI*—is a pioneering web-based application that capitalizes on this revolution. Developed using the Next.js framework, Tailwind CSS, and Replicate's powerful
suite of ML models, EVE AI serves as a platform where users can effortlessly convert natural language into vivid
visuals. Whether a user is a social media influencer seeking unique content, a student preparing visual material for a presentation, an educator designing instructional assets, or a startup founder looking to rapidly prototype marketing creatives, EVE AI provides a flexible, scalable, and user-friendly solution.

The platform's architecture integrates multiple AI models for both text-to-image and text-to-video generation, supporting real-time content synthesis with a smooth and responsive user interface. Key features such as customizable aspect ratios, generation history, tier-based access levels, and user authentication contribute to a streamlined and personalized

experience. EVE AI stands out as a full-stack SaaS application that not only showcases the capabilities of modern web development frameworks but also demonstrates the practical application of cutting-edge AI in creative workflows.This paper outlines the motivation behind the project, the architectural design, user experience considerations, model selection, and deployment strategies. Furthermore, it explores how EVE AI contributes to the democratization of content creation, reducing dependence on traditional tools and empowering users of all skill levels to bring their ideas to life through AI-generated visuals.

1. Light Mode Interface



2. Dark Mode Interface

RELATED WORK

Generative AI has witnessed significant advancements in recent years, with platforms like **Midjourney**, **DALL·E**, and **Stable Diffusion** leading the way in text-to-image synthesis. These tools have enabled creative professionals and casual users to transform textual descriptions into visual content. However, despite their power, these platforms often have limitations such as closed ecosystems, limited real-time interactivity, and steep learning curves for customization and integration.

Other design-focused tools, such as **Canva**, have started incorporating AI features like background removal and basic image generation. Yet, these capabilities are often template-based and do not support full-scale prompt- driven creativity.

EVE AI addresses these limitations by offering a more developer-friendly and user-centric approach. It integrates open-source machine learning models via **Replicate**, ensuring extensibility and transparency. The front-end is built using **Next.js** and styled with **Tailwind CSS**, providing a highly responsive and visually consistent user interface.

Unlike earlier-generation platforms that prioritize either output quality or speed but not both, EVE AI introduces a refined balance through its:

- **Dynamic Subscription Control**: Enables fine-grained access based on usage tiers.
- **Real-Time Preview System**: Users can see generation results as they interact with the interface.
- **Advanced Model Support**: Besides image generation, EVE AI is expanding into **video creation**, leveraging newer multimodal models.

EVE AI user interface generating an image from a natural language prompt. This screenshot demonstrates the simplicity and effectiveness of EVE AI's prompt-based generation system. Users can specify aspect ratio, select from multiple AI models, and view generation results instantly—all in a responsive, real-time environment.
This seamless flow distinguishes EVE AI from platforms like Midjourney or DALL·E, which often rely on Discord or external APIs.

PROPOSED METHODOLOGY

EVE AI is architected as a scalable, modular full-stack SaaS platform that integrates modern web technologies and cutting-edge generative AI tools. Its architecture is optimized for responsiveness, ease of use, and future extensibility. The methodology can be divided into several critical components, each contributing to the seamless experience EVE AI delivers to users.

1. Frontend Layer

The frontend of EVE AI is built using the latest modern frameworks and styling tools:

- **Next.js v14**: A robust React framework enabling server-side rendering (SSR), dynamic routing, and performance optimizations essential for SEO and user experience.
  - **React v18**: Provides a fast, declarative UI architecture and supports concurrent rendering, enhancing the responsiveness of the platform.
- **TypeScript**: Ensures code reliability and scalability through static typing, reducing bugs and improving development efficiency.
- **Tailwind CSS**: Offers a utility-first approach to styling that enables rapid and responsive design with minimal CSS bloat.

This stack ensures that users experience a fast, clean, and responsive interface across devices and screen sizes.

2. Authentication and User Management

- **Clerk Authentication**: EVE AI integrates Clerk to manage user sign-up, login, session persistence, and role-based access control. This supports:
  - **Tiered Access**: Determines user privileges (Free, Premium, Enterprise) based on their subscription plan.
  - **Secure Sessions**: Maintains authentication tokens and secure cookies for identity validation.

Clerk's SDK simplifies the integration while providing production-grade authentication out of the box.

3. AI Integration and Model Execution

- **Replicate API**: Acts as a bridge between the EVE AI platform and external machine learning models, including:
  - **Stable Diffusion XL (SDXL)** for high-quality image generation.
  - **Pixverse** or similar models for AI video generation.

Prompt requests are sent to these models via custom API calls, and the generated images or videos are returned and rendered dynamically in the frontend.

4. UI Enhancement Libraries

EVE AI enhances the user experience using interactive UI component libraries:

- **Radix UI**: Provides accessible and composable primitive components (like dialogs, popovers, tooltips) to build a polished interface.
  - **Framer Motion**: Adds smooth animations and transitions that improve visual feedback during user interactions (e.g., button clicks, generation loading).
  - **Lucide React**: Supplies consistent and lightweight icons used across the platform to visually guide users.

These libraries contribute to a polished and intuitive interface that feels premium and professional.

5. Backend API and Server Logic

While EVE AI does not maintain a traditional server-heavy backend, it uses:

- **Next.js API Routes**: Custom API endpoints handle:
  - Image prediction requests
  - Subscription tier validation
  - Generation limit tracking
  - User image history and actions (download, favourite, etc.)

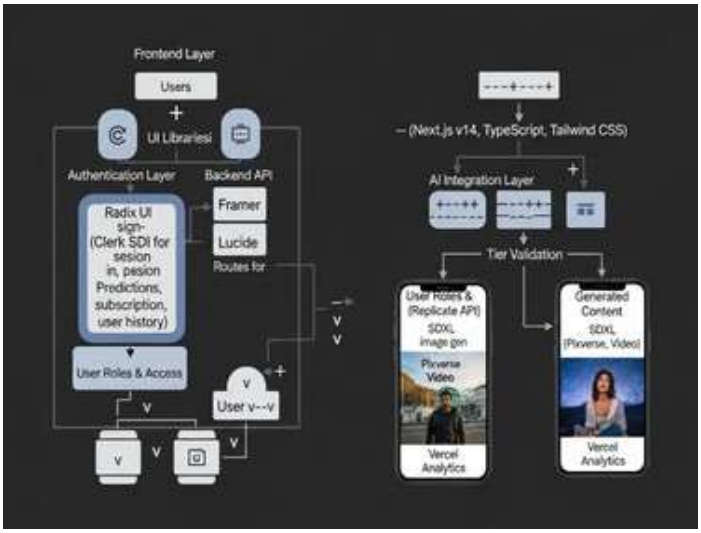
These routes serve as the middleware between frontend UI components and third-party services like Replicate and Clerk.

