Project 7: Image-Generating Chatbot

Objective (Why?)

Build an intelligent image-generating chatbot that creates high-quality images from natural language descriptions. This project introduces multimodal AI capabilities to your existing chat platform from Project 6. You will practice:

- ➢ Multimodal AI Integration: Combining text processing with image generation APIs
- ➢ Asynchronous Processing: Handling long-running image generation tasks efficiently
- ➢ Image Persistence: Storing generated images directly within the application's database
- ➢ Real-time UI Updates: Implementing progress indicators and streaming responses

Core Requirements (Must-have)

Component	Requirement
Image Generation	Integrate OpenAI DALL·E 3 or Stable Diffusion for text-to-image conversion with high-quality outputs
Async Processing	Implement background task processing for image generation with real-time progress updates
Storage & Persistence	Store generated images as Base64 strings in the PostgreSQL database to keep the architecture self-contained for Phase 1.
Chat Integration	Seamlessly integrate image generation into existing chat platform from Project 6.
Gallery Management	Store, organize, and retrieve generated images with search and download functionality

Milestone 1: Image Generation Infrastructure & API

Deliverables:

- DALL-E 3 API integration with secure key management
- Async task processing with Celery and Redis
- Image metadata and Base64 content schema in database
- Basic image generation endpoint with error handling

Review Requirements (Must Pass to Proceed):

- Security Review: API key security, input validation, rate limiting
- Architecture Review: Scalable async processing architecture
- Performance Review: Efficient image generation and storage

Milestone 2: Real-Time Communication & Chat

Deliverables:

- Progress tracking using polling or Server-Sent Events (SSE)
- Enhanced chat interface with image generation commands
- Image display and gallery components
- Progress indicators and loading states
- Integration with existing Project 6 chat system

Review Requirements (Must Pass to Proceed):

- AI Integration Review: Seamless image generation within conversations
- Architecture Review: Clean chat and image generation integration
- Performance Review: Real-time communication efficiency

Milestone 3: Production Features & Creative Tools

Deliverables:

- Advanced image management (organize, search, download)
- Creative prompt assistance and generation history
- Rate limiting and cost management systems
- Performance optimization and caching strategies
- Comprehensive testing and production deployment

Review Requirements (Must Pass for Project Completion):

- AI Integration Review: Creative AI workflow optimization
- Architecture Review: Complete creative platform architecture
- Security Review: Production security and cost control
- Performance Review: Optimized image generation performance

Milestone Progression Rules:

- Cannot advance to next milestone without passing all review requirements
- Flexible timing allows for learning at individual pace

- Quality gates ensure each milestone meets professional standards
- Mentor support available for concept clarification and review failures

Measurable Goals & Review Template Compliance

Primary Objectives

- Creative AI Mastery: Pass AI Integration Review with 9.0/10+ score (image generation)
- Architecture Excellence: Pass Architecture Review
- Security Standards: Pass Security Review (API security)
- Performance Optimization: Sub-30s image generation with real-time feedback
- Code Quality Standards: Pass Code Quality Review

Performance Standards

- Image Generation: < 30 seconds for typical DALL-E requests
- Real-Time Updates: < 1 second update latency for progress indicators
- Storage Performance: < 2 seconds for image upload to cloud
- Chat Integration: Seamless image commands with immediate feedback

Task Tracking & Project Management Integration

Epic: Project 7 - Image-Generating Chatbot

Epic ID: P7-IMAGE-CHAT
Priority: High

Milestone 1: Image Generation Infrastructure

Feature 7.1: Core Image Generation System

Task ID: P7-M1-CORE
Priority: Critical
Dependencies: None

Sub-tasks:

- P7-M1-CORE-01: DALL-E API integration
  - Description: Secure OpenAI DALL-E 3 API implementation
  - Acceptance Criteria: Working image generation with error handling
- P7-M1-CORE-02: Async processing setup
  - Description: Celery and Redis configuration for background tasks
  - Acceptance Criteria: Non-blocking image generation workflow
- P7-M1-CORE-03: Cloud storage integration
  - Description: Cloudinary or AWS S3 setup for image storage
  - Acceptance Criteria: Automated image upload and retrieval

