MENTOR Tool Comprehensive B-Tests Suite

Executive Summary

This comprehensive testing suite is designed to evaluate the Multimodal AI Mentor across three phases (Ideation, Visualization, Materialization) with integrated cognitive benchmarking. The tests are engineered to maximize the potential of the cognitive benchmarking tool by incorporating the six key metrics: Cognitive Offloading Prevention (COP), Deep Thinking Engagement (DTE), Scaffolding Effectiveness (SE), Knowledge Integration (KI), Learning Progression (LP), and Metacognitive Awareness (MA).

Test Structure Overview

Core Testing Framework

- **Phase 1**: Ideation Phase Tests
- **Phase 2**: Visualization Phase Tests
- **Phase 3**: Materialization Phase Tests
- **Cross-Phase**: Integration & Progression Tests
- **Benchmarking**: Cognitive Metrics Validation

Participant Groups

- **Group A**: MENTOR Tool Users (Experimental)
- **Group B**: Generic AI Tool Users (ChatGPT/Claude - Control)
- **Group C**: No AI Assistance (Traditional Control)

Phase 1: Ideation Phase Tests

Test 1.1: Architectural Concept Development

Duration: 15 minutes
Scenario: Urban Community Center Design

Pre-Test Assessment

- Critical Thinking Assessment (Halpern CTDA - abbreviated 8 questions)
- Architectural Knowledge Baseline (12 questions)
- Spatial Reasoning Test (5 questions)

Main Task

Prompt: "You are tasked with designing a community center for a diverse urban neighborhood of 15,000 residents. The site is a former industrial warehouse (150m x 80m x 12m height). Consider: community needs, cultural sensitivity, sustainability, and adaptive reuse principles."

Phase-Specific Interactions

Group A (MENTOR) - Expected Socratic Dialogue Pattern:

1. **Initial Context Reasoning**: "Before we begin designing, what do you think are the most important questions we should ask about this community?"
2. **Knowledge Synthesis Trigger**: "What are some successful examples of warehouse-to-community transformations you're aware of?"
3. **Socratic Questioning**: "Why might the existing industrial character be valuable to preserve? What would be lost if we completely transformed it?"
4. **Metacognitive Prompt**: "How are you approaching this problem differently than a typical new-build community center?"

Group B (Generic AI): Standard prompt with ability to ask direct questions **Group C (No AI)**: Research materials and traditional resources only

Measured Outputs

- **Design Concept Quality** (Expert evaluation 1-10 scale)
- **Process Documentation** (thinking progression)
- **Cognitive Engagement Metrics**:
  - Question depth and frequency
  - Iterative refinement cycles
  - Assumption questioning behavior
  - Cultural consideration integration

Benchmarking Integration

- **COP Score**: Tracks ratio of exploratory vs. direct answer-seeking queries
- **DTE Score**: Measures reflection pauses, reasoning chains, complexity of responses
- **SE Score**: Evaluates appropriateness of guidance to user proficiency level
- **KI Score**: Assesses connection of architectural principles to design decisions

---

Test 1.2: Spatial Program Development

Duration: 10 minutes
Scenario: Functional Space Allocation

Task Description

"Based on your community center concept, develop a detailed spatial program. Consider: circulation patterns, adjacency requirements, flexibility needs, and community input integration."

Multi-Agent Interaction Triggers

- **Context Reasoning Agent**: "How do the functional relationships between spaces reflect community social patterns?"
- **Knowledge Synthesis Agent**: "What precedents can inform your adjacency decisions?"
- **Socratic Dialogue Agent**: "What assumptions are you making about how this community gathers and interacts?"
- **Metacognitive Agent**: "How is your programming methodology evolving as you think through this problem?"

Assessment Criteria

- **Spatial Logic Quality** (1-10)
- **Community Needs Integration** (1-10)
- **Flexibility & Adaptability** (1-10)
- **Justification Depth** (1-10)

Phase 2: Visualization Phase Tests

Test 2.1: 2D Design Development & Analysis

Duration: 20 minutes
Scenario: Schematic Design with Computer Vision Integration

Task Setup

Participants upload hand sketches or CAD drawings of their community center concept for AI analysis and critique.

