📱CaseCraft.ai: GenAI-Powered Phone Skins

Use case: Image generator: Synthesize and generate images

Team Member: Rakshat Shetty

Introduction

Rakshat Shetty

Python Developer with 2 years of experience in building AI/VFX pipelines, RESTful APIs, and machine learning applications. Skilled in deploying scalable backend services and developing intuitive tools using open-source AI technologies. Passionate about leveraging GenAI to create impactful user experiences.

📧 **Email: rakshat2001@gmail.com**

📞 **Phone: +91-99675-27367**

Problem Statement

- Users lack personalized and creative phone skin design options.
- Current platforms only offer fixed templates or static designs.
- We aim to use GenAI to empower users to create 100% custom skins.
- Estimated phone accessories market size: **$90+ Billion**

User Pain Points

- Lack of unique, personal designs
- Time-consuming design process
- Expensive professional services
- Limited inventory in physical stores
- No perfect fit guarantees

💡 CaseCraft.ai Solution

🎨 AI-Powered Generation

Theme-based fine-tuned VLMs for specialized design generation (anime, sports, movies, nature)

📐 Device-Specific Sizing

Precise dimensions for 50+ phone models ensuring perfect fit

🔄 Iterative Refinement

Multi-modal input: text prompts + image modifications

🏪 Digital Marketplace

Community-driven platform for sharing and monetizing designs

Key Features

Core Generation

- Multi-device support (50+ models)
- 8+ specialized theme models
- Batch generation (2-3 variations)
- High-resolution output (300 DPI)
- Iterative design refinement

User Experience

- Guest access (2 free generations)
- Authenticated personal design library
- Multiple download formats
- Mobile-responsive interface

📈 Scalability Scope

- Vertical Scaling: Larger instance sizes for high-res inference (2K)
- Horizontal Scaling: Multiple inference workers via ECS.

Tech Stack

🧠 AI Models

- Stable Diffusion
- ControlNet
- VLM models (e.g.,BLIP-2, LLaVA)
- Real-ESRGAN

🐍 Backend

- HF API & Replicate API
- FastAPI
- **PyTorch**
- PostgreSQL (Supabase)