Feature 7.2: Image Management System

Task ID: P7-M1-MGMT
Priority: High
Dependencies: P7-M1-CORE completion

Sub-tasks:

- P7-M1-MGMT-01: Image metadata database
  - Description: Schema for image metadata, prompts, and user associations
  - Acceptance Criteria: Complete image tracking and organization
- P7-M1-MGMT-02: Generation history tracking
  - Description: Store and retrieve user's image generation history

- Acceptance Criteria: Full generation history with search capabilities

Milestone 2: Real-Time Communication & Chat Integration

Feature 7.3: Real-Time Generation Interface

Task ID: P7-M2-REALTIME
Priority: High
Dependencies: P7-M1-MGMT completion

Sub-tasks:

- P7-M2-REALTIME-01: Real-time progress tracking
  - Description: Progress updates for image generation using polling or SSE
  - Acceptance Criteria: Live progress tracking and notifications
- P7-M2-REALTIME-02: Chat interface enhancement
  - Description: Image generation commands within chat conversations
  - Acceptance Criteria: Natural language image requests in chat

Feature 7.4: Image Display & Gallery

Task ID: P7-M2-GALLERY
Priority: Medium

Sub-tasks:

- P7-M2-GALLERY-01: Image display components
  - Description: Gallery view with image preview and metadata
  - Acceptance Criteria: Responsive image gallery with search/filter
- P7-M2-GALLERY-02: Download and sharing features
  - Description: Image download, sharing, and export functionality
  - Acceptance Criteria: Multiple format downloads and sharing options

Milestone 3: Production Features & Creative Tools

Feature 7.5: Creative AI Enhancement

Task ID: P7-M3-CREATIVE
Priority: Medium

Sub-tasks:

- P7-M3-CREATIVE-01: Prompt assistance tools
  - Description: AI-powered prompt suggestions and optimization
  - Acceptance Criteria: Intelligent prompt enhancement features
- P7-M3-CREATIVE-02: Rate limiting and cost management
  - Description: API usage monitoring and cost control systems
  - Acceptance Criteria: Prevent API abuse and manage generation costs
- P7-M3-CREATIVE-03: Performance optimization
  - Description: Caching, optimization, and production readiness
  - Acceptance Criteria: Meet all performance standards

Phase Progression Requirements

Project 7 → Project 8 Advancement Requirements

Mandatory Review Template Compliance that covers:

- AI Integration Review & Architecture Review
- Security Review & Performance Review

Project 8 Preparation Requirements

- Data Analysis Libraries: pandas, numpy, matplotlib, seaborn
- Excel Integration: openpyxl, xlsxwriter for Excel file processing
- Statistical Analysis: Basic statistics and data visualization
- Advanced Data Processing: Complex Excel operations and analysis patterns

Enhanced Timeline Benefits Excellence:

- Solid foundation for complex async processing and API integration

- Proper time for cloud storage and real-time communication setup
- Quality chat integration without rushing UI development
- Production-ready optimization, testing, and professional polish

Technical Specifications

Image Generation Pipeline

- Implement OpenAI DALL·E 3 integration with proper prompt engineering and error handling
- Create background task processing using Celery worker queues for non-blocking operations
- Build real-time progress communication using polling or Server-Sent Events (SSE) for user feedback

Database Storage

- Design the database schema to store the generated image directly as a Base64 encoded string in a `TEXT` field.
- Store essential metadata alongside the image, such as the prompt, user association, and creation timestamp.
- Implement helper functions to handle the encoding (binary to Base64) and decoding (Base64 to image for display) process.