Technical Integration

- **Computer Vision Processing**: Automated analysis of spatial proportions, circulation patterns, and design elements
- **Region Segmentation**: Identification of functional zones
- **Vision-Language Analysis**: Semantic understanding of design intent

MENTOR Tool Response Pattern

1. **Spatial Analysis**: "I notice your main gathering space represents 40% of the total area. How does this proportion relate to your intended community capacity?"
2. **Circulation Critique**: "Your circulation pattern creates a linear progression through spaces. What are the implications for spontaneous community interaction?"
3. **Proportion Questioning**: "The scale relationship between your entrance and main hall suggests a particular hierarchy. Was this intentional?"
4. **Design Principle Integration**: "How do your proportional decisions reflect principles of inclusive community design?"

Measured Outputs

- **Design Quality Improvement** (Pre/Post upload comparison)
- **Spatial Reasoning Development** (Measured through dialogue depth)
- **Visual Analysis Comprehension** (Response to AI feedback quality)
- **Design Iteration Cycles** (Number and sophistication of revisions)

Benchmarking Metrics

- **MA Score**: Self-reflection on design decisions after AI feedback
- **LP Score**: Skill progression from initial concept to refined design
- **KI Score**: Integration of feedback into design evolution

---

Test 2.2: Environmental & Contextual Integration

Duration: 10 minutes
Scenario: Site Responsiveness and Environmental Design

Task Description

"Integrate your community center design with environmental factors: natural lighting, ventilation, solar orientation, and urban context. Consider how the building responds to its surroundings."

Multi-Modal Interaction

- **Image Analysis**: Upload site photos for contextual analysis
- **Environmental Data Integration**: Solar studies, wind patterns, urban fabric analysis
- **Cultural Context Mapping**: Neighborhood character and community identity

MENTOR Dialogue Framework

- **Light & Orientation Agent**: "How might the industrial windows influence your natural lighting strategy throughout the day?"
- **Cultural Context Agent**: "What elements of the surrounding neighborhood architecture should your design respond to or contrast with?"
- **Sustainability Integration**: "How do your environmental strategies support community activities while honoring the building's industrial heritage?"

Assessment Dimensions

- **Environmental Responsiveness** (1-10)
- **Cultural Sensitivity** (1-10)
- **Technical Integration** (1-10)
- **Holistic Thinking** (1-10)

Phase 3: Materialization Phase Tests

Test 3.1: 3D Spatial Analysis & Material Systems

Duration: 20 minutes
Scenario: Detailed Design Development with 3D Analysis

Task Components

1. **3D Model Development**: Create detailed spatial model of community center
2. **Material Selection**: Choose appropriate materials for adaptive reuse
3. **Structural Integration**: Consider existing structural systems
4. **Construction Methodology**: Plan for community involvement in construction

3D Analysis Integration

- **Scene Graph Parsing**: Automated analysis of 3D geometry and spatial relationships
- **Spatial Analysis**: Volumetric studies, circulation flow analysis
- **Semantic Labeling**: Identification of functional zones and their relationships

- **Material Properties Integration**: Analysis of proposed material choices

MENTOR 3D Interface Interactions

- **Spatial Reasoning Challenges**: "Your double-height spaces create opportunities for visual connection. How do you envision this affecting community interaction patterns?"
- **Material Logic Questioning**: "You've chosen to expose the existing steel structure. How does this decision support both structural efficiency and community identity?"
- **Construction Reality Check**: "Given the community involvement you've proposed, how do your material choices support participatory construction methods?"

Complex Cognitive Challenges

- **Structural Engineering Integration**: "How do your design modifications work with the existing structural grid?"
- **Building Systems Coordination**: "Where will your new HVAC systems integrate with the preserved industrial elements?"
- **Accessibility & Universal Design**: "How does your vertical circulation strategy ensure inclusive access for all community members?"

