ENR106 Project Problem Definition Details

Project Title: GALLERY GENIE

|  |  |  |  |
| --- | --- | --- | --- |
| **Roll No** | **Name**  **(Firstname Lastname)** | **Email Id** | **Contact No.** |
| AU2540285 | MANAS DOSHI | Manas.d@ahduni.edu.in | 8511231693 |

## Objectives of developing software/application:

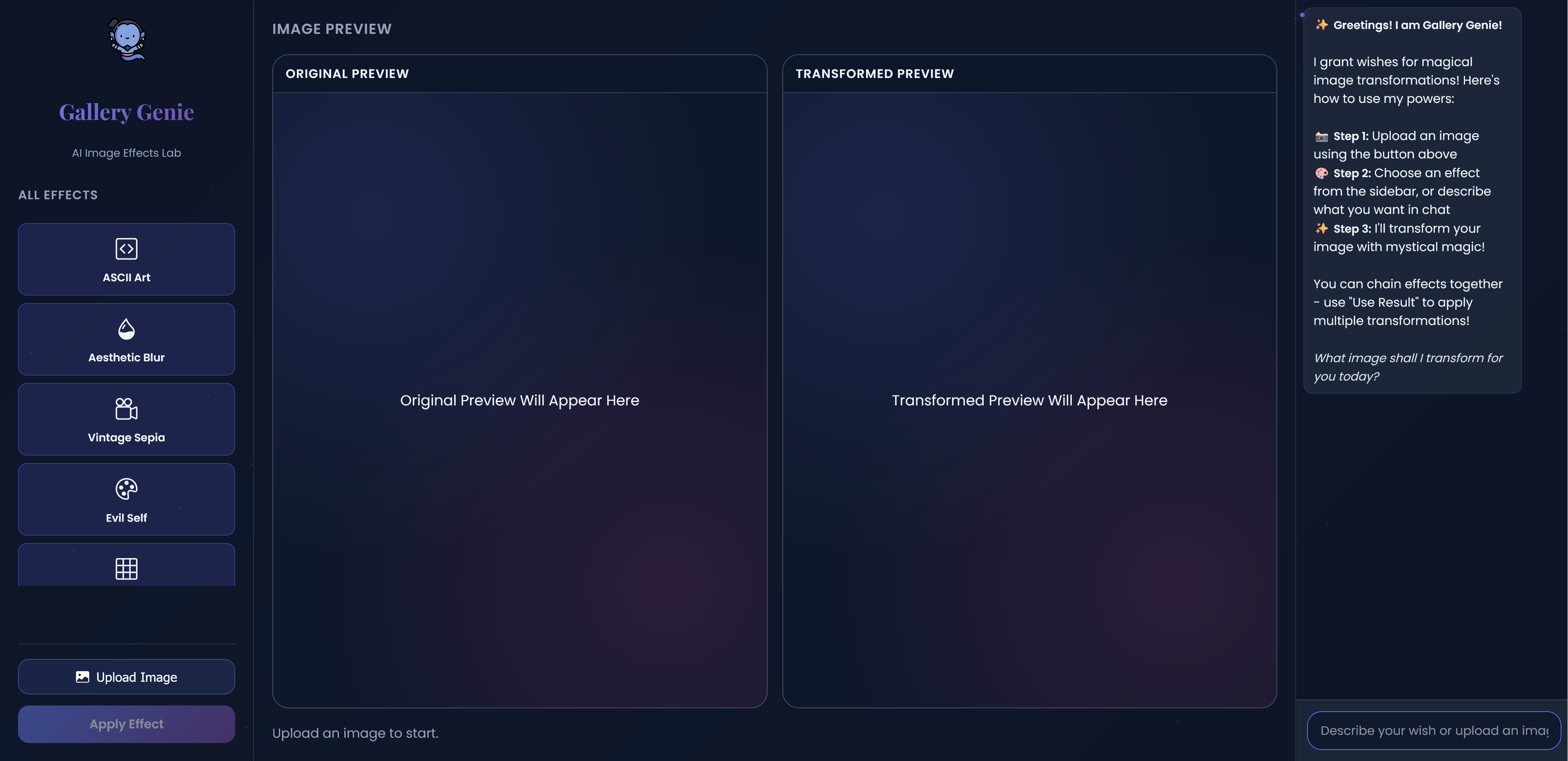
* To democratize image editing to anyone with a computer without needing a PHD in photoshop or image editing in general
* AI-Powered Intent Recognition: Implement state-of-the-art semantic understanding to accurately match user descriptions with appropriate image effects.​
* To support cross-platform web-based image viewing and processing, enhancing accessibility for users on different operating systems.
* To create a photo editing app that is not built to ostracize casual users and is done by creating a friendly UI and fun experience
* To create a local alternative to google photos and canva that puts user’s privacy first reducing risk of cyber crime

