Arclight

AI-Powered Image Editor

A Report for an AI Powered Mobile Photo Editor Envisioned for 2030

Frontend + Backend Architecture Report for Inter IIT Tech Meet 14.0

Framework:	React Native 0.81.5 with Expo
Language:	TypeScript (100% coverage)
Backend:	Node.js + Express 5.1.0 + MongoDB
AI Integration:	Google Gemini + Custom Models
Version:	1.0.0

December 2024
Arclight Development Team

Contents

Part I
Frontend Documentation

1 Project Overview

Arclight is a production-ready, AI-powered image editing application that revolutionizes mobile photo editing through natural language processing and advanced AI capabilities.

1.1 Core Capabilities

- **AI Enhancement**: Face restoration, denoising, deblurring, relighting
- **Object Manipulation**: Intelligent object/background removal
- **Style Transfer**: Apply artistic styles from reference images
- **Natural Language**: Gemini AI for text-based editing
- **Complete History**: Full timeline with undo/revert

1.2 Dual Workspace Architecture

Aspect	Labspace (Sequential)	Flowspace (Layer-Based)
Purpose	Apply AI operations sequentially	Professional multi-layer based editing
Key Features	Real-time filters, operation history, gesture navigation	Layer management and background harmonization
Use Case	Quick AI-powered enhancements	Complex composite editing

2 UI Design Philosophy

The Arclight interface prioritizes **clarity**, **simplicity**, and **user focus**.

2.1 Design Principles

1. Minimalist Interface

- Clean layouts with generous white space
- Contextual controls appear only when relevant
- Visual hierarchy through typography and spacing

2. Intelligent Feedback Systems

- Custom animated loaders with educational fun facts
- Real-time progress indicators during uploads
- Contextual alerts with smooth auto-dismiss animations
- Visual markers for interactive operations
- Image state labels showing “Original” vs “Updated”

2.2 Micro-Interactions & Animation

- Spring-based animations using React Native Reanimated for 60fps smoothness
- Gesture recognition with custom PanResponder
- Physics-based transitions

2.3 Typography & Iconography

Element	Implementation
Primary Font	Geist Mono
Display Font	Dela Gothic One
Icon System	Lucide React Native
Weight Hierarchy	Regular → Medium → SemiBold → Bold

3 Technical Architecture

3.1 Technology Stack

Category	Technology	Purpose
Framework	React Native 0.81.5	Cross-platform mobile development
Navigation	Expo Router 6.0.15	File-based routing with deep linking
State	React Hooks + Context	Lightweight global state management
Animations	Reanimated 4.1.1	60fps native-driver animations
Storage	AsyncStorage	Persistent local data
Networking	Custom API Service	Centralized client with retry logic

3.2 Design Patterns

1. **Context API for Global State**: ThemeContext and SideBarContext
2. **Service Layer Pattern**: Centralized ApiService with JWT injection
3. **Custom Hooks**: `useAlert`, `useTheme`
4. **Component Composition**: Small, focused components

3.3 Project Structure

```
frontend/
 app/                    # Expo Router file-based routing
    (auth)/             # Authentication screens
    (app)/              # Main screens
 components/             # Reusable UI components
 context/                # React Context providers
 services/api.ts         # Centralized API client
 constants/              # Colors, API configuration
 hooks/                  # Custom React hooks
 utils/                  # Utility functions
```

4 Key Features & Implementation

4.1 AI Operations

Operation	Description
Natural Relighting	Adjusts scene lighting with configurable brightness
Face Restoration	Enhances facial details using CodeFormer model
Enhancement	Denoise or deblur images with AI
Subject/Background Removal	Intelligent segmentation and removal
Object Removal	Interactive tap-to-remove with SAM + LaMa
Style Transfer	Apply artistic style from reference image

4.2 Natural Language Editing with Gemini AI

Users type natural language prompts like "remove the person" or "enhance the colors." Gemini AI analyzes intent, selects the appropriate operation, and executes automatically.

4.3 Real-Time GPU Filters

- Brightness (-50 to +50)
- Contrast (-50 to +50)
- Saturation (-50 to +50)
- Temperature/Warmth
- Shadows/Highlights
- Sharpen

4.4 Complete History & Timeline

Every AI operation is tracked with input/output images, operation type, parameters, timestamp, and revert capability.

5 Engineering Challenges & Solutions

5.1 Network Reliability

Problem: Frequent network errors during image uploads.

Solution: Exponential backoff with 5 retries, extended 120s timeout, real-time feedback. Upload success rate improved from 60% to 95%.

