Project Report

Karthik Venkat Malavathula

Executive Summary

This project successfully developed an innovative **Human-in-the-Loop Sandbox Generator** using **LangGraph** for workflow management and **AI models** for educational content generation. The system enables educators to create comprehensive educational sandboxes through collaborative human-AI interaction, generating interactive simulations, assessments, and theoretical content.

What Was Accomplished

1. Core System Development

- **LangGraph Workflow Engine**: Built a sophisticated 7-step workflow using LangGraph's StateGraph for managing sandbox generation
- **Multi-Model AI Integration**: Successfully integrated both Google Gemini and OpenAI models with dynamic switching capabilities
- **State Management**: Implemented robust state management for tracking progress and user feedback throughout the generation process

2. Dual Interface Implementation

- **Streamlit GUI**: Created a modern, responsive web interface with three-column layout (settings, content generation, chat)
- **CLI Interface**: Developed command-line interface for power users and automation
- **Real-time Progress Tracking**: Implemented live progress monitoring and state updates

3. Content Generation Pipeline

- **7-Step Workflow**: Sandbox Name → Aim → Pretest → Posttest → Theory → Procedure → References
- **Interactive Feedback Loop**: Users can review, provide feedback, and request updates at each step
- **Complete Package Generation**: Automatically creates all necessary files including interactive web simulations

4. File System Integration

- **Structured Output**: Generates organized directory structure with all educational materials
- **Interactive Simulations**: Creates HTML/JS/CSS web simulations for each sandbox
- **Multiple Download Options**: Individual files and complete package downloads

Key Features Implemented

AI-Powered Content Generation

- **Multi-Model Support**: Gemini 2.5 Flash, Gemini 1.5 Pro, GPT-4, GPT-3.5 Turbo
- **Dynamic Model Switching**: Users can change AI models during generation
- **Structured Prompts**: Comprehensive prompt templates for each content type
- **JSON Parsing**: Intelligent parsing of AI-generated quiz questions

Human-in-the-Loop Workflow

- **Step-by-Step Review**: Users review each generated component before proceeding
- **Feedback Integration**: AI models update content based on user feedback
- **Progress Tracking**: Real-time progress bar and completion status
- **Flexible Actions**: Update, save & continue, or skip options at each step

Educational Content Types

- **Learning Objectives** (aim.md): Detailed educational goals and outcomes
- **Assessment Questions** (pretest.json, posttest.json): Pre and post-sandbox evaluations
- **Theoretical Background** (theory.md): Scientific principles and concepts
- **Experimental Procedures** (procedure.md): Step-by-step instructions
- **Academic References** (reference.md): Scholarly sources and citations
- **Interactive Simulations** (simulation/): Web-based educational simulations

User Experience

- **Modern Web Interface**: Clean, responsive Streamlit-based GUI
- **Chat Integration**: Interactive AI assistance and guidance
- **File Management**: Comprehensive download and export options
- **Error Handling**: Robust error handling and user feedback

Technical Architecture

LangGraph Implementation

```
# Core workflow structure
workflow = StateGraph(SandboxState)
workflow.add_node("workflow_step", self.workflow_step)
workflow.add_conditional_edges("workflow_step", self.should_continue)
```

State Management

```
class SandboxState(TypedDict):
    sandbox_topic: str
    current_step: str
    progress: float
    user_feedback: str
    user_action: Literal["update", "save", "continue"]
    # ... additional state fields
```

AI Model Integration

- **Factory Pattern**: Dynamic model selection and initialization
- **Error Handling**: Graceful fallback and error recovery
- **Content Validation**: JSON parsing and content structure validation

Challenges Faced and Solutions

1. LangGraph Version Compatibility

Challenge: Frequent breaking changes in LangGraph API between versions **Solution**:

- Created dependency management scripts (`fix_dependencies.py`, `fix_langgraph.py`)
- Implemented version-specific compatibility checks
- Used flexible version ranges in requirements.txt

2. AI Model Integration Complexity

Challenge: Different API structures for Gemini vs OpenAI models **Solution**:

- Implemented unified interface with model-specific adapters
- Created abstraction layer for content generation
- Added model validation and error handling

3. State Management Complexity

Challenge: Managing complex state across multiple workflow steps **Solution**:

- Used TypedDict for type-safe state management
- Implemented immutable state updates
- Added comprehensive state validation

4. User Interface Responsiveness

Challenge: Real-time updates and state synchronization in Streamlit **Solution**:

- Implemented session state management
- Used conditional rendering for dynamic UI updates
- Added progress tracking and status indicators

5. File System Operations

Challenge: Generating complex directory structures and file formats **Solution**:

- Created modular file generation system
- Implemented template-based simulation generation
- Added comprehensive error handling for file operations

Technical Achievements

LangGraph Expertise

- Successfully implemented complex workflow management using LangGraph
- Created conditional edge logic for workflow branching
- Implemented state persistence and recovery mechanisms

AI Integration

- Seamless integration of multiple AI providers
- Intelligent content parsing and validation
- Robust error handling and fallback mechanisms

Web Development

- Modern, responsive Streamlit interface
- Real-time state management and updates
- Comprehensive file download and export functionality

Software Architecture

- Modular, maintainable code structure
- Comprehensive error handling and logging
- Scalable design for future enhancements

Impact and Applications

Educational Technology

- **Automated Content Creation**: Reduces time for educators to create educational materials
- **Standardized Quality**: Ensures consistent structure and quality across educational content
- **Interactive Learning**: Provides engaging web-based simulations for students

Research Applications

- **Rapid Prototyping**: Enables quick creation of educational experiments
- **Content Validation**: Human-in-the-loop ensures content accuracy and relevance
- **Scalable Production**: Can generate multiple sandboxes efficiently

Future Potential

- **Multi-Language Support**: Framework ready for internationalization
- **Advanced Simulations**: Architecture supports complex physics engines
- **Collaboration Features**: Foundation for multi-user collaboration

Lessons Learned

LangGraph Development

- **Version Management**: Critical to pin compatible versions and handle updates carefully
- **State Design**: Proper state structure is essential for complex workflows
- **Error Handling**: Comprehensive error handling is crucial for production systems

AI Integration

- **Model Abstraction**: Unified interfaces simplify multi-model support
- **Content Validation**: Always validate AI-generated content before use
- **User Feedback**: Human oversight improves content quality significantly

User Experience

- **Progressive Disclosure**: Step-by-step approach reduces cognitive load
- **Real-time Feedback**: Users appreciate immediate response and progress updates
- **Flexible Workflows**: Allow users to customize their experience

Conclusion

This project successfully demonstrates the potential of **LangGraph** for complex workflow management and **AI models** for educational content generation. The Human-in-the-Loop approach ensures high-quality, relevant content while maintaining human oversight and creativity.

The system provides a solid foundation for educational technology applications and showcases best practices for:

- **LangGraph workflow design**
- **Multi-model AI integration**
- **Interactive web application development**
- **Educational content generation**

The modular architecture and comprehensive documentation make this project a valuable reference for similar applications in educational technology and AI-assisted content creation.

Key Technologies: LangGraph, Streamlit, Google Gemini, OpenAI, Python **Development Time**: Comprehensive development with iterative improvements **Documentation**: Complete with setup guides, API reference, and troubleshooting