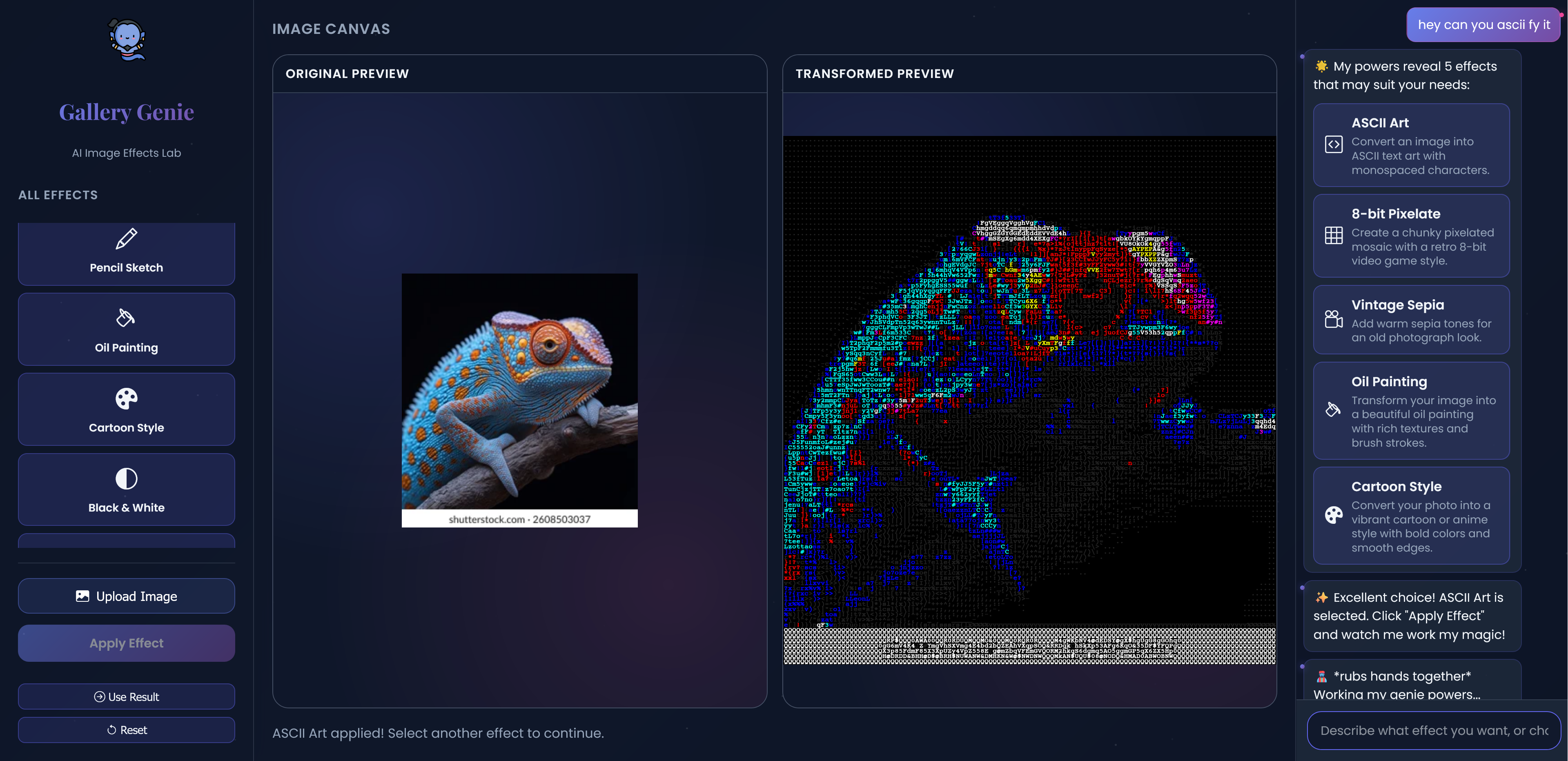
## End-Users of the software:

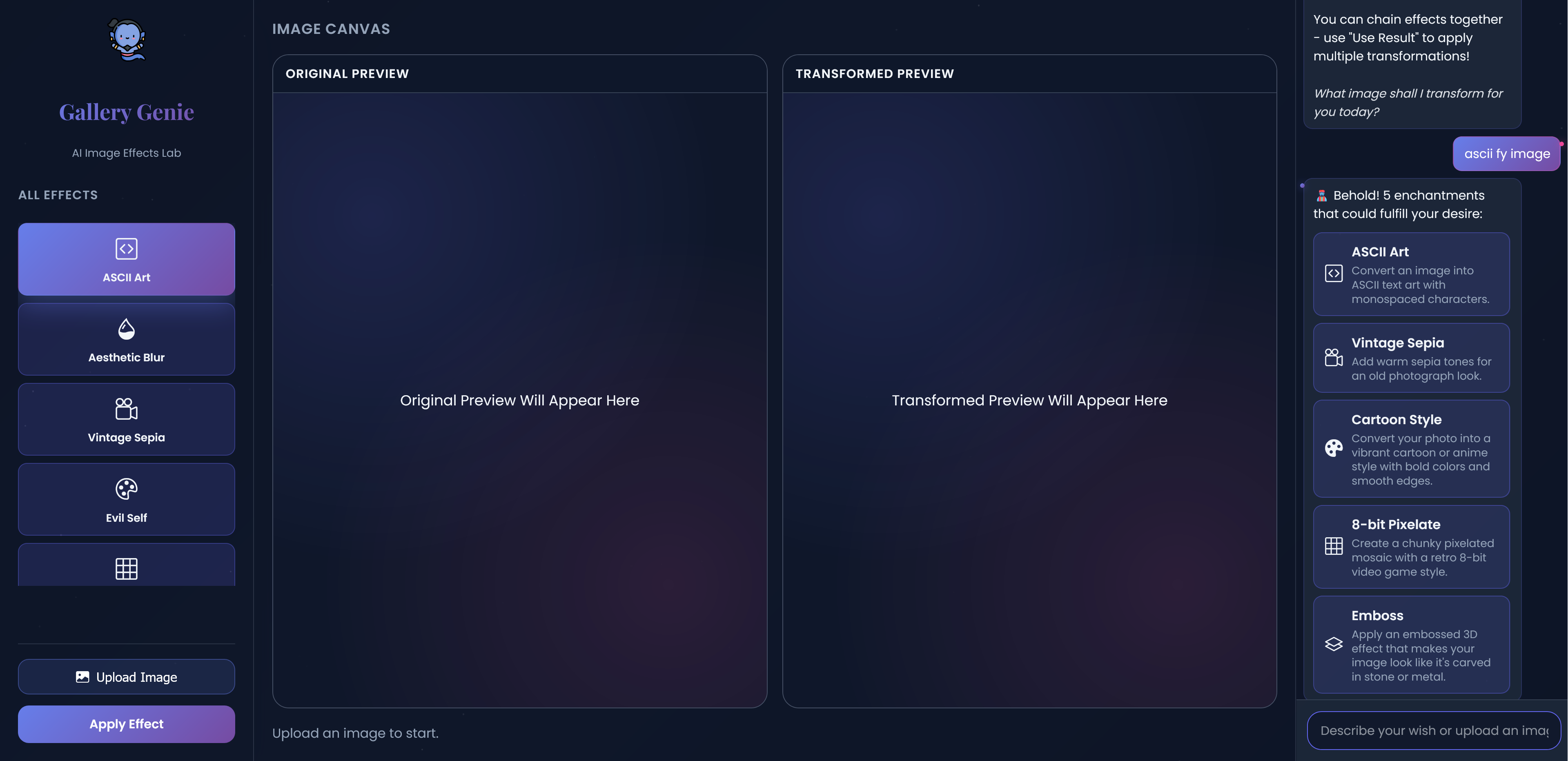
* **General Users** - Individuals interested in experimenting with image effects for creative or fun purposes, such as hobbyists or students.
* **Design Enthusiasts and** **Digital Artists** - Users who want quick transformations like sepia, sketches, or pixelation without complex software like Photoshop.
* **Content Creators** - Social media influencers, bloggers, and digital artists seeking quick aesthetic transformations for their visual content with minimal friction.
* **Students and Educators** - Academic users exploring image processing concepts through hands-on experimentation with various filters and effects.
* **Hobbyists** - Photography enthusiasts experimenting with artistic transformations and retro aesthetics for personal projects