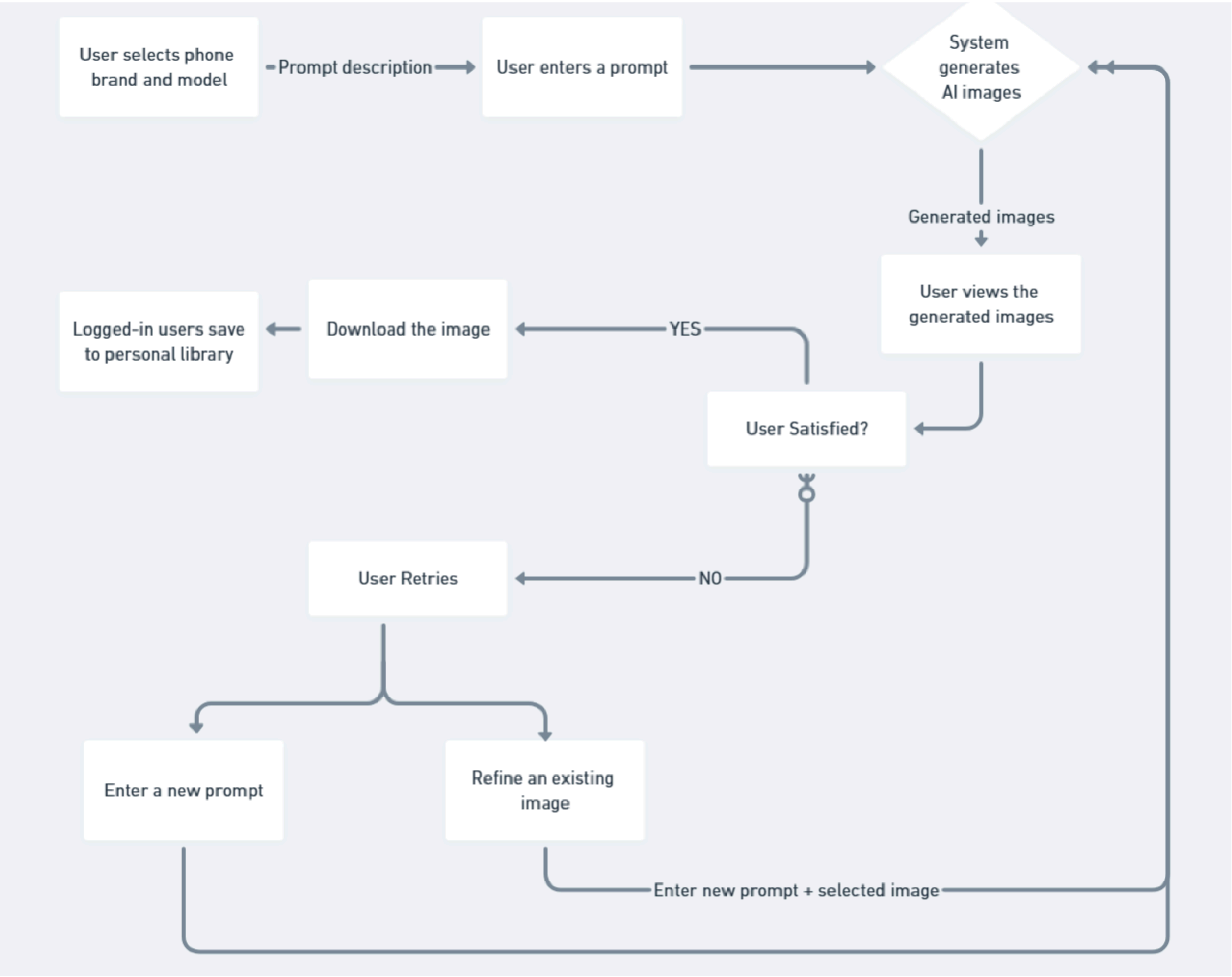
🎨 Frontend

- Vanilla Js (MVP)
- React.js (Final product)

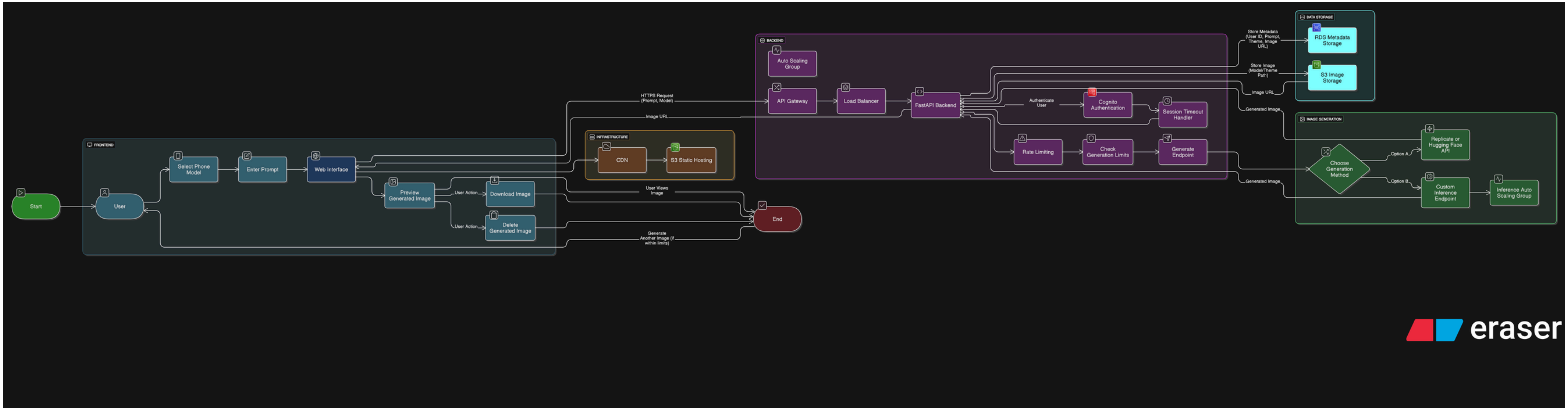
AWS Services

Service	Purpose
AWS EC2	Host FastAPI Backend
AWS Lambda	Run backend logic when a user triggers generation Eg: Make API calls to HF/Replicate API, Image resizing and conversion
API Gateway	Expose HTTPS endpoints for frontend to call Lambda
S3	Store generated images or user assets
CloudFront (CDN) (Optional; use when the app is production-ready)	Fast image delivery worldwide
Cognito	User authentication (Login, Register)
Aurora Serverless (Optional; use when the app is production-ready)	Store user info, generation history, etc.
CloudWatch	Logs and metrics for monitoring

Flowchart


User selects phone brand and model
Prompt description
User enters a prompt
System generates AI images
Generated images
User views the generated images
User Satisfied?
YES
Download the image
Logged-in users save to personal library
NO
User Retries
Enter a new prompt
Refine an existing image
Enter new prompt + selected image