Chat Interface Enhancement

- Extend existing Project 6 chat system to handle image generation requests seamlessly
- Create responsive image display components with gallery view and individual image management
- Implement download functionality supporting multiple formats (PNG, JPEG, WebP)

Project Structure

- Extend the existing Project 6 architecture with image generation capabilities
- Add new backend services for image processing, cloud storage, and async task management

- Enhance React frontend with image display components, progress indicators, and gallery interface
- Integrate Redis for caching and Celery for background task processing

Advanced Features (Stretch Goals)

- WebSocket Communication: Implement real-time WebSocket progress updates for smoother UX
- Style Variations: Support different artistic styles (photorealistic, cartoon, abstract)
- Prompt Enhancement: Auto-improve user prompts for better image quality
- Batch Generation: Generate multiple variations from a single prompt
- Image Editing: Support modification requests (change colors, add objects)
- Cost Optimization: Implement intelligent caching and rate limiting

Deliverables

1. Complete Image Generation System with text-to-image functionality
2. Real-time Progress Tracking via polling or SSE
3. IMAGE_GALLERY.md with sample generated images and feature demonstrations
4. GitHub Repository with complete integration into existing chat system
5. Live Demo showing end-to-end image generation workflow

Evaluation Rubric (100 Points)

Criterion	Points	Details
Image Generation	35 pts	DALL·E 3 integration working correctly High-quality image outputs Proper error handling
Async Processing	25 pts	Background task processing functional Real-time progress updates Non-blocking user interface

Storage & Gallery	20 pts	Cloud storage integration working Image gallery with search/download Efficient metadata management
Chat Integration	15 pts	Seamless integration with existing chat Intuitive user experience Responsive design
Code Quality	5 pts	Clean architecture and documentation Performance optimization Security best practices

Performance Requirements

Generation Performance

- Image generation completion: < 30 seconds for standard 1024x1024 images
- Real-time progress updates: < 1 second response time for updates
- Gallery loading: < 2 seconds for 50+ images

Quality Standards

- Image resolution: Support 1024x1024 minimum, 1792x1024 for wide formats
- Generation success rate: > 95% for valid prompts
- Storage optimization: Automatic compression and format conversion

Testing Scenarios

Image Generation Testing

- Generate images from simple text prompts
- Handle complex, detailed descriptions
- Test various artistic styles and formats
- Verify error handling for invalid prompts
- Validate generation time and quality

System Integration Testing

- Seamless chat-to-image workflow
- Real-time progress updates working
- Image storage and retrieval functional
- Gallery search and filtering operational
- Download functionality for multiple formats

Performance Testing

- Multiple concurrent generation requests
- Large file handling and optimization
- API rate limiting and error recovery
- Mobile responsiveness and touch interactions
- Cost monitoring and usage tracking

Security & Cost Considerations

API Security

- Secure OpenAI API key management with environment variables
- Implement rate limiting to prevent API abuse and cost overruns
- Validate and sanitize all user inputs before processing

Cost Management

- Monitor API usage and implement daily/monthly limits
- Cache frequently requested images to reduce redundant generations
- Implement prompt optimization to improve generation efficiency

Quick Start Resources

- OpenAI DALL·E 3: https://platform.openai.com/docs/guides/images
- Celery Documentation: https://docs.celeryq.dev/
- Cloudinary React: https://cloudinary.com/documentation/react_integration
- WebSocket FastAPI: https://fastapi.tiangolo.com/advanced/websockets/

Sample User Interactions

Basic Image Generation

None

User: "Create an image of a sunset over mountains with a lake reflection"
Bot: "Generating your image... This may take up to 30 seconds"
[Progress bar shows 0% → 100%]
Bot: [Displays generated image] "Here's your sunset mountain landscape! Would you like me to create any variations?"

Style Modifications

None

User: "Make it more artistic, like a watercolor painting"
Bot: "Creating a watercolor version of your landscape..."
[New image generated with artistic style]
Bot: [Shows before/after comparison] "Here's the watercolor version! You can download both images."

Success Criteria Checklist

- Text prompts successfully generate relevant, high-quality images
- Real-time progress tracking keeps users informed during generation
- Images are automatically stored and accessible in gallery view
- Download functionality works for multiple formats and resolutions
- System handles errors gracefully with meaningful user feedback
- Integration with existing chat system feels natural and seamless
- Performance meets specified targets for generation and loading times
- Cost monitoring and rate limiting prevent API overuse
- Mobile-responsive design works across different devices
- Gallery search and filtering help users manage their creations