Measured Outputs

- **3D Spatial Sophistication** (Expert evaluation)
- **Material System Logic** (Technical assessment)
- **Constructability & Feasibility** (Professional review)
- **Community Engagement Integration** (Social sustainability assessment)

Advanced Benchmarking

- **SE Score**: Adaptive scaffolding effectiveness across increasing complexity
- **KI Score**: Integration of technical, social, and cultural knowledge domains
- **DTE Score**: Deep thinking engagement with complex multi-variable problems

---

Test 3.2: Realization & Implementation Strategy

Duration: 15 minutes
Scenario: Project Implementation and Community Engagement

Task Description

"Develop a comprehensive implementation strategy for your community center, including: phased construction, community engagement process, funding strategies, and long-term stewardship plans."

Real-World Integration Challenges

- **Stakeholder Analysis**: Who are the key community stakeholders?
- **Phasing Strategy**: How can the building serve the community during construction?
- **Resource Allocation**: How do you balance community desires with budget constraints?
- **Long-term Adaptability**: How will the design evolve with changing community needs?

MENTOR Strategic Guidance

- **Implementation Reality**: "Your design proposals are ambitious. How would you prioritize elements if budget was reduced by 30%?"
- **Community Process**: "What methods will you use to ensure diverse community voices are heard in the design refinement process?"
- **Temporal Considerations**: "How might this community center need to adapt over the next 20 years?"

Cross-Phase Integration Tests

Test 4.1: Design Evolution Analysis

Duration: 10 minutes
Scenario: Reflective Analysis of Design Journey

Cognitive Progression Assessment

Participants review their complete design process from initial concept through final implementation strategy.

Reflection Prompts

1. "How did your understanding of the design problem evolve throughout the three phases?"
2. "What were the most significant learning moments in your design process?"
3. "How did the AI mentor influence your thinking patterns?"
4. "What would you approach differently in future projects?"

Benchmarking Integration

- **MA Score**: Metacognitive awareness and self-reflection quality

- **LP Score**: Learning progression across the complete design process
- **Overall Cognitive Development**: Integrated assessment across all six metrics

---

Test 4.2: Knowledge Transfer Challenge

Duration: 15 minutes
Scenario: Application to New Design Problem

Transfer Task

"Apply the principles and methodologies you've developed to a new scenario: Adaptive reuse of a former shopping mall into a mixed-use community hub."

Assessment Focus

- **Principle Transfer**: Application of learned design methodologies
- **Cognitive Strategy Transfer**: Use of questioning and analytical approaches
- **Knowledge Integration**: Synthesis of architectural, social, and technical knowledge
- **Independent Problem-Solving**: Ability to work without AI assistance

---

Specialized Cognitive Assessment Instruments

Pre/Post Critical Thinking Assessment

Modified Halpern Critical Thinking Assessment for Design

Sample Questions

1. **Verbal Reasoning**: "A community center design includes a large open space that can be divided. This flexibility is important because: [multiple choice with reasoning required]"
2. **Argument Analysis**: "Evaluate this design rationale: 'The entrance should be monumental because community centers need to make a strong civic statement.' Identify assumptions and evaluate the logic."
3. **Hypothesis Testing**: "If your community center design aims to promote intergenerational interaction, what specific design features would test this hypothesis and how would you measure success?"
4. **Likelihood & Uncertainty**: "Given limited community input data, how would you approach making design decisions about program allocation? Discuss your reasoning process."
5. **Problem Solving**: "A community group wants both a quiet library space and an active children's area in limited square footage. Describe your approach to resolving this apparent conflict."

Spatial Reasoning Assessment

3D Mental Rotation and Spatial Relationship Tasks

Sample Tasks

1. **Mental Rotation**: Identify matching 3D architectural forms from different viewpoints
2. **Spatial Relationships**: Analyze circulation patterns and spatial adjacencies
3. **Scale Perception**: Evaluate proportional relationships in architectural spaces
4. **Transformation Visualization**: Predict how spaces change with different configurations

Architectural Knowledge Assessment

Domain-Specific Knowledge Evaluation

Content Areas

1. **Building Types & Precedents** (25%)
2. **Structural Systems & Technology** (25%)
3. **Environmental Design & Sustainability** (25%)
4. **Social Architecture & Community Design** (25%)

Sample Questions

1. "Compare the spatial organization strategies of three notable community centers. How do their designs reflect different approaches to community interaction?"
2. "Explain how adaptive reuse strategies differ from new construction in terms of structural, environmental, and social considerations."
3. "Describe the relationship between building orientation, window placement, and natural lighting in community spaces."