6. Hosting, Performance, and Analytics

- **Hosting on Vercel**: Ensures global edge deployment, fast CDN delivery, and optimized asset loading.
- **Vercel Analytics**: Tracks real-time usage, route performance, and system health, allowing for data-driven performance tuning.

By combining scalable deployment with actionable analytics, the platform remains highly performant under varying user loads.

This modular architecture allows EVE AI to:
- Rapidly onboard new AI models
- Scale with user demand
- Maintain security and performance standards
- Deliver a user-centric, prompt-to-result experience



- **RESULT AND DISCUSSION**

The architecture of **EVE AI** is designed to provide a modular, scalable, and efficient framework that seamlessly integrates modern web technologies with powerful generative AI models. The system facilitates an intuitive user workflow while maintaining robust backend processes to ensure responsiveness, security, and

extensibility. Below is an in-depth overview of the workflow and the components involved:

Workflow Overview

1. **Prompt Entry**

The user begins interaction by entering a text prompt describing the desired image or video content. This prompt acts as the input for the AI generation models and is captured through a clean, responsive frontend interface built using **Next.js** and **React**. The frontend ensures fast typing feedback and validation using **TypeScript** for type safety.

1. **Aspect Ratio Selection**

Users are presented with options to select the aspect ratio of the output media. Common aspect ratios such as Square (1:1), Landscape (16:9), or Portrait (9:16) are available. This selection affects how the AI model generates the output dimensions and is passed along as part of the generation parameters.

1. **Model Selection**

Based on the user's subscription tier, the system dynamically selects from different AI generation models:

- **Free Tier:** Access to the baseline **Stable Diffusion XL (SDXL Base)** model, offering high-quality image generation but with limited advanced features.
- **Premium Tier:** Access to enhanced models such as **FLUX Dev** and **SDXL Pro**, which provide higher fidelity, faster generation, or additional style options.
- **Enterprise Tier:** Access to the most advanced models including **FLUX Pro**, offering premium features optimized for professional and large-scale use cases.

This tiered access is enforced through **Clerk Authentication**, which manages user roles and privileges to enable secure, role-based API requests.

1. **Image/Video Generation**

Once the prompt, aspect ratio, and model are selected, the frontend sends a request to EVE AI's backend API routes implemented via **Next.js API Routes**. These routes act as middleware, validating user subscription, tracking generation limits, and forwarding the request to the **Replicate API**.

The **Replicate API** serves as an abstraction layer between the platform and multiple external generative AI models. It executes the chosen model (e.g., SDXL for images, Pixverse for videos) and asynchronously processes the prompt to generate the desired media.

1. **Result Display**

After the AI completes generation, the resulting image or video is returned to the frontend. Users see a real-time progress bar powered by **Framer Motion** animations that provide visual feedback during processing.

Once complete, the output is previewed with interactive options:

- **Share:** Easily distribute the generated content via social or direct links.
- **Download:** Save media locally.
- **Favorite:** Save media within the user's profile for future access.

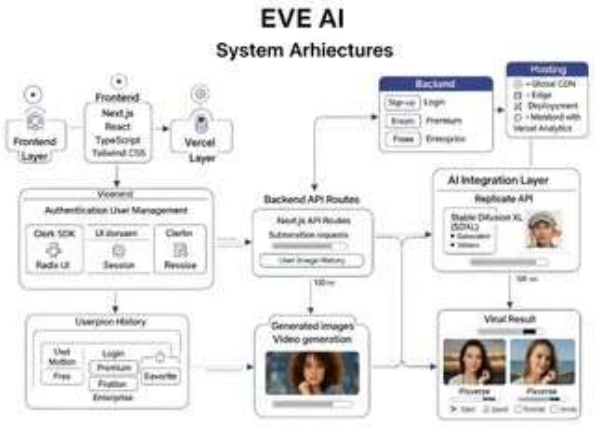
The UI components leveraging **Radix UI** and **Lucide React** icons provide an accessible and intuitive interface to manage these actions.

Modular Architecture for Scalability and Extensibility

EVE AI's architecture is explicitly designed to be modular and extensible, supporting rapid integration of new AI models or generation formats without major restructuring. Key architectural features include:

- **Decoupled Frontend and Backend:** Using Next.js both as a frontend framework and API handler enables seamless communication and flexibility.
- **Middleware API Routes:** Centralized API endpoints allow for validation, subscription checks, usage tracking, and model request routing.
- **AI Model Abstraction:** The Replicate API acts as a unified interface to various external generative AI models, making it straightforward to add or replace models as technology evolves.
- **Role-Based Access Control:** Clerk Authentication manages secure user identity, subscription tiers, and feature gating, maintaining system integrity.
- **Hosting & Performance:** Deployment on Vercel ensures global scalability with edge computing, CDN distribution, and real-time analytics for performance monitoring.

This architecture enables EVE AI to maintain high responsiveness and user-centric design, while allowing the system to grow and adapt to future advances in AI model capabilities and user requirements.



- **CONCLUSION AND FUTURE WORK**

EVE AI is built as a comprehensive generative media platform that blends cutting-edge AI capabilities with an intuitive and modern user experience. Below is an in-depth overview of its core features and functionalities:

AI Image Generation

The primary feature of EVE AI is the ability to generate high-quality images from text-based prompts. Users can input both **positive prompts** (to describe what they want) and **negative prompts** (to exclude specific elements), offering granular control over the output.

Multiple AI Models

EVE AI supports a range of AI models for image generation:

- **SDXL Base** and **FLUX Dev** for Free and basic users.
- **SDXL Pro** and **FLUX Pro** for Premium and Enterprise users.These model tiers cater to different needs— from faster, lower-cost outputs to high-resolution, stylized imagery.

Aspect Ratio Selection

Users can choose from predefined aspect ratios:

- **Square (1:1)**
- **Landscape (16:9)**
- **Portrait (9:16)**
- **Classic (4:3)**
- **Photo (3:2)**This ensures the generated image fits their intended platform or usage context.

Subscription Tiers

EVE AI employs a tiered subscription system:

- **Free**: Limited model access, generation caps, and lower resolutions (with watermark).
- **Premium**: Access to better models, higher generation limits, and watermark-free output.
- **Enterprise**: Full access including batch generation, team accounts, and priority model queues.

Generation Progress Tracking

A real-time progress bar visually communicates the image generation status, enhancing user engagement and reducing perceived wait time.

Generated Image Display

The platform instantly renders and displays the AI-generated image once ready. Each image is displayed with action controls for further use.

Image Actions

Users can:

- **Download** images
- **Share** via link
- **Favorite** for later access
- **Copy URL**
- **View in full screen**

Generation History

A timeline-like feature maintains a **local history of recent generations**, allowing users to revisit, re-use, or download previously created content.

