Project Kage - Complete Development Plan & Action Steps

Date: June 27, 2025 Project: AI-Powered Habit & Task Management System Goal: Build Flask web version for personal use, then expand to mobile

Immediate Next Steps (Planning Phase Completion)

Step 1: Technical Environment Setup Planning

Before any coding begins, we need to establish:

Database Decision Matrix

Factor	SQLite	PostgreSQL	Decision Rationale
Development Speed	✅ Fast	⚠️ Moderate	SQLite for rapid prototyping
Complex Relationships	⚠️ Limited	✅ Excellent	PostgreSQL for production
JSON Support	❌ Basic	✅ Native	Need for AI conversation storage
Migration Path	⚠️ Complex	✅ Smooth	PostgreSQL for scalability

Recommendation: Start PostgreSQL from Day 1

- Reason: Complex data relationships, JSON storage needs, easier production migration
- Setup: Use PostgreSQL with Docker for development consistency

Development Environment Specification

bash

```
# Required Software Stack
- Python 3.11+
- PostgreSQL 15+ (via Docker)
- Node.js 18+ (for frontend build tools if needed)
- Git (version control)
- VS Code or PyCharm (IDE with Python support)

# Python Dependencies (requirements.txt preview)
Flask==3.0.0
SQLAlchemy==2.0.23
Flask-SQLAlchemy==3.1.1
Flask-Migrate==4.0.5
psycopg2-binary==2.9.9
marshmallow==3.20.1
python-dotenv==1.0.0
APScheduler==3.10.4
```

Step 2: Project Structure Implementation Plan

File Creation Priority Order:

Phase 1A: Project Foundation (Day 1-2)

1. **Create project directory structure** (following our ultra-modular plan)
2. **Initialize Git repository** with proper .gitignore
3. **Set up virtual environment** and dependencies
4. **Configure PostgreSQL** connection and test
5. **Create basic Flask app** with Blueprint registration
6. **Set up database migrations** with Alembic

Phase 1B: Core Models (Day 3-4)

1. **User model** (future-proofing for multi-user)
2. **Task models** (QuickTask, DailyTask, StructuredTask)
3. **Habit models** (Habit, HabitCompletion)
4. **Goal models** (Goal, Milestone)
5. **Journal models** (JournalEntry)
6. **Calendar models** (TimeBlock, Schedule)

Phase 1C: Basic Routes (Day 5-7)

1. **Dashboard blueprint** with basic routes
2. **Task blueprint** with CRUD operations
3. **Habit blueprint** with CRUD operations
4. **Goal blueprint** with CRUD operations
5. **Journal blueprint** with CRUD operations

Step 3: Feature Implementation Roadmap

Sprint 1: Quick Capture & Basic Task Management

Goal: Have a functional brain dump and basic task system

Tasks to Complete:

- [ ] Quick capture interface (simple text input)
- [ ] Quick task storage with timestamps
- [ ] Basic task list display
- [ ] Task completion marking

- [ ] Task promotion to daily tasks

Acceptance Criteria:

- Can quickly add tasks with one text input
- Can see list of all quick tasks
- Can mark quick tasks as complete
- Can promote quick tasks to daily tasks with date assignment

Sprint 2: Daily Task Management

Goal: Manage today's scheduled tasks

Tasks to Complete:

- [ ] Daily task assignment interface
- [ ] Today's task view
- [ ] Task completion with time tracking
- [ ] Task rescheduling functionality
- [ ] Basic task analytics (completion times)

Acceptance Criteria:

- Can assign tasks to specific dates
- Can see today's tasks in dedicated view
- Can complete tasks with timestamp recording
- Can reschedule incomplete tasks
- Can see basic completion time patterns

Sprint 3: Habit Foundation

Goal: Basic habit tracking without AI

Tasks to Complete:

- [ ] Habit creation with DOSE categories
- [ ] Daily habit completion interface
- [ ] Streak calculation and display
- [ ] Habit frequency settings (daily, weekly, custom)
- [ ] Basic habit analytics

Acceptance Criteria:

- Can create habits with frequency and DOSE category
- Can mark habits complete with quality rating

- Streak calculations are accurate
- Can see habit completion patterns
- Basic habit success rate displayed

Sprint 4: Goal Integration

Goal: Connect habits to larger goals

Tasks to Complete:

- [ ] Goal creation and management
- [ ] Habit-goal linking system
- [ ] Goal progress calculation
- [ ] Milestone tracking
- [ ] Goal completion workflows

Acceptance Criteria:

- Can create goals with deadlines and descriptions
- Can link multiple habits to goals
- Goal progress reflects habit completions
- Can set and track milestones
- Can mark goals as complete

Sprint 5: Journal System

Goal: Reflective writing with data connections

Tasks to Complete:

- [ ] Journal entry creation and editing
- [ ] Mood tracking integration
- [ ] Linking entries to habits/goals/tasks
- [ ] Basic search and filtering
- [ ] Entry timeline view

Acceptance Criteria:

- Can create rich text journal entries
- Can assign mood ratings to entries
- Can link entries to specific habits, goals, or tasks
- Can search entries by content and filters
- Can view entries in chronological timeline

Sprint 6: Calendar Integration

Goal: Time blocking and schedule optimization

Tasks to Complete:

- [ ] Calendar view interface
- [ ] Time block creation and editing
- [ ] Task scheduling into time blocks
- [ ] Actual vs planned time tracking
- [ ] Schedule conflict detection

Acceptance Criteria:

- Can view daily/weekly calendar
- Can create time blocks for tasks
- Can drag and drop tasks into time slots
- System tracks actual time spent vs planned
- Warns about scheduling conflicts

Sprint 7: Dashboard Integration

Goal: Unified view of all systems

Tasks to Complete:

- [ ] Today's overview (tasks, habits, goals)
- [ ] Quick action buttons
- [ ] Progress visualizations
- [ ] Recent activity feed
- [ ] Weekly summary view

Acceptance Criteria:

- Dashboard shows all today's items
- Can perform common actions without leaving dashboard
- Visual progress indicators for habits and goals
- Can see recent completions and activities
- Weekly overview shows patterns and progress

Step 4: AI Learning Curriculum

Week 1: AI Fundamentals

Daily Learning Goals:

- **Day 1-2**: Understanding LLMs and tokens
- **Day 3-4**: Local vs Cloud AI options
- **Day 5-7**: Hands-on with Ollama setup and basic models

Deliverable: Ollama running locally with test model

Week 2: Model Selection & Testing

Daily Learning Goals:

- **Day 1-3**: Test 5+ different models for conversation quality
- **Day 4-5**: Test models for data analysis capabilities
- **Day 6-7**: Performance benchmarking and selection

Deliverable: Selected model with performance documentation

Week 3: Integration Architecture

Daily Learning Goals:

- **Day 1-2**: LangChain fundamentals
- **Day 3-4**: Conversation state management
- **Day 5-7**: Design AI service architecture for Project Kage

Deliverable: AI integration architecture document

Week 4: DOSE Knowledge Base

Daily Learning Goals:

- **Day 1-2**: Compile habit formation research
- **Day 3-4**: Create DOSE principle explanations
- **Day 5-7**: Build conversation templates

Deliverable: Complete knowledge base for AI training

Step 5: Quality Assurance Planning

Code Quality Standards

- **File Size Limit**: Max 300 lines per file (excluding templates)
- **Function Size Limit**: Max 50 lines per function
- **Documentation**: Docstrings for all functions and classes
- **Testing**: Unit tests for all service functions

- **Code Review**: Self-review checklist before moving to next feature

Testing Strategy

- **Unit Tests**: For all business logic functions
- **Integration Tests**: For database operations
- **Feature Tests**: For complete user workflows
- **Performance Tests**: For timestamp-heavy operations
- **User Testing**: Personal daily use validation

Modular Validation Checklist

- [ ] Each feature can be disabled without breaking others
- [ ] JavaScript files are feature-specific
- [ ] CSS files are component-specific
- [ ] Database models are in separate files
- [ ] Business logic is in service layer, not routes
- [ ] Templates use reusable components

Context Transfer Preparation for Future Conversations

Essential Artifacts for New Conversations

1. **This Development Plan** - Complete roadmap and context
2. **Methodology Document** - Your established development patterns
3. **Project Kage Documentation** - Complete feature specifications
4. **Current Progress State** - What's been completed
5. **Decision Log** - Technical decisions made and reasoning

Key Information for Context Transfer

- **Project Goal**