5.2 Precise Coordinate Mapping

Problem: Object removal required accurate coordinate transformation.

Solution: Custom algorithm accounting for aspect ratio and letterboxing with immediate visual feedback.

5.3 State Management Complexity

Solution: Clear state hierarchy, AsyncStorage synchronization, history key pattern for forced refresh.

6 Quality Metrics

Metric	Value
Initial Load Time	<3.5 seconds
Upload Success Rate	95%
Theme Switch Time	<100ms
Gesture Recognition	95%+ accuracy

Part II
Backend Documentation

7 Backend Overview

Attribute	Value
Tech Stack	Node.js + Express 5.1.0 + MongoDB
AI Integration	8 Docker Containers + Google Gemini
Database	MongoDB Atlas (Cloud)
Image Storage	Cloudinary CDN
Authentication	JWT + Google OAuth 2.0
API Version	v1
Base URL	/api/v1/adobe-ps

8 Backend Architecture

8.1 Directory Structure

```
backend/
 server.js              # Entry point
 app.js                 # Express configuration
 config.js              # Environment loader
 config/
    cloudinary.js      # Cloudinary configuration
 controllers/           # Business logic
 models/                # MongoDB schemas
 routes/                # API endpoints
 middleware/            # Auth, upload, error handling
 utils/                 # Helpers
 AI/                    # Gemini AI integration
 image/                 # Temporary storage
```

8.2 Technology Stack

Category	Technology	Version
Runtime	Node.js	20.x LTS
Framework	Express.js	5.1.0
Database	MongoDB (Mongoose ODM)	8.19.4
Authentication	JSON Web Tokens	–
Image Processing	Sharp	0.34.5
File Upload	Multer	2.0.2
Cloud Storage	Cloudinary	–
AI/ML	Google Gemini API	2.0-flash-exp
Password Hashing	Bcrypt	6.0.0
Email	Nodemailer	7.0.10
HTTP Client	Axios	1.13.2

9 Database Models

9.1 User Model

Field	Type	Description
name	String	User's display name
email	String	Unique email address
password	String	Bcrypt hashed password
image	String	Profile image URL
isVerified	Boolean	OTP verification status
otp	String	4-digit verification code
otpExpiry	Date	OTP expiration (10 minutes)
accounts	Array	OAuth provider accounts
settings.maxVersions	Number	Version limit (1-15, default: 10)

9.2 Project Model (Layer-Based)

Field	Type	Description
user	ObjectId	Reference to User
title	String	Project name
originalImage	Object	Original image metadata
layers	Array[ObjectId]	References to Layer documents
canvas	Object	Width, height, backgroundColor
thumbnail	Object	Preview image
history	Array	Action snapshots (max: 50)

9.3 Layer Model

Field	Type	Description
project	ObjectId	Parent project reference
name	String	Layer name
type	Enum	background, foreground, object, text, adjustment
imageUrl	String	Cloudinary image URL
publicId	String	Cloudinary identifier
maskUrl	String	Optional mask image URL
order	Number	Layer z-index
visible	Boolean	Visibility toggle
locked	Boolean	Edit lock
opacity	Number	0-100 opacity value
blendMode	Enum	normal, multiply, screen, overlay, soft-light
position	Object	{x, y} coordinates
dimensions	Object	{width, height}
transformations	Object	rotation, scaleX, scaleY, flipX, flipY

9.4 AIProject Model (Sequential Editing)

Field	Type	Description
user	ObjectId	Reference to User
title	String	Project title
description	String	Project description
originalImage	Object	Initial uploaded image
currentImage	Object	Latest processed image
operations	Array	Operation history with inputs/outputs
thumbnail	Object	Latest output as thumbnail
status	Enum	active, archived, deleted

9.5 Image Model

Field	Type	Description
user	ObjectId	Reference to User
publicId	String	Unique Cloudinary identifier
imageUrl	String	Full Cloudinary URL
format	String	Image format (jpg, png, etc.)
width	Number	Image width in pixels
height	Number	Image height in pixels
size	Number	File size in bytes

10 API Routes

10.1 Route Overview

Route Group	Endpoints	Auth
/auth	signup, login, verify-otp, resend-otp, google, logout	No
/images	upload, upload-multiple, crop, delete, get	Yes
/projects	CRUD for layer-based projects	Yes
/layers	CRUD, reorder, duplicate layers	Yes
/ai-projects	CRUD for sequential AI projects	Yes
/ai	8 AI operations	Yes
/gemini	AI analysis, prompts, auto-enhance	Yes
/settings	User preferences	Yes