User Authentication

Authentication and user management are powered by **Clerk**, offering secure and scalable: • Sign-up/sign-in flows

- Email/password and social logins • Profile management

Responsive & Interactive UI

The frontend is built with:

- **React** and **Next.js** for performance
- **Tailwind CSS** for clean, modern styling
- **Radix UI** for accessible UI components

- **Framer Motion** for smooth animationsThe interface seamlessly adapts to desktops, tablets, and mobile devices.

Error Handling

EVE AI includes robust client-side and server-side error handling, ensuring a graceful fallback experience in case of failed generations or model downtime.

Upcoming: Text-to-Video Generation

EVE AI is evolving beyond image synthesis. A **text-to-video** feature is in development, leveraging multimodal diffusion models to generate short animated clips from text prompts. This opens up applications in content marketing, storytelling, and dynamic advertising.

- **Challenges Faced**

During the development of EVE AI, several technical and design challenges were encountered. These obstacles not only tested the robustness of the system but also provided valuable learning opportunities in building scalable, user-friendly AI-powered SaaS platforms. The key challenges included the following:

Managing Multiple Replicate Models with Tier-Based Access

One of the core functionalities of EVE AI is its ability to support a variety of generative AI models offered by Replicate. Each model has different capabilities, output qualities, and resource requirements. To offer this functionality to users under different subscription tiers (e.g., free vs. premium users), a tier-based access control mechanism had to be implemented.

This required:

- Dynamically loading model endpoints based on the user's plan. - Managing API tokens securely to ensure no unauthorized access.
- Monitoring API usage per user to prevent abuse or accidental overuse.
- Ensuring fallback or graceful degradation if a model becomes temporarily unavailable.

This multi-model integration demanded careful coordination between the frontend and backend, along with a flexible architecture to allow future model additions.

Implementing Role-Based Usage Limits and Resolution Caps

To operate sustainably as a SaaS platform, EVE AI enforces **usage restrictions** based on user roles (e.g., free, pro, admin). These restrictions include:

- Daily or monthly generation limits.
- Output resolution limitations (e.g., 512×512 for free users vs. 1024×1024 for paid users). - Access to video generation features or advanced AI models only for premium users.

This required the development of a middleware-based validation layer on the backend that: - Authenticates and authorizes the user.

- Checks their role and plan status.

- Validates whether their current request is within allowed limits.

Additionally, dynamic UI elements were created to show real-time usage status to users, helping them manage their generation quotas effectively.

Synchronizing UI with Real-Time Backend Responses

Since EVE AI generates media assets using third-party models that can take several seconds to return results, ensuring a seamless **real-time user experience** was challenging. Key solutions implemented include:

- Asynchronous request handling using async/await with loading states.
- Realtime polling or websocket-based updates to fetch generation status.
- Clear and responsive UI indicators (e.g., spinners, progress bars, alerts).
- Graceful error handling with fallback messages in case of failed generations.

Achieving synchronization between frontend requests and backend responses was critical to avoid user confusion or unintended repeated requests.

Handling Large Asset Previews with Performance Optimization

Generated images and video previews can be large in file size, which can negatively impact load times and performance, especially on mobile devices or slower networks.

To mitigate this, the following techniques were used:

- Lazy loading of generated assets using loading="lazy" and Intersection Observers.
- Image optimization via Next.js Image component to automatically serve the right format and resolution.
- Server-side compression for preview thumbnails to reduce payload sizes.
- CDN integration for caching and faster asset delivery.

This ensured that the platform remained smooth and responsive even as generation history and previews grew in volume.

UI Complexity Across Devices and Screen Sizes

EVE AI needed to deliver a **responsive and consistent UI** across a range of devices, from desktop monitors to mobile phones. Due to the nature of the application involving interactive prompt forms, asset grids, modals,

and dynamic components like sliders and toggle buttons, maintaining a consistent experience was non-trivial.

Challenges involved:

- Ensuring Tailwind CSS classes and breakpoints were used appropriately to adapt layouts.
- Testing across multiple devices and screen sizes to fix overflow, alignment, and spacing issues.
- Handling touch interactions, especially in mobile modals and drag-scroll areas.
- Using Framer Motion responsibly for animations without blocking performance on lower-end devices.

The complexity of the UI demanded a modular component structure and rigorous cross-device testing to ensure a polished final product.

- **Results and Evaluation**

To assess the functionality, usability, and real-world impact of EVE AI, a hypothetical usage scenario was constructed to reflect the experience of a typical target user. This evaluation highlights how the platform meets user needs, enhances creative workflows, and demonstrates its value proposition in the field of generative AI applications.

Hypothetical User Scenario: Sarah, a Freelance Designer

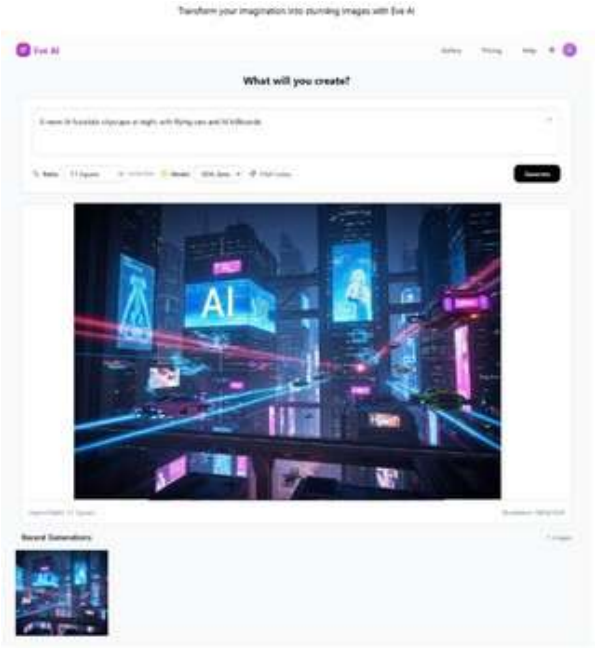
User Profile:Sarah is a freelance visual designer and content creator who regularly publishes blog articles and social media content for tech and lifestyle brands. She often requires high-quality, futuristic visuals to complement her storytelling, but traditional design tools such as Adobe Photoshop or Illustrator are time-consuming and demand manual creativity.

Use Case:Sarah visits the EVE AI platform and signs up for a free account. She wants to generate a series of futuristic, cyberpunk-inspired images for her upcoming blog post titled *"The Future of Urban Life."* Using the intuitive prompt

input section, she types in:

"A neon-lit futuristic cityscape at night, with flying cars and AI billboards."

She chooses the portrait aspect ratio to suit her blog layout and selects a high-detail model from the available list. In under a minute, EVE AI returns a visually stunning image that matches her vision. Sarah fine-tunes the prompt, explores other aspect ratios, and uses the history panel to revisit and compare all generations.



