Solution Architecture

Project Design Phase

Date: 14 June 2025
Team ID: LTVIP2025TMID24661
Project Name: SmartTeach AI
Maximum Marks: 4 Marks

Solution Architecture Overview

Architecture Goals

- **Find the best tech solution** to solve personalized education problems
- **Describe the structure, characteristics, and behavior** of SmartTeach AI platform
- **Define features, development phases, and solution requirements** for scalable educational AI
- **Provide specifications** for platform definition, management, and delivery

---

1. High-Level Architecture Pattern

Architecture Style: Modular Monolithic Architecture with Service-Oriented Components

Core Architectural Principles

- **Modularity:** Loosely coupled components for independent development and scaling
- **Separation of Concerns:** Clear boundaries between presentation, business logic, and data layers
- **Scalability:** Horizontal scaling capabilities with cloud-native design
- **Security:** End-to-end encryption and secure authentication
- **Reliability:** Fault tolerance with graceful error handling and fallback mechanisms

2. System Architecture Layers

2.1 Presentation Layer (User Interface)

Streamlit Web Framework

- Responsive UI Components
- Interactive Navigation
- Custom CSS Styling
- Real-time Updates
- Cross-browser Compatibility

Components:

- **Authentication Interface:** Login/Registration forms with validation
- **Dashboard:** Progress tracking, analytics visualization, user overview
- **Chat Interface:** AI-powered conversational assistant
- **Quiz Interface:** Dynamic question generation and assessment
- **File Upload Interface:** Multi-format document processing
- **Resource Finder:** Educational content discovery and recommendation

2.2 Application Layer (Business Logic)

- Session Management
- Request Processing
- Workflow Orchestration
- Business Rules Engine
- Input Validation
- Response Formatting

Core Modules:

- **Authentication Manager:** User credential validation, session handling
- **Chat Controller:** Query processing, context management, response generation
- **Quiz Engine:** Question generation, difficulty adaptation, scoring logic
- **File Processor:** Document parsing, OCR processing, content extraction
- **Analytics Engine:** Performance calculation, progress tracking, trend analysis

2.3 Service Layer (AI & Processing)

- IBM WatsonX Integration
- Model Selection Algorithm
- Context Processing Engine
- File Processing Services
- Analytics Computation
- External API Handlers

Service Components:

- **AI Service:** IBM WatsonX API integration, model selection, response optimization
- **Document Service:** PDF processing (PyMuPDF), OCR (Tesseract), image handling (Pillow)
- **Quiz Service:** Dynamic question generation, difficulty adjustment, performance evaluation
- **Analytics Service:** Data processing, visualization, trend analysis

2.4 Data Layer (Storage & Management)

- JSON Data Storage
- User Credential Management
- Session Data Persistence
- Chat History Storage
- Quiz Performance Records
- File Metadata Management

Data Components:

- **User Data:** Encrypted credentials, profiles, preferences
- **Session Data:** Active sessions, authentication tokens, user state
- **Educational Data:** Chat logs, quiz results, progress metrics
- **Content Data:** Uploaded files, processed documents, extracted text
- **Analytics Data:** Performance metrics, usage statistics, trend data

2.5 Integration Layer (External Services)

- IBM WatsonX API Gateway
- Third-party Service Connectors
- Error Handling & Fallbacks
- Rate Limiting & Throttling
- API Authentication
- Response Caching

---

3. Detailed Component Architecture

3.1 AI Processing Pipeline

User Query → Context Analysis → Model Selection →

IBM WatsonX API → Response Processing →

Context Integration → User Response

Workflow Steps:

1. **Query Reception:** User input validation and preprocessing
2. **Context Analysis:** Document context integration, user history analysis
3. **Model Selection:** Automatic selection based on query complexity
4. **API Processing:** IBM WatsonX Granite model inference
5. **Response Enhancement:** Context integration, formatting, validation
6. **Delivery:** Real-time response with performance metrics

3.2 Quiz Generation Architecture

Topic Selection → Difficulty Analysis → Question Generation →

Context Integration → Dynamic Adjustment →

Real-time Scoring → Performance Analytics

Components:

- **Topic Parser:** Subject and complexity identification
- **Question Generator:** Dynamic creation based on educational content
- **Difficulty Engine:** Adaptive adjustment based on user performance
- **Scoring System:** Real-time evaluation and feedback
- **Analytics Processor:** Performance tracking and trend analysis