## Listing of functionality/features/main modules:

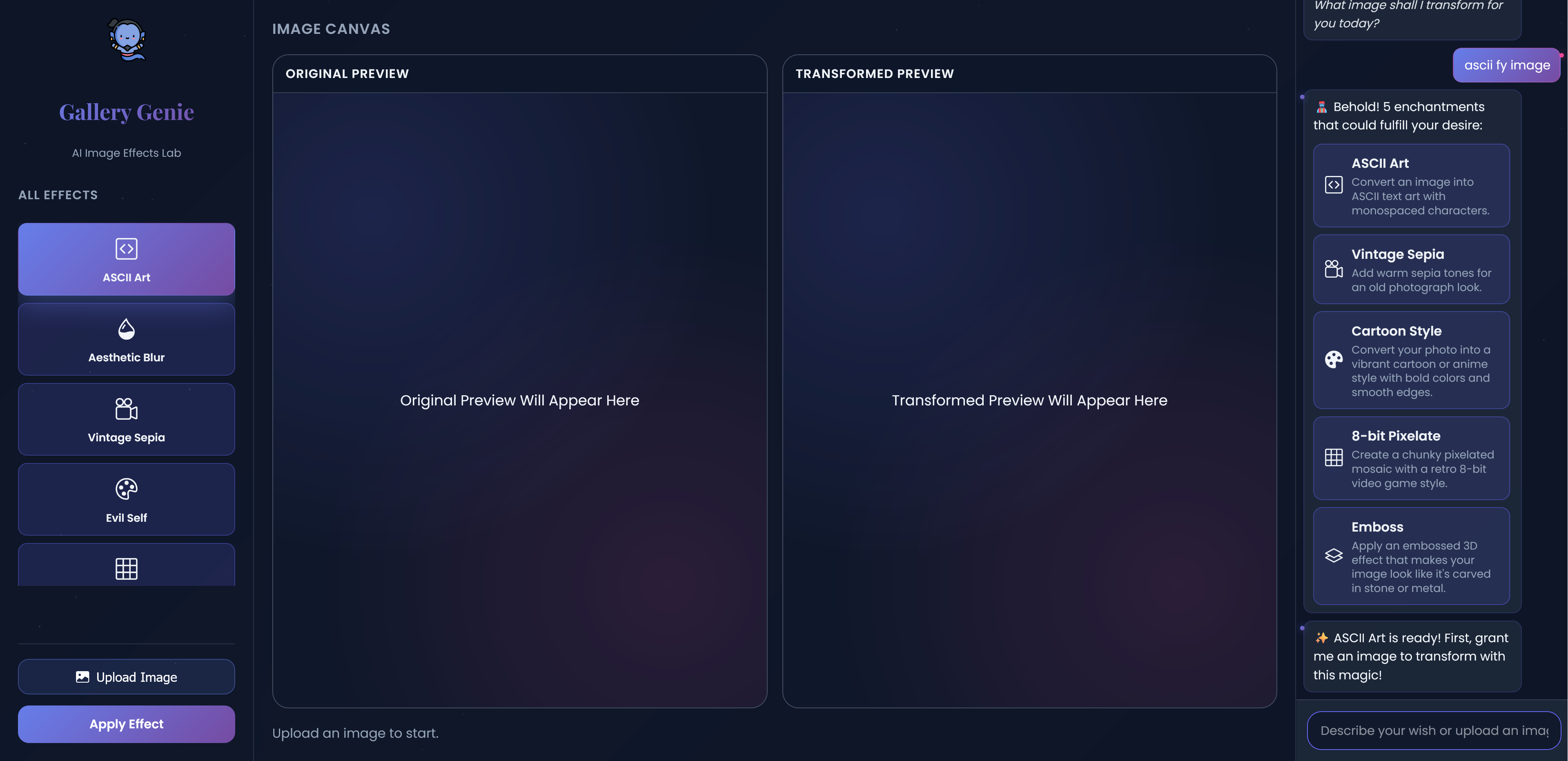
* User Interface – A Playful Mascot (that is reflected in the chat)and a visually simulating first page to engage viewer with helpful hints