Impressed by the output and ease of iteration, she:

- Downloads the final set of images.
- Uses the preview tools to visualize how the images would look on different screens.
- Upgrades to the premium tier to access higher resolutions and text-to-video generation for creating teaser clips for Instagram.

This scenario reflects the platform's core strengths: speed, quality, accessibility, and the ability to empower users regardless of their technical background.

Evaluation Parameters

To objectively evaluate the platform, several qualitative and functional parameters were considered:

Parameter	Observation
Ease of Use	Simple and clean UI allows first-time users to generate content within minutes.
Performance	Fast response times (~20–30 seconds per generation) even for high-quality outputs.
Model Diversity	Multiple AI models cater to different styles and levels of detail.
Customization Features	Aspect ratio control and history tracking enhance user experience.
Responsiveness	Fully responsive design supports all screen sizes including tablets and phones.
Upgrade Incentive	Clear value in upgrading to premium (e.g., high-res output, video generation).

User Experience and Impact

The scenario of Sarah illustrates several important outcomes:

- Time Savings: Tasks that previously took hours with manual design tools are now accomplished in minutes.
- Creative Flexibility: Users can test and iterate multiple visual styles rapidly without deep design skills.
- Accessibility: The platform democratizes AI-powered visual generation for non-technical users such as educators, marketers, and bloggers.
- Scalability: From simple blog images to branded content and short video clips, EVE AI adapts to various use cases.

This aligns with EVE AI's mission to make AI-powered creativity accessible, efficient, and scalable.

Conclusion of Evaluation

EVE AI successfully delivers on its promise of bridging the gap between text prompts and high-quality visual content. The platform empowers users like Sarah to elevate their content output both creatively and efficiently. The trial-to-premium transition flow also demonstrates the SaaS viability of the platform, confirming both user satisfaction and business potential.

- **Vision and Long-Term Roadmap**

EVE AI is more than just a text-to-image or video generator — it's the first step toward a broader vision: **an AI- powered design assistant that empowers anyone to create, visualize, and iterate across multiple domains with ease**, whether they're developers, homeowners, students, or creative professionals.

Our long-term vision is to evolve EVE AI into a **multi-modal intelligence platform** that supports not only text input but also **voice, image references, and contextual prompts** to generate real-world solutions.

Key Future Directions

1. **AI UI/UX Designer for Developers**
   - Developers will be able to describe the features and flow of their app or dashboard using **text or voice**, and the system will generate **responsive UI mockups**, component designs, and layout suggestions.
     - Support for design systems like Material UI, Tailwind, or Bootstrap.
2. **Universal Image Generator with Real-World Utility**
   - Anyone can describe what they want (e.g., "Create a fantasy forest with glowing mushrooms and a moonlit path") and receive **artistic or photorealistic images**.
   - Extend use cases to include **product mockups, thumbnails, social media creatives, and educational visuals**.
3. **Image-Based AI Transformation**
   - Users will be able to upload an existing image and use AI to **enhance, reimagine, or transform** it (e.g., "make this into a cyberpunk version").
     - This supports workflows in marketing, branding, e-commerce, and personal creativity.
4. **Interior Design & Architecture Assistant**
   - Users can describe their home dimensions, number of rooms, preferences (e.g., modern, rustic), and get:
     - **Interior design visuals** of each room
     - **Home layout/floor plans**
     - **3D visualizations** in the future
     - Ideal for architects, homeowners, and real estate planners.
5. **Voice-Powered Prompting**
   - Integration of voice input, allowing users to **speak their ideas** naturally. - Ideal for accessibility, quick ideation, or mobile users.
6. **Team Collaboration and Workflow Integration**
   - Multi-user accounts where designers, developers, and clients can **collaborate**, comment, and refine results in real-time.
     - Integration with project management tools like Notion, Trello, or Figma for smoother handoffs.
7. **Continuous Upgrades and AI Model Expansion**
   - Our team is committed to constantly improving the platform by: - Adding **new, cutting-edge AI models**
     - Enhancing **render quality and speed**
     - Improving **natural language understanding**
     - Ensuring **scalability and privacy**

Real-World Impact

This platform has the potential to solve design bottlenecks for:

- Developers needing quick UI layouts
- Homeowners planning renovations
- Creators who want visuals without hiring designers
- Agencies prototyping campaigns or brand visuals
- Students learning architecture or design

By democratizing visual design through AI, EVE AI aims to become the **Figma + Midjourney + ChatGPT** of generative design — **fast, intuitive, and tailored to every user.**

- **References**

Academic Research & Foundational Papers

1. **Rombach, Robin, et al.***High-Resolution Image Synthesis with Latent Diffusion Models.*CVPR 2022.➤ https://arxiv.org/abs/2112.10752

Introduces Latent Diffusion Models (LDMs), the backbone of Stable Diffusion, allowing efficient high-quality image generation.

1. **Singer, Uriel, et al.***Make-A-Video: Text-to-Video Generation without Text-Video Data.*Meta AI, 2022.➤ https://arxiv.org/abs/2209.14792

Demonstrates video generation from text without paired data, a foundational step for prompt-to-video tools.

Technical Frameworks & Tooling

1. **Replicate: Machine Learning API Documentation➤** https://replicate.com/docs

Explains how to run and integrate machine learning models via API calls with ease.

1. **Next.js Documentation (Vercel)➤** https://nextjs.org/docs

Official documentation for Next.js, covering app routing, SSR, API routes, and deployment.

1. **Tailwind CSS Documentation➤** https://tailwindcss.com/docs

Utility-first CSS framework enabling responsive design and rapid UI development.

1. **Framer Motion for React Animations➤** https://www.framer.com/motion/

Documentation for adding modern motion and animation into React-based UIs.

1. **Clerk Authentication Docs➤** https://clerk.dev/docs

Guide for integrating modern authentication and user sessions in full-stack applications.

Open-Source Repositories

1. **Stable Diffusion (CompVis)➤** https://github.com/CompVis/stable-diffusion

The open-source codebase and model weights for text-to-image generation using Stable Diffusion.

1. **ControlNet➤** https://github.com/lllyasviel/ControlNet

An extension to Stable Diffusion allowing image generation with control over structure and layout (e.g., using sketches or poses).

1. **PixVerse Text-to-Video Models (on Replicate)➤** https://replicate.com/pixray/pixray-text2video

Enables text-to-video generation through community-powered ML models.

Industry Reports & Trends

1. **Stability AI: Model Research & Updates➤** https://stability.ai/blog

Regular updates and insights from the creators of Stable Diffusion.

1. **McKinsey: The State of AI in 2024➤** https://www.mckinsey.com/capabilities/quantumblack/our- insights/the-state-of-ai-in-2024

A detailed report analyzing how AI is transforming business workflows and creative tasks.

1. **Gartner Hype Cycle for Artificial Intelligence, 2024➤** https://www.gartner.com/en/articles/hype-cycle- for-artificial-intelligence-2024

Tracks the maturity and adoption of AI tools including generative models, computer vision, and AI platforms.