Software Architecture



Implementation Plan

Phase 1

Milestone 1

- Build HTML/CSS/JS interface with dropdown for phone model and themes
- Prompt input and fetch() call to /generate endpoint
- Show 2-3 generated images with download buttons

Milestone 2

- Set up FastAPI backend with /generate route
- Integrate with Replicate/Hugging Face
- Save image metadata to RDS, image files to S3

Milestone 3

- Integrate Cognito login/logout
- Enforce image generation limit for anonymous users (max 2)
- Display logged-in user's gallery with download links

Implementation Plan

Phase 1.5

Milestone 1

- Train small LoRA models (10-30 images/theme) on AWS EC2
- Upload to Hugging Face or serve via Triton + FastAPI
- Dynamically load LoRA models based on selected theme

Phase 2

Milestone 1

- Frontend migration to React + Tailwind + shadcn/ui
- Hosting via S3 + CloudFront
- FastAPI served via AWS Lambda + API Gateway (serverless)

Milestone 2

- BLIP/LLaVA for prompt improvement or captioning
- User gallery with S3 links and metadata from RDS

Implementation Plan

Phase 3

Milestone 1

- Logged-in users can publish their generated designs
- Add payment gateway integration like Razorpay for selling styles
- Marketplace with likes/download tracking and user profiles

Cost Estimation

🧮 Overall Budget Summary

Phase	Description	Est. Monthly Cost (USD)
🧪 MVP (FastAPI + Vanilla JS)	Basic working version using Replicate/HF API	~$15–25/month
🎨 LoRA Theme Fine-Tuning	Training fine-tuned theme models (optional)	~$90–120 one-time
🚀 Production Infra 1 Month	Fully scaled version with React + Lambda	~$100–160/month
🧮 Total Estimation (3 Months)	MVP + Training + 1 Month Production	~$250–320 total

Cost Estimation

- MVP (3–Weeks)

Service	Usage	Est. Cost (USD)
EC2 (t3.micro)	FastAPI backend	~$5/month (Free Tier if eligible)
S3	Image storage (up to 10 GB)	~$1–2/month
RDS	db.t3.micro PostgreSQL	~$3/month
Cognito	Auth	Free Tier
Replicate API	Image generation (1000 req)	~$20 (approx 0.02/img)

Cost Estimation

◆ LoRA Theme Training (One-Time)

Service	Usage	Est. Cost (USD)
EC2 (g4dn.xlarge)	90 hrs @ ~$1/hr	~$90
S3	Model storage (~5–10 GB)	~$0.50
Hugging Face	Hosting trained model (free tier up to 30GB)	Free (or ~$5 for PRO)
Subtotal	1–2 themes	~$90–120 one-time

Cost Estimation

- Production-Ready Infra (1 Month for Launch)

Service	Usage	Est. Cost (USD)
EC2 (g4dn or custom inference)	Self-hosted LoRA inference	~$40–60
Lambda + API Gateway	Serverless backend	~$5–10
RDS	Production DB (db.t3.micro or better)	~$15/month
S3	Scaled image storage (30 GB)	~$0.70
CloudFront	CDN for images (100 GB delivery)	~$8.50
CloudWatch/Logs	Monitoring	~$2–5
React Hosting	S3 + CloudFront or Vercel	~$5–10
Subtotal	Prod Infra	**~$100–160/month**

Potential Impact & Limitations

🌍 Potential Impact

- Democratize design — make anyone a designer
- Monetization avenue for creators
- Personalized, creative phone skins
- Great demo of VLMs + Diffusion models for real-world personalization

✅ Benefits

- Scalable and extensible architecture
- High user engagement through iteration
- Monetization via marketplace and print options

🧷 Dependencies

- Hugging Face Inference API or Diffusers library
- Supabase & AWS Services
- Pretrained VLMs & fine-tuning infrastructure
- OpenCV / PIL for resizing
- Phone Model Dimensions and shape dataset

⚠️ Limitations

- GPU inference can be expensive at scale
- Accuracy of dimensions and cutout placement needs manual validation
- Image generation might take upto 30 seconds
- Initially only English will be supported

Ethical Considerations

- 🛡️ Prevent generation of NSFW, violent, or copyrighted content
- 🔐Maintain user privacy and data security
- ⚖️ AI models may exhibit training bias Mitigation: Diverse training data, bias testing, user feedback
- © Users might generate copyrighted content Mitigation: Content filtering, DMCA process, watermarking

🚀 Ready to Transform Phone Personalization?

CaseCraft.AI - Where AI Meets Personal Style

Thank you for your consideration
Impetus & AWS GenAI Hackathon 2025