10.2 Authentication Routes

Method	Endpoint	Description
POST	/signup	Register new user
POST	/login	Authenticate with credentials
POST	/verify-otp	Verify OTP sent to email
POST	/resend-otp	Resend OTP if expired
POST	/google	Google OAuth authentication
GET	/logout	Clear JWT cookie

10.3 Image Routes

Method	Endpoint	Description
POST	/upload	Upload single image
POST	/upload-multiple	Upload up to 2 images
PATCH	/crop	Crop image in place
DELETE	/:publicId	Delete image
GET	/:publicId	Get image metadata

10.4 AI Operations Routes

Method	Endpoint	Description
POST	/relight	Low-light image enhancement
POST	/enhance	Denoise or deblur image
POST	/face-restore	Face restoration
POST	/style-transfer	Artistic style transfer
POST	/remove-background	Remove image background
POST	/object-removal	Remove objects with inpainting
POST	/separate-layers	Auto-separate into layers
POST	/replace-background	Replace with new background

10.5 Gemini AI Routes

Method	Endpoint	Description
POST	/analyze	Analyze image with Gemini AI
POST	/prompt	Process natural language prompt
POST	/auto-enhance/:projectId	Get enhancement recommendations
POST	/apply-enhancements/:projectId	Apply recommendations

11 AI Processing Pipeline

11.1 Docker-Based AI Models

Container Name	Model	Function
lowlight-service	LytNet	Low-light enhancement
codeformer-service	CodeFormer	Face restoration
nafnet-service	NAFNet	Denoising and deblurring
style-transfer-service	PCA	Artistic style transfer
background-removal-service	U2Net/rembg	Background removal
pct-net-service	PCT-Net	Background harmonization
object-masking-service	SAM	Object segmentation
object-remover-service	LaMa	Object inpainting

11.2 Processing Flow

1. Download image from Cloudinary to temp directory
2. Check/start Docker container if not running
3. Copy image into container via `docker cp`

4. Execute processing command via `docker exec`
5. Copy result back from container
6. Upload processed image to Cloudinary
7. Clean up temporary files
8. Return new Cloudinary URL to client

11.3 Google Gemini Integration

- **Model**: gemini-2.0-flash-exp
- **Functions**: Image classification, subject detection, edit suggestions, auto-enhancement analysis, quality issue detection, natural language prompt processing

12 Editing Workflows

12.1 Workflow 1: Layer-Based Editing

```
Create Project -> Upload Image -> Auto-separate
    -> Add layers -> Adjust properties
    -> Reorder/duplicate -> Export composite
```

Features: Multi-layer composition, blend modes, transformations, visibility controls, history tracking, layer duplication.

12.2 Workflow 2: AI Sequential Editing

```
Create AIProject -> Upload Image -> Apply AI Operation 1
    -> Apply AI Operation 2 -> Undo/Revert any time
```

Features: Chain AI operations, full operation history, undo last operation, revert to any state, timeline visualization.

13 Security Implementation

Feature	Implementation
Password Hashing	Bcrypt with salt rounds: 12
JWT Authentication	7-day expiration, HttpOnly cookies
CSRF Protection	SameSite=lax cookie attribute
XSS Prevention	HttpOnly cookies
File Upload Security	Multer validation, 20MB limit
Email Verification	4-digit OTP, 10-minute expiration
Authorization	User-scoped queries
HTTPS in Production	Secure cookie flag enabled

14 External Services

Service	Purpose	Config
MongoDB Atlas	Cloud database	DATABASE, DATABASE_PASSWORD
Cloudinary	Image CDN	CLOUD_NAME, API_KEY, API_SECRET
Google Gemini	AI analysis	GEMINI_API_KEY
Gmail SMTP	OTP emails	EMAIL_USER, EMAIL_PASSWORD
Google OAuth	Social auth	client_id, client_secret

15 Error Handling

Error Type	Handler
MongoDB CastError	Invalid ObjectId formatting
Duplicate Fields	Unique constraint violations
Validation Errors	Schema validation failures
JWT Invalid	Invalid token signature
JWT Expired	Token expiration
Cloudinary Errors	Image processing failures
Multer Errors	Upload size/type violations

16 Conclusion

Arclight represents a complete, production-ready image editing platform combining:

- Innovative natural language editing
- Dual workspace system (sequential + layer-based)
- 8 specialized AI models via Docker
- Intuitive mobile-first user experience

Inter IIT Tech 14.0 - Adobe Problem Statement

December 2024