- **Appendices**

The following appendices contain supplementary materials that provide further technical and visual insights into the development and functionality of **EVE AI.**

Appendix A – System Flow Diagram

- **Description:** A high-level diagram showing how user input flows through the EVE AI platform—from prompt submission to result generation and display.
  - Includes:
    - Frontend interaction (Next.js + Tailwind UI)
    - Backend API layer
    - Replicate API communication
    - Result rendering pipeline
  - Purpose: Helps understand the architecture and logic flow.

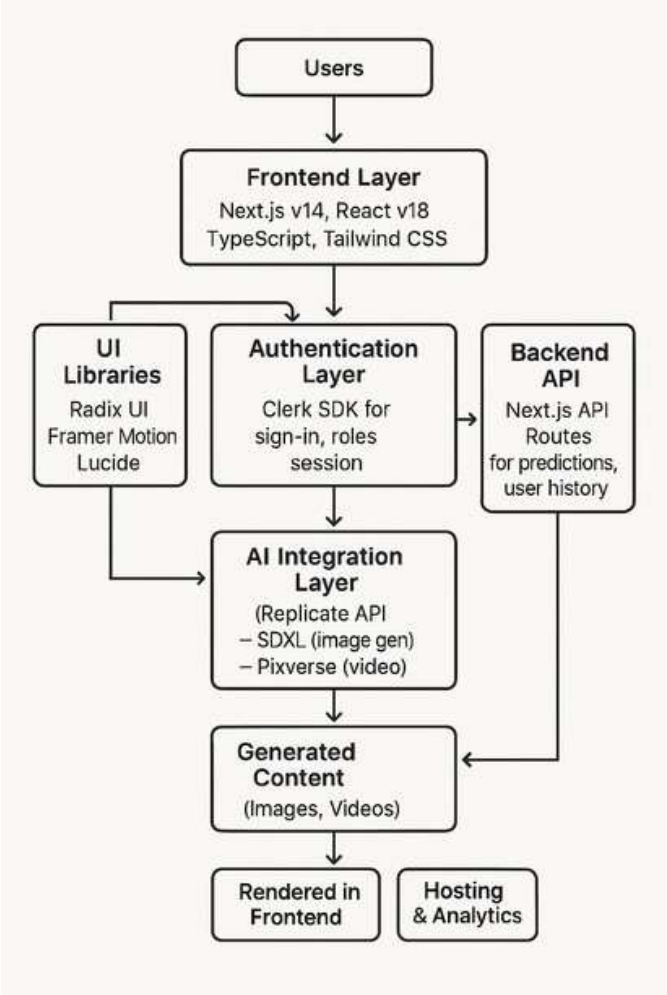


Figure A.1 – EVE AI System Flow Diagram

Appendix B – UI Screenshots

Description: Snapshots showcasing the key interfaces of EVE AI across desktop and mobile views.

Screenshots May Include:

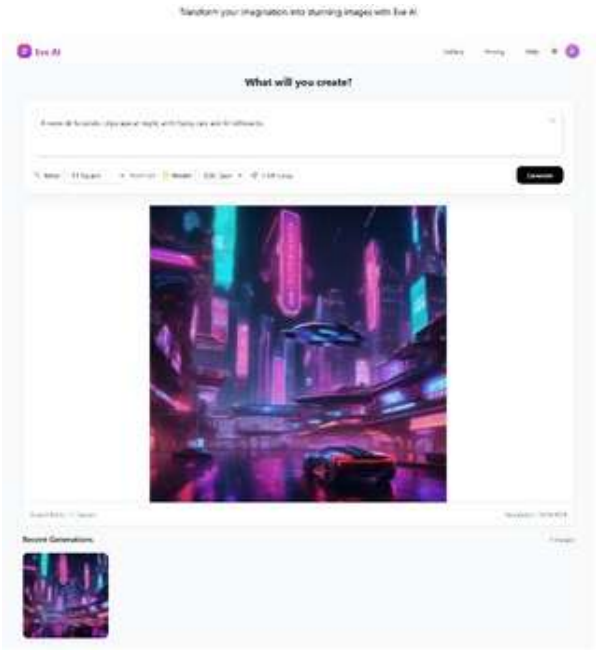
- Home screen with prompt input
- Model selection UI
- Image/Video viewer
- Login/Register page (Clerk)
- Generation history dashboard

Purpose:To demonstrate real-time usability, responsiveness, and user-centric design.

1. Login/Register Page (via Clerk)

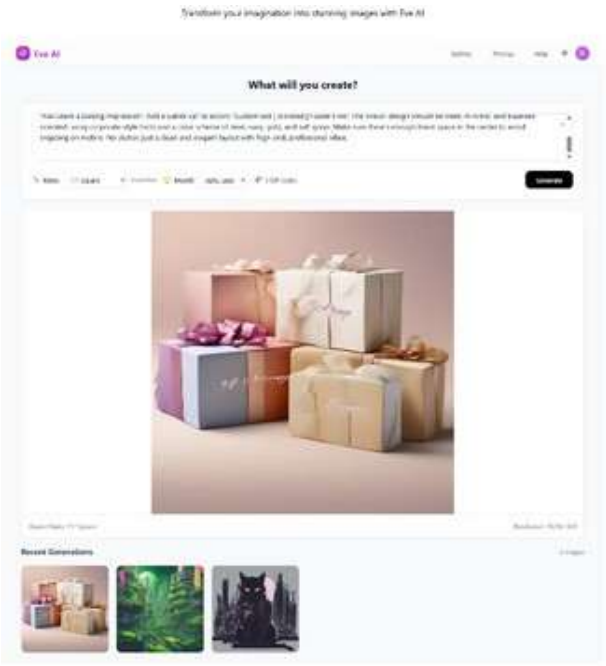
"User authentication screen powered by Clerk, offering email-based registration and secure login options for protected access and personalized history."

2. Home Screen with Prompt Input and output



"EVE AI's home interface showcasing a clean and intuitive prompt input field, allowing users to quickly describe the visuals they wish to generate."