3.3 File Processing Pipeline

File Upload → Format Detection → Content Extraction →

OCR Processing → Text Analysis → Context Integration →

Searchable Knowledge Base

Processing Steps:

1. **Upload Handler:** Multi-format file reception and validation
2. **Format Detection:** Automatic file type identification
3. **Content Extraction:** PDF text extraction, image processing
4. **OCR Processing:** Handwritten and image text recognition
5. **Context Integration:** Knowledge base construction and indexing

---

4. Technology Stack Architecture

4.1 Frontend Architecture

- Framework: Streamlit 1.28+
- State Management: streamlit-cookies-manager
- Styling: Custom CSS, Responsive Design
- Components: Interactive widgets, Real-time updates
- Security: Client-side validation, HTTPS

4.2 Backend Architecture

- Runtime: Python 3.10+
- AI Platform: IBM WatsonX Foundation Models
- Data Format: JSON with schema validation

- Security: hashlib encryption, secure sessions
- File Processing: PyMuPDF, Pillow, pytesseract
- Utilities: BeautifulSoup4, OS operations

4.3 AI & ML Architecture

- Foundation Models: IBM Granite LLMs
- NLP Processing: Advanced language understanding
- Model Selection: Automatic optimization algorithm
- Context Processing: Document analysis & integration
- Performance: <2.3s average response time

5. Data Flow Architecture

5.1 User Authentication Flow

Registration/Login → Credential Validation →

Password Hashing → Session Creation →

Cookie Management → Dashboard Access

5.2 AI Chat Flow

User Query → Context Retrieval → Query Processing →

Model Selection → IBM WatsonX API → Response Processing →

Context Integration → User Interface Update

5.3 Quiz Generation Flow

Topic Selection → Difficulty Assessment →

Question Generation → User Interaction →

Real-time Scoring → Performance Storage →

Analytics Update → Progress Dashboard

5.4 File Processing Flow

File Upload → Format Detection → Content Extraction →

OCR Processing → Text Analysis → Context Database →

AI Integration → Enhanced Responses

6. Security Architecture

6.1 Authentication & Authorization

- **Multi-layer Security:** Client-side validation, server-side verification
- **Encrypted Storage:** Password hashing with salt, secure session tokens
- **Session Management:** Secure cookie handling, timeout mechanisms
- **Access Control:** Role-based permissions, secure API endpoints

6.2 Data Protection

- **Encryption in Transit:** HTTPS/TLS for all communications
- **Encryption at Rest:** Secure storage of user data and credentials
- **Input Validation:** Comprehensive sanitization and validation
- **Privacy Compliance:** Data minimization, user consent management

---

7. Performance & Scalability Architecture

7.1 Performance Optimization

- **Response Time:** <2.3 seconds average for AI queries
- **Memory Efficiency:** <500MB average memory consumption
- **Concurrent Users:** Support for 100+ simultaneous users
- **Caching:** Intelligent response caching for performance

7.2 Scalability Design

- **Horizontal Scaling:** Load balancing and distributed processing
- **Modular Components:** Independent scaling of services
- **API Gateway:** Rate limiting and request distribution
- **Database Optimization:** Efficient query processing and indexing

---

8. Deployment Architecture

8.1 Cloud Infrastructure

Platform: Streamlit Cloud

Hosting: Web-based application deployment

CDN: Global content delivery network

Monitoring: Performance and health monitoring

Backup: Automated data backup and recovery

8.2 CI/CD Pipeline

- **Version Control:** Git-based development workflow
- **Automated Testing:** Unit tests, integration tests, performance tests
- **Deployment Automation:** Continuous integration and deployment
- **Monitoring:** Real-time application health and performance monitoring

9. Future Architecture Considerations

9.1 Microservices Migration

- **Service Decomposition:** Breaking monolith into microservices
- **API Gateway:** Centralized routing and management
- **Container Orchestration:** Kubernetes deployment strategy
- **Service Mesh:** Inter-service communication and monitoring

9.2 Advanced AI Integration

- **Multi-Model Architecture:** Integration of specialized AI models
- **Edge Computing:** Local processing for improved performance
- **Real-time Learning:** Adaptive model training and optimization
- **Advanced Analytics:** Machine learning-powered insights

This architecture provides a robust, scalable foundation for SmartTeach AI while maintaining flexibility for future enhancements and growth.