Synthex App - Project Task List

PROJECT DEADLINE: JUNE 10, 2025

Completed Tasks (Days 1-3, May 9-11)

- [] Create GitHub Repository
- [] Setup Development Environment
- [] Project Structure
- [] Streamlit App Skeleton
- [] FastAPI Backend Foundation
- [] Model Integration Research
- [] Basic UI Components (Streamlit)
- [] Documentation & Planning
- [] Code Explanation Feature (Frontend)
- [] Backend API for Explanations
- [] Code Generation Feature (Frontend)
- [] Backend API for Generation
- [] API Integration
- [] UI Polish Phase 1
- [] Learning Mode Feature (Frontend)
- [] Backend API for Learning
- [] Session History Implementation (Frontend)

FT1: Core Backend & API Enhancement (May 12-16)

- Implement all core API endpoints (explain, generate, learn, quiz, etc.)
- Add error handling and validation
- Integrate database for user/session/history (PostgreSQL/MongoDB)
- Add CORS middleware and robust error responses
- Implement API rate limiting and request queueing
- Complete backend unit tests (>90% coverage)

FT2: Enhanced Frontend & UI (May 17-21)

- Build comprehensive Streamlit panels and forms
- Implement advanced code explanation, generation, and learning UIs
- Add syntax highlighting and markdown rendering
- Add dark/light mode toggle
- Add loading animations and enhanced error handling
- Add file upload/drag-and-drop functionality

- Add multi-language code support selector
- Add settings/configuration UI with user preferences

FT3: Multi-Model Integration & Advanced Features (May 22-30)

Enhanced Model Integration

- Modular LLM provider architecture
- Add OpenAI GPT and Anthropic Claude model support
- Implement model selection UI and automatic fallback
- Add context window expansion and intelligent chunking

Code Generation & Prompt Features

- Advanced prompt-to-code generator with customizable parameters
- Inline code suggestions and completions
- Multi-step code generation with iterative improvements

• Visualization & Analysis Tools

- Code flow diagram generation
- Algorithm visualization with step-by-step animations
- Performance comparison charts and complexity analysis

• Version Control Integration

- GitHub/GitLab repository integration
- Commit message generation from code changes
- Code review assistant with automated suggestions
- Git diff explanation and analysis

FT4: Interactive Learning & Export Features (May 31-June 7)

Interactive Learning System

- Automated quiz generation from learning content
- Progress tracking for completed topics and skills
- Personalized learning paths based on user preferences
- Follow-up question system for deeper understanding

Export & Sharing Features

- PDF and Markdown export for explanations and code
- Shareable links for code explanations and generated content
- Integration with communication tools (Slack, Discord, Teams)
- Code snippet sharing with syntax highlighting

Advanced UI/UX Features

- Real-time collaboration on code explanations
- Advanced search and filtering in session history
- Keyboard shortcuts for common actions
- Responsive design optimization for mobile/tablet

FT5: Performance, Testing & Final Polish (June 8-10)

Scalability & Performance

- Implement horizontal scaling configurations
- Database optimization and connection pooling
- Caching layer for frequently requested explanations
- Performance monitoring and analytics

Testing & Quality Assurance

- Expand backend and frontend test coverage (>95%)
- Integration testing for all API endpoints
- Load testing for concurrent users
- Security testing and vulnerability assessment

• Documentation & Deployment

- Complete API documentation with interactive examples
- User guide with screenshots and video tutorials
- Deployment guide for production environments
- Final UI/UX polish and accessibility improvements

DEVELOPMENT APPROACH:

- Focus on creating a comprehensive web application with advanced features
- Each day should end with a working, committed version
- Prioritize features that enhance user experience and learning outcomes
- Start each day with testing tasks to ensure quality
- End each day with documentation updates
- Use feature flags for experimental or incomplete features

NEW CORE FEATURES SUMMARY:

Enhanced Code Intelligence

- Multi-model support (OpenAl GPT, Anthropic Claude, Groq Llama)
- Advanced prompt-to-code generation with customizable parameters
- Intelligent code suggestions and completions
- Context-aware explanations with expanded window support

📊 Visualization & Analysis

- Interactive code flow diagrams
- Step-by-step algorithm animations
- Performance comparison charts
- Code complexity analysis and optimization suggestions

Integration & Collaboration

- GitHub/GitLab repository integration
- Automated commit message generation
- Code review assistant with AI suggestions
- Git diff analysis and explanation

Advanced Learning System

- Interactive quiz generation from any code or topic
- Progress tracking and personalized learning paths
- Follow-up question system for deeper understanding
- Multi-language programming support

💾 Export & Sharing

- PDF/Markdown export functionality
- Shareable links for explanations and code
- Communication tool integrations (Slack, Discord, Teams)
- Real-time collaboration features

Performance & Scale

- Database integration for persistent storage
- API rate limiting and request queueing
- Horizontal scaling support
- Comprehensive caching layer

ADDITIONAL NOTES:

- Deadline: June 10th, 2025
- Focus on creating a comprehensive learning and development platform
- Prioritize user experience and educational value
- Ensure scalability for multiple concurrent users
- Implement robust error handling and fallback mechanisms
- Document all API changes and new endpoints thoroughly
- Consider technical dependencies when prioritizing implementation