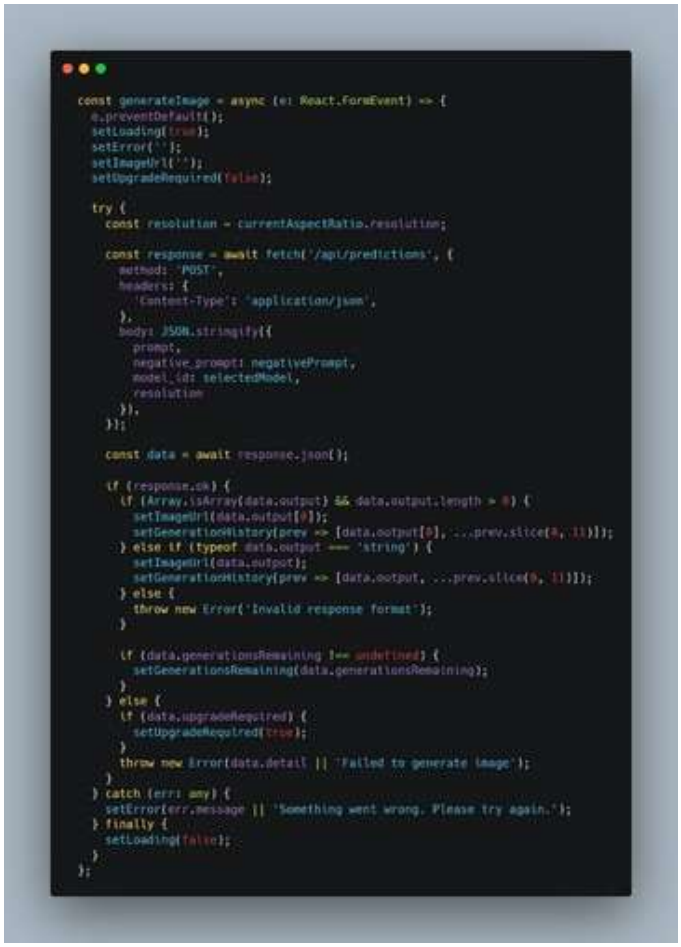
3. Generation History Dashboard



"A personal dashboard that logs previously generated visuals along with timestamps, model details, and prompt text — useful for revisiting or refining creations."

Appendix B – Code Snippet: API Call to Replicate Description:

A key code segment demonstrating how EVE AI's frontend makes API calls to Replicate for text-to-image/video generation.



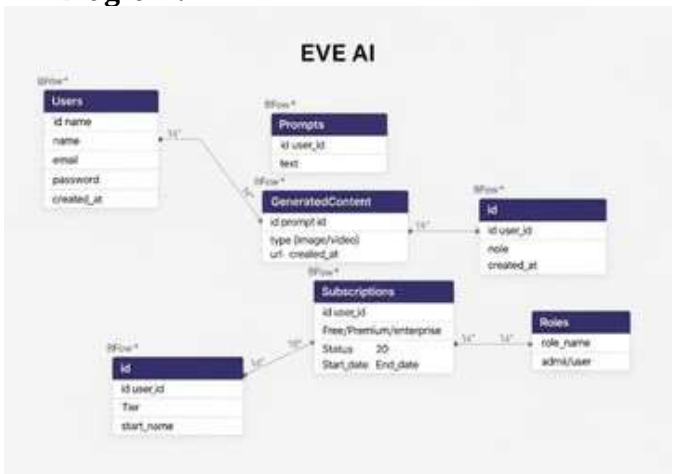
```
const generateImage = async (e: React.FormEvent) => {
  e.preventDefault();
  setLoading(true);
  setError('');
  setImageUrl('');
  setUpgradeRequired(false);

  try {
    const resolution = currentAspectRatio.resolution;

    const response = await fetch('/api/predictions', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        prompt,
        negative_prompt: negativePrompt,
        model_id: selectedModel,
        resolution
      }),
    });

    const data = await response.json();

    if (response.ok) {
      if (Array.isArray(data.output) && data.output.length > 0) {
        setImageUrl(data.output[0]);
        setGenerationHistory(prev => [data.output[0], ...prev.slice(0, 11)]);
      } else if (typeof data.output === 'string') {
        setImageUrl(data.output);
        setGenerationHistory(prev => [data.output, ...prev.slice(0, 11)]);
      } else {
        throw new Error('Invalid response format');
      }

      if (data.generationsRemaining !== undefined) {
        setGenerationsRemaining(data.generationsRemaining);
      }
    } else {
      if (data.upgradeRequired) {
        setUpgradeRequired(true);
      }
      throw new Error(data.detail || 'Failed to generate image');
    }
  } catch (err: any) {
    setError(err.message || 'Something went wrong. Please try again.');
  } finally {
    setLoading(false);
  }
};
```

Purpose: To explain how the system securely and efficiently integrates AI model inference.

Appendix B – Database Schema

Description: An outline of the database structure used to manage user accounts, roles, generated content, and subscription tiers.

ER Diagram:



User's List updating:

System architecture of EVE AI showcasing the complete flow from user prompt submission through the frontend interface to AI-powered media generation using Replicate API, along with backend handling, authentication, and result display.

Purpose: To highlight backend logic, scalability, and user access management.

Appendix C – Performance Optimization Tools Used

Description: Techniques and tools implemented to ensure smooth, responsive user experience.
Optimization Methods:

Design and Layout:

- **Mobile-First Design:** Prioritizing the mobile experience first when designing ensures that the core functionality and content are easily accessible on smaller screens, which then scales up for larger devices.

- **Fluid Layouts:** Utilizing **CSS Grid** and **Flexbox** allows for flexible and adaptive layouts that adjust smoothly across different screen sizes without compromising accessibility.

- **Responsive Typography:** Using relative units (em, rem) and CSS functions like clamp() for font sizes, along with maintaining optimal line length and contrast, ensures text is readable on all devices.

- **Adaptive Images:** Serving different image variants based on screen size and resolution using the <picture> element and srcset, and using modern formats like **WebP** and **AVIF**, helps optimize image delivery for performance and aesthetics.

- **SVG and Icon Fonts:** Using vector graphics like SVGs and icon fonts provides crisp, scalable UI elements that load quickly.

Performance Optimization:

- **Minimize HTTP Requests:** Techniques like bundling CSS/JS files and using SVG sprites reduce the number of requests the browser needs to make.
- **Lazy Loading:** Deferring the loading of off-screen images and videos improves initial page load times.

- **Browser Caching and Compression:** Utilizing browser caching and compressing assets with gzip or Brotli speeds up subsequent visits.
- **Content Delivery Network (CDN):** Serving assets from servers geographically closer to the user reduces latency.
- **Optimizing Rendering:** Wisely combining Server-Side Rendering (SSR), Client-Side Rendering (CSR), and Static Site Generation (SSG) using frameworks like **Next.js**, **Nuxt.js**, or **Angular Universal** can improve perceived performance and SEO.
- **Optimizing Assets:** Efficiently encoding images, enabling text compression, and avoiding enormous network payloads and excessive DOM size contribute to faster loading.

Accessibility:

- **Adhering to Standards:** Following **WCAG guidelines** and using semantic HTML5 elements ensures the application is accessible to users with disabilities.
- **Accessible Interactive Elements:** Designing interactive elements with sufficient touch target sizes, providing visual feedback, and supporting keyboard navigation are crucial.