Data Collection & Analysis Framework

Real-Time Data Capture

Interaction Logging

```
python
```

```
interaction_data = {
    'session_id': unique_session_id,
    'timestamp': iso_timestamp,
    'phase': current_phase,
    'user_input': sanitized_user_text,
    'ai_response': ai_response_text,
    'interaction_type': ['question', 'answer', 'reflection', 'critique'],
    'cognitive_load_indicators': pause_duration,
    'engagement_metrics': typing_speed_analysis,
    'navigation_patterns': interface_interaction_log
}
```

Cognitive Metrics Calculation

python

```
def calculate_realtime_metrics(session_data):
    cop_score = calculate_cop(session_data)
    dte_score = calculate_dte(session_data)
    se_score = calculate_se(session_data, user_profile)
    ki_score = calculate_ki(session_data)
    lp_score = calculate_lp(session_data)
    ma_score = calculate_ma(session_data)

    return {
        'cop': cop_score,
        'dte': dte_score,
        'se': se_score,
        'ki': ki_score,
        'lp': lp_score,
        'ma': ma_score,
        'composite': weighted_average(all_scores)
    }
```

Expert Evaluation Protocols

Design Quality Assessment Rubric

Creativity & Innovation (25%)

- Originality of approach (1-4)
- Creative problem-solving (1-4)

- Innovative use of existing structures (1-4)

Technical Competence (25%)

- Structural understanding (1-4)
- Environmental integration (1-4)
- Material selection appropriateness (1-4)

Community Responsiveness (25%)

- Cultural sensitivity (1-4)
- Accessibility & inclusion (1-4)
- Community engagement integration (1-4)

Design Process Quality (25%)

- Iterative development evidence (1-4)
- Justification depth (1-4)
- Self-reflection quality (1-4)

Statistical Analysis Plan

Primary Outcomes

- **Between-group comparisons**: ANOVA across three groups
- **Pre/post improvements**: Paired t-tests within groups
- **Cognitive metric correlations**: Pearson correlations between benchmarking scores and design quality

Secondary Analyses

- **Learning trajectory analysis**: Growth curve modeling over time
- **Interaction pattern analysis**: Sequence analysis of dialogue patterns
- **Transfer effectiveness**: Comparison of performance on novel tasks

---

Implementation Instructions for Claude Code

Development Environment Setup

```bash
```

```
# Project structure
mentor-btests/
├── src/
│   ├── data_collection/
│   │   ├── interaction_logger.py
│   │   ├── metrics_calculator.py
│   │   └── real_time_analyzer.py
│   ├── assessment_tools/
│   │   ├── critical_thinking_test.py
│   │   ├── spatial_reasoning_test.py
│   │   └── architecture_knowledge_test.py
│   ├── benchmarking/
│   │   ├── cognitive_metrics.py
│   │   ├── benchmark_engine.py
│   │   └── analysis_pipeline.py
│   ├── ui_components/
│   │   ├── test_interface.py
│   │   ├── upload_handler.py
│   │   └── progress_tracker.py
│   └── utils/
│       ├── data_processing.py
│       ├── export_handlers.py
│       └── validation.py
├── tests/
├── data/
│   ├── sessions/
│   ├── assessments/
│   └── exports/
├── docs/
└── requirements.txt
```

Core Dependencies

requirements

```
# AI & ML
anthropic==0.25.0
openai==1.14.0
torch==2.2.0
transformers==4.38.0
sentence-transformers==2.5.1

# Data Processing
pandas==2.2.0
numpy==1.26.4
scipy==1.12.0
scikit-learn==1.4.0

# Computer Vision
opencv-python==4.9.0
Pillow==10.2.0

# Web Framework
fastapi==0.109.0
uvicorn==0.27.0
streamlit==1.31.0

# Database
sqlalchemy==2.0.25
redis==5.0.1

# Analysis & Visualization
matplotlib==3.8.3
seaborn==0.13.2
plotly==5.18.0

# Testing
pytest==8.0.0
pytest-asyncio==0.23.5
```