* A AI chat interface on sidebar that guides user to what to like if they are stuck or if they forgot and image or if they cant find a image etc to make sure every one irrespective of their skills
* 
* And a descriptive Tool description is provided when user chats with genie to make sure they understand what they are doing
* Tool Suggestion Engine:uses semantic keyword matching to recommend image effects based on user's natural language query (e.g., "make it retro" suggests pixelate-8bit). And shows options to provide a fallback for user in case the algorithm is wrong



* Error Handling & Validation



* Image Processing Modules: Implements effects using OpenCV and ascii\_magic:
* **ASCII Art:** converts pixels to monospaced characters for retro terminals. *(ASCII Art)*
* **Aesthetic Blur:** Gaussian soft-focus for dreamy portraits. *(Blur Effect)*
* **Vintage Sepia:** warm film tones for nostalgic storytelling. *(Sepia Effect)*
* **Invert Colors (Evil Self):** high-contrast negative for cyberpunk vibes. *(Invert Effect)*
* **8-bit Pixelate:** blocky mosaic resembling classic video games. *(Pixelate Effect)*
* **Pencil Sketch:** grayscale linework mimicking hand-drawn sketches. *(Sketch Effect)*
* **Oil Painting:** bilateral filtering plus color quantization for brushy strokes. *(Oil Painting)*
* **Cartoon Style:** bilateral smoothing + adaptive edges for anime looks. *(Cartoon Effect)*
* **Black & White:** balanced grayscale conversion for timeless photos. *(B&W Effect)*
* **Emboss:** relief-style shading that adds 3D texture. *(Emboss Effect)*
* **Edge Detection:** Canny outlines for technical diagrams. *(Edge Detection)*
* **Watercolor:** layered bilateral blurs for soft, flowing pigments. *(Watercolor Effect)*
* **Posterize:** reduces palette to bold pop-art tones. *(Posterize Effect)*
* **Neon Glow:** blends heat-mapped edges for electric highlights. *(Neon Glow)*
* **Mirror Flip:** horizontal reflection for symmetry exploration. *(Mirror Effect)*
* **Color Boost:** HSV saturation uplift for vivid imagery. *(Color Boost)*
* Path Management: Remembers and prompts for last used image path, validates file existence, and generates descriptive output filenames.
* Output Handling: Saves processed images with effect-specific names and opens them in the default system viewer.
* Interactive Menu: Allows manual tool selection if suggestion is not used or score is low.
* 1. \*\*Secure Image Upload Module\*\*
* - Validates extensions (`png`, `jpg`, `jpeg`, `gif`, `bmp`).
* - Saves uploads to `static/uploads` with timestamped filenames.
* - \_Screenshot Placeholder (Upload UI)\_
* 2. \*\*Processing Engine (JFEATURE Slots)\*\*
* - Routes `/api/process` requests to OpenCV pipelines (blur, sepia, invert, pixelate, sketch, oil-painting, cartoon, monochrome, emboss, edge-detection, watercolor, posterize, neon glow, mirror, color boost, ASCII art).
* - Writes processed assets to `static/processed`.
* - \_Screenshot Placeholder (Processing Status)\_
* - \_Screenshot Placeholder (Result Gallery)\_
* 3. \*\*Genie Chat Assistant\*\*
* - Endpoint `/api/chat` interprets user prompts, suggests best-fit filters using fuzzy keyword matching, and returns motivational responses.
* - Provides fallback suggestions when confidence is low.
* - \_Screenshot Placeholder (Chat Panel)\_
* 4. \*\*Tips & Guidance Module\*\*
* - Displays filter-specific “pro tips” to teach when each effect shines.
* - Encourages iterative experimentation with “use\_processed” option for chained effects.
* - \_Screenshot Placeholder (Tips Sidebar)\_
* 5. \*\*Static Asset Delivery\*\*
* - `/processed/<filename>` serves generated artifacts for download or preview in the frontend.
* - Ensures consistent naming via `output\_path\_from` helper with timestamps.
* - \_Screenshot Placeholder (Download View)\_
* 6. \*\*Error Handling & Validation\*\*
* - Graceful JSON errors for missing files, unknown tools, or unreadable images.
* - Uses Flask `MAX\_CONTENT\_LENGTH` and allowlist to guard uploads.
* - \_Screenshot Placeholder (Error Toast)\_

Technical Details

* Technology Stack: Python 3, Flask, OpenCV, NumPy, ascii-magic, Bootstrap Icons.
* - Deployment: runs on `0.0.0.0:5000` with configurable `UPLOAD\_FOLDER` and `PROCESSED\_FOLDER`.