Measurement and Testing:

- **User Feedback and Analytics:** Collecting and analyzing user feedback through surveys (**Zigpoll**), monitoring behavior analytics (**Hotjar**, **FullStory**), and tracking performance metrics helps in making data-driven decisions for UX improvements.
- **Performance Models and Metrics:**
- The **RAIL model** (Response, Animation, Idle, Load) provides a user-centric structure for thinking about performance goals (e.g., responding to input in < 100ms, producing a frame in < 10ms).
- **Core Web Vitals** (**Largest Contentful Paint (LCP)**, **Interaction to Next Paint (INP)**, and **Cumulative Layout Shift (CLS)**) are key metrics that measure loading performance, interactivity, and visual stability, respectively.
- **Testing Tools:** Tools like **Google Lighthouse**, **Chrome DevTools**, and **WebPageTest** can be used to audit and measure performance against these metrics.
- **Continuous Testing:** Rigorous testing across various devices, screen sizes, operating systems, and browsers, using both automated tools and real devices, is essential.

Other Important Considerations:

- **Intuitive Navigation:** Designing a clear and logical navigation structure is vital for user experience.
- **Simplified Authentication and Personalization:** Streamlining login processes and offering personalization options can enhance user engagement.

- **Privacy and Security:** Ensuring user data privacy and application security builds trust.

- **Progressive Web App (PWA) Features:** Implementing features like ofline access and push notifications can provide a more native app-like experience.

- **Internationalization and Localization (i18n, L10n):** Adapting the application for different languages and cultures is important for a global audience.

- **Scalability and Maintainability:** Architecting the application with component-driven development and design systems ensures that UX improvements are sustainable.

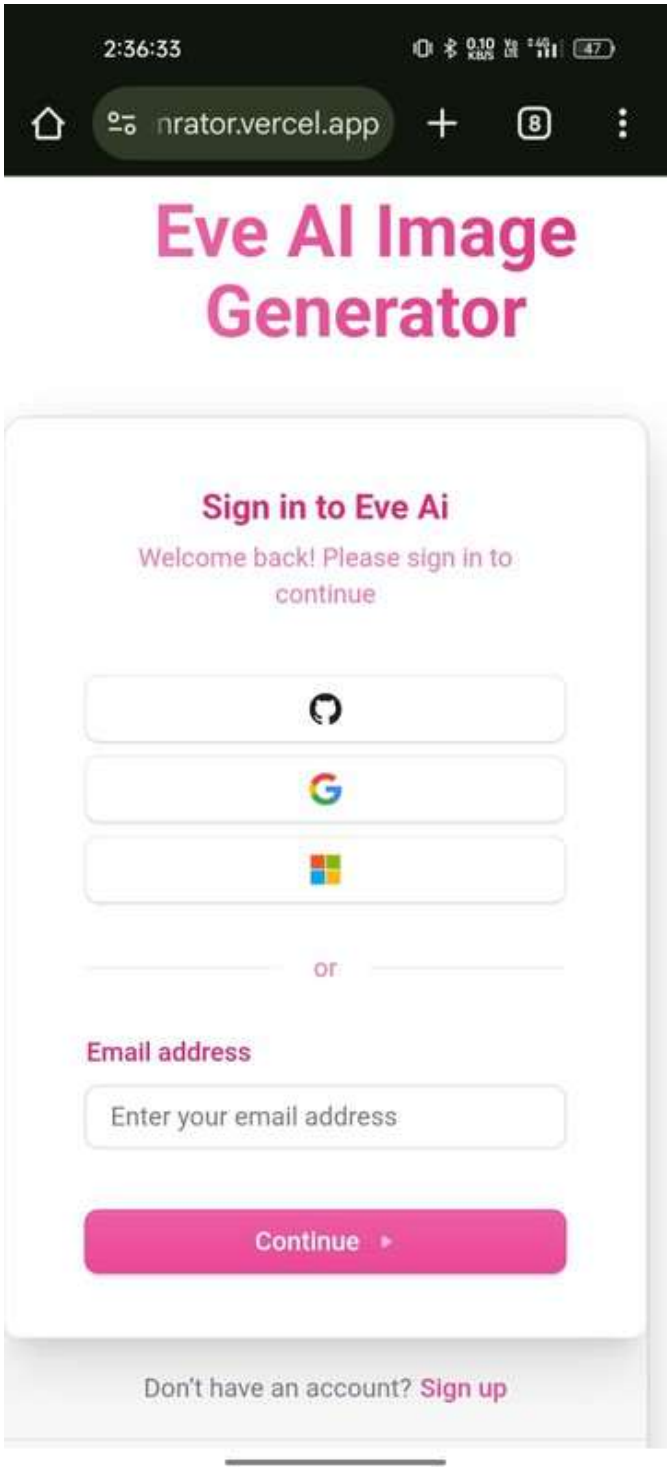
Appendix D – Testing Scenarios
Description: A structured summary of test cases conducted during development to ensure functionality, responsiveness, and robustness.