Database Schema

```sql
```

```sql
-- Sessions table
CREATE TABLE sessions (
    id UUID PRIMARY KEY,
    participant_id VARCHAR(50),
    group_assignment VARCHAR(20),
    start_time TIMESTAMP,
    end_time TIMESTAMP,
    phase VARCHAR(20),
    completed BOOLEAN DEFAULT FALSE
);

-- Interactions table
CREATE TABLE interactions (
    id UUID PRIMARY KEY,
    session_id UUID REFERENCES sessions(id),
    timestamp TIMESTAMP,
    interaction_type VARCHAR(50),
    user_input TEXT,
    ai_response TEXT,
    cognitive_metrics JSONB,
    processing_time INTEGER
);

-- Assessments table
CREATE TABLE assessments (
    id UUID PRIMARY KEY,
    session_id UUID REFERENCES sessions(id),
    assessment_type VARCHAR(50),
    questions JSONB,
    responses JSONB,
    scores JSONB,
    completion_time INTEGER
);

-- Design outputs table
CREATE TABLE design_outputs (
    id UUID PRIMARY KEY,
    session_id UUID REFERENCES sessions(id),
    phase VARCHAR(20),
    output_type VARCHAR(50), -- 'text', 'image', '3d_model'
    content TEXT,
    file_path VARCHAR(255),
    expert_scores JSONB,
```

```
    created_at TIMESTAMP
);
```

Key Implementation Classes

1. Cognitive Metrics Calculator

python

```python
# src/benchmarking/cognitive_metrics.py
class CognitiveMetricsCalculator:
    def __init__(self):
        self.nlp_processor = SentenceTransformer('all-MiniLM-L6-v2')
        self.complexity_analyzer = TextComplexityAnalyzer()

    def calculate_cop_score(self, session_data):
        """Calculate Cognitive Offloading Prevention score"""
        direct_queries = self.count_direct_queries(session_data)
        exploratory_queries = self.count_exploratory_queries(session_data)
        inquiry_depth = self.analyze_inquiry_depth(session_data)

        if direct_queries + exploratory_queries == 0:
            return 0

        cop_ratio = exploratory_queries / (direct_queries + exploratory_queries)
        weighted_score = cop_ratio * inquiry_depth

        return self.normalize_score(weighted_score)

    def calculate_dte_score(self, session_data):
        """Calculate Deep Thinking Engagement score"""
        response_complexity = self.analyze_response_complexity(session_data)
        reasoning_chains = self.extract_reasoning_patterns(session_data)
        reflection_markers = self.count_reflection_language(session_data)
        pause_patterns = self.analyze_thinking_pauses(session_data)

        dte_score = (
            response_complexity * 0.3 +
            reasoning_chains * 0.3 +
            reflection_markers * 0.2 +
            pause_patterns * 0.2
        )

        return self.normalize_score(dte_score)
```

2. Test Interface Controller

python

```python
# src/ui_components/test_interface.py
class TestInterface:
    def __init__(self, db_connection, metrics_calculator):
        self.db = db_connection
        self.metrics = metrics_calculator
        self.current_session = None

    async def start_test_session(self, participant_id, group_assignment):
        """Initialize new test session"""
        session = Session(
            id=uuid4(),
            participant_id=participant_id,
            group_assignment=group_assignment,
            start_time=datetime.now()
        )

        await self.db.save_session(session)
        self.current_session = session

        return session.id

    async def process_user_interaction(self, user_input, phase):
        """Process user interaction and calculate real-time metrics"""
        interaction = Interaction(
            session_id=self.current_session.id,
            timestamp=datetime.now(),
            user_input=user_input,
            phase=phase
        )

        # Generate AI response based on group assignment
        ai_response = await self.generate_ai_response(
            user_input,
            self.current_session.group_assignment,
            phase
        )

        interaction.ai_response = ai_response

        # Calculate real-time cognitive metrics
        metrics = self.metrics.calculate_realtime_metrics(interaction)
        interaction.cognitive_metrics = metrics
```

```
        await self.db.save_interaction(interaction)

        return ai_response, metrics
```

3. Assessment Engine

python

```python
# src/assessment_tools/critical_thinking_test.py
class CriticalThinkingAssessment:
    def __init__(self):
        self.questions = self.load_questions()

    def generate_adaptive_test(self, participant_level='intermediate'):
        """Generate adaptive critical thinking test"""
        base_questions = self.questions[participant_level]
        randomized_questions = random.sample(base_questions, 15)

        return {
            'test_id': uuid4(),
            'questions': randomized_questions,
            'time_limit': 1800,  # 30 minutes
            'adaptive': True
        }

    def score_responses(self, responses):
        """Score critical thinking responses"""
        total_score = 0
        detailed_scores = {}

        for question_id, response in responses.items():
            question = self.get_question(question_id)
            score = self.evaluate_response(question, response)
            detailed_scores[question_id] = score
            total_score += score

        return {
            'total_score': total_score,
            'percentage': (total_score / len(responses)) * 100,
            'detailed_scores': detailed_scores,
            'skill_areas': self.analyze_skill_areas(detailed_scores)
        }
```

Data Export & Analysis Scripts

1. Benchmarking Results Export

python

```python
# src/utils/export_handlers.py
class BenchmarkingExporter:
    def __init__(self, db_connection):
        self.db = db_connection

    async def export_session_data(self, session_id, format='json'):
        """Export complete session data for analysis"""
        session = await self.db.get_session(session_id)
        interactions = await self.db.get_interactions(session_id)
        assessments = await self.db.get_assessments(session_id)
        design_outputs = await self.db.get_design_outputs(session_id)

        export_data = {
            'session': session.to_dict(),
            'interactions': [i.to_dict() for i in interactions],
            'assessments': [a.to_dict() for a in assessments],
            'design_outputs': [d.to_dict() for d in design_outputs],
            'cognitive_progression': self.calculate_progression(interactions),
            'benchmark_summary': self.generate_benchmark_summary(interactions)
        }

        if format == 'json':
            return json.dumps(export_data, indent=2)
        elif format == 'csv':
            return self.convert_to_csv(export_data)
        elif format == 'xlsx':
            return self.convert_to_excel(export_data)
```

2. Statistical Analysis Pipeline

python

```python
# src/benchmarking/analysis_pipeline.py
class StatisticalAnalysisPipeline:
    def __init__(self):
        self.stats_engine = StatisticsEngine()

    def run_between_groups_analysis(self, group_a_data, group_b_data, group_c_data):
        """Run ANOVA and post-hoc tests between groups"""
        results = {}

        # Design quality scores
        design_scores = [
            group_a_data['design_quality'],
            group_b_data['design_quality'],
            group_c_data['design_quality']
        ]

        f_stat, p_value = stats.f_oneway(*design_scores)
        results['design_quality_anova'] = {'f_stat': f_stat, 'p_value': p_value}

        # Cognitive metrics
        for metric in ['cop', 'dte', 'se', 'ki', 'lp', 'ma']:
            metric_scores = [
                group_a_data[metric],
                group_b_data[metric],
                group_c_data[metric]
            ]
            f_stat, p_value = stats.f_oneway(*metric_scores)
            results[f'{metric}_anova'] = {'f_stat': f_stat, 'p_value': p_value}

        return results

    def analyze_learning_progression(self, longitudinal_data):
        """Analyze learning progression over time"""
        progression_results = {}

        for participant in longitudinal_data:
            participant_progression = []
            for session in participant['sessions']:
                composite_score = session['cognitive_metrics']['composite']
                participant_progression.append(composite_score)

            # Calculate learning velocity
            velocity = self.calculate_learning_velocity(participant_progression)
```

```
            progression_results[participant['id']] = {
                'progression': participant_progression,
                'velocity': velocity,
                'trend': self.analyze_trend(participant_progression)
            }

        return progression_results
```

Deployment Instructions

1. Environment Setup

```bash
# Create virtual environment
python -m venv mentor-btests-env
source mentor-btests-env/bin/activate  # On Windows: mentor-btests-env\Scripts\activate

# Install dependencies
pip install -r requirements.txt

# Set up environment variables
cp .env.example .env
# Edit .env with your API keys and database connections
```

2. Database Setup

```bash
# Run database migrations
python -m alembic upgrade head

# Initialize test data
python scripts/initialize_test_data.py

# Verify setup
python scripts/verify_installation.py
```

3. Application Launch

```bash
```

```
# Start the API server
uvicorn src.main:app --reload --port 8000

# Start the Streamlit interface (in another terminal)
streamlit run src/ui_components/streamlit_app.py --server.port 8501

# Run the test suite
pytest tests/ -v
```

4. Configuration Files

config.yaml

yaml

```yaml
# Application configuration
app:
  name: "MENTOR B-Tests"
  version: "1.0.0"
  debug: false

# Database configuration
database:
  url: "postgresql://user:password@localhost:5432/mentor_btests"
  pool_size: 10
  echo: false

# AI Models configuration
ai_models:
  anthropic:
    api_key: "${ANTHROPIC_API_KEY}"
    model: "claude-3-sonnet-20240229"

  openai:
    api_key: "${OPENAI_API_KEY}"
    model: "gpt-4-turbo-preview"

# Benchmarking configuration
benchmarking:
  metrics:
    cop_weight: 0.20
    dte_weight: 0.20
    se_weight: 0.15
    ki_weight: 0.15
    lp_weight: 0.15
    ma_weight: 0.15

  thresholds:
    excellent: 0.85
    good: 0.70
    adequate: 0.55
    needs_improvement: 0.40

# Testing configuration
testing:
  session_timeout: 7200  # 2 hours
  auto_save_interval: 30  # 30 seconds
```

```
max_file_size: 10485760  # 10MB
allowed_file_types: ['.jpg', '.png', '.pdf', '.dwg', '.3dm']
```

Validation & Quality Assurance

1. Test Validation Protocol

- **Content Validity**: Expert review by 3 architectural educators and 2 cognitive scientists
- **Construct Validity**: Factor analysis of cognitive metrics
- **Reliability Testing**: Test-retest reliability with subset of participants
- **Inter-rater Reliability**: Multiple expert evaluations of design outputs

2. Data Quality Checks

- **Completeness Validation**: Ensure all required data points are collected
- **Consistency Checks**: Validate logical consistency across test phases
- **Outlier Detection**: Identify and investigate unusual response patterns
- **Missing Data Handling**: Implement appropriate imputation strategies

3. Ethical Considerations

- **Informed Consent**: Comprehensive consent process for data collection
- **Privacy Protection**: Data anonymization and secure storage
- **Participant Wellbeing**: Monitoring for fatigue or stress indicators
- **Right to Withdraw**: Clear procedures for participant withdrawal

---

Expected Outcomes & Success Metrics

Primary Hypotheses

1. **MENTOR group will show higher COP scores** (reduced cognitive offloading)
2. **Enhanced DTE scores** in MENTOR group (deeper thinking engagement)
3. **Improved design quality** with maintained cognitive engagement
4. **Better knowledge transfer** to novel design problems

Success Criteria

- **Effect size > 0.8** for cognitive metrics improvement
- **Statistically significant differences** ($p < 0.05$) between MENTOR and control groups

- **Qualitative evidence** of enhanced thinking processes
- **Expert validation** of improved design outcomes

Long-term Impact Assessment

- **6-month follow-up** to assess retention of cognitive strategies
- **Transfer study** with different design problems
- **Peer evaluation** of design thinking approaches
- **Self-reported behavioral changes** in design practice

This comprehensive testing suite provides the foundation for rigorous evaluation of the MENTOR tool while maximizing the potential of your cognitive benchmarking system. The integration of real-time metrics, expert evaluation, and longitudinal analysis ensures robust data collection for both immediate assessment and future research applications.