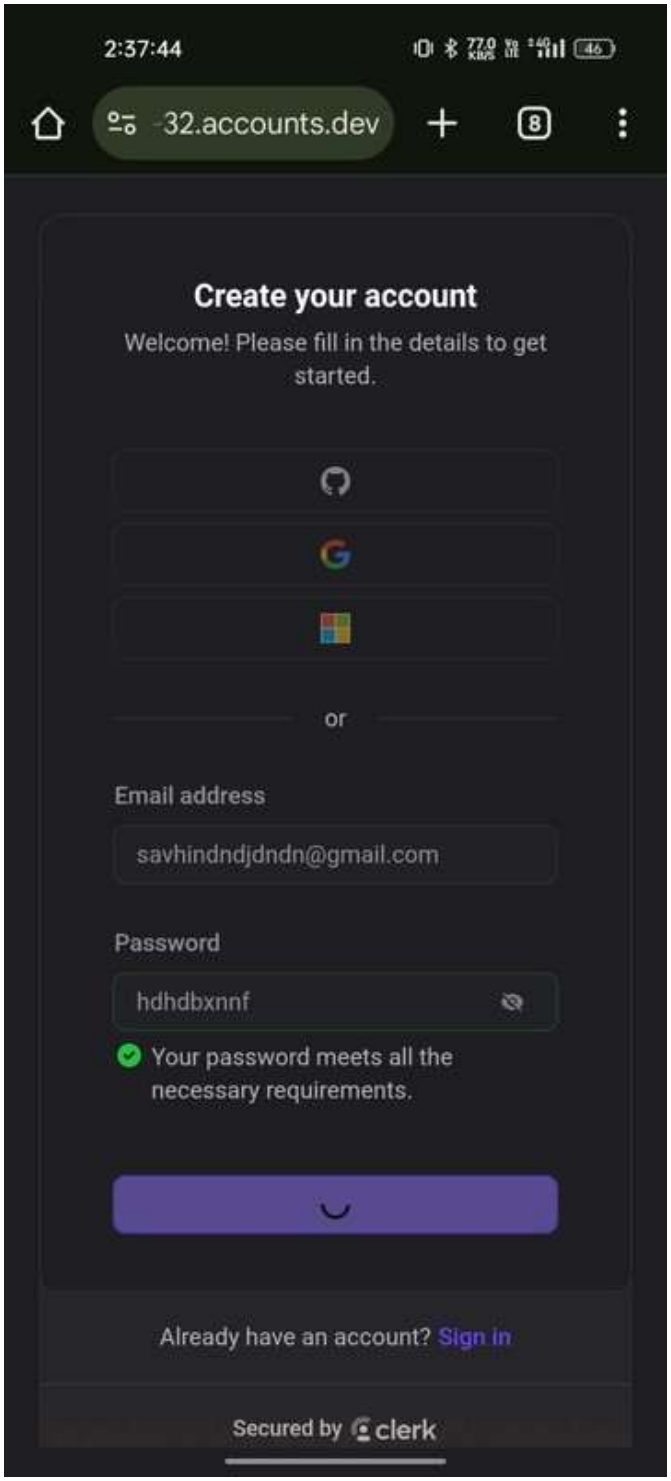
1. Image and Video Predictions

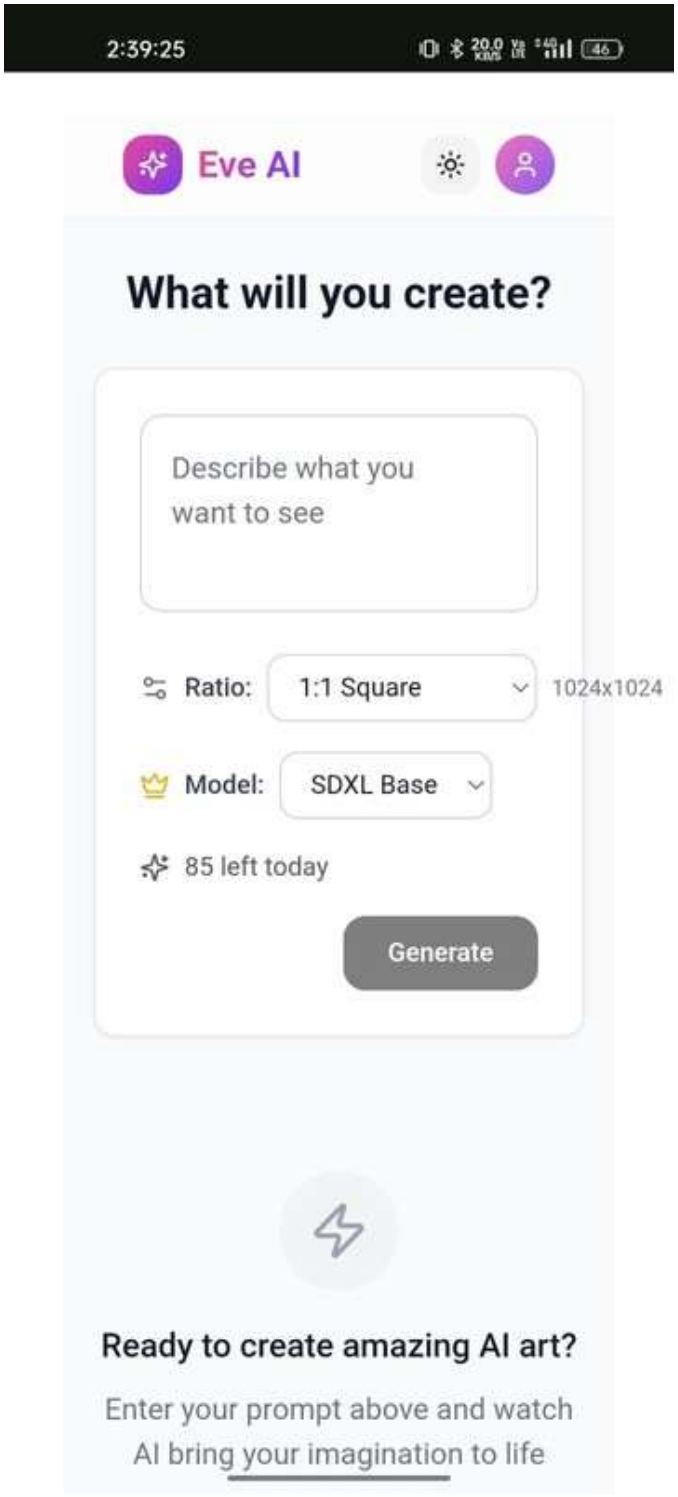
2. Login System

- When Invalid credential s

3. UI on Mobile




2:37:44
32.accounts.dev
Create your account
Welcome! Please fill in the details to get started.
or
Email address
savhindndjdndn@gmail.com
Password
hdhdbxnnf
Your password meets all the necessary requirements.
Already have an account? Sign in
Secured by clerk



Appendix E – Feature Wheel Infographic

Description: The following infographic illustrates the future roadmap of EVE AI through a radial "feature wheel", designed to visually communicate upcoming enhancements. Each feature represents a core innovation that extends the usability, accessibility, and functionality of the platform for both end-users and developers.

Feature	Description
Voice-to-Image Generation	Allows users to generate images via voice commands using speech-to-text and AI image synthesis.
3D Scene Creation	Enables the generation of immersive 3D environments and objects from text prompts.
API Access for Developers	Provides programmable interfaces so developers can integrate EVE AI's capabilities into their own tools and products.
Personal UI Assistant	Introduces an AI-powered assistant that helps users navigate, optimize prompts, and manage workflows.
Collaborative Prompting	Enables multiple users to co-create or refine prompts in real-time, supporting creativity and team-based workflows.

Purpose & Vision: This infographic acts as a **visual blueprint** of EVE AI's long-term evolution, reflecting a commitment to:

- Broader multimodal interaction
- Developer empowerment
- Enhanced personalization
- Collaborative and community-driven content creation

Purpose:To provide stakeholders with a clear visual of EVE AI's long-term vision and development trajectory, supporting both academic analysis and investor-facing documentation.

- **Conclusion**

EVE AI represents a modern integration of AI-driven creativity with user-centric design and a scalable backend architecture. This research explored the system's technical architecture, database schema, workflow pipelines, and proposed future enhancements that elevate its capabilities beyond current generative tools.

By combining technologies such as Stable Diffusion, Next.js, and secure user management via Clerk SDK, EVE AI stands as a robust foundation for both individual users and enterprise clients. The proposed roadmap—
including voice-to-image generation, personal assistants, and collaborative prompting—shows significant potential for future expansion.

Ultimately, this paper showcases EVE AI not just as a product, but as a **platform** built for evolution, aiming to empower creators, developers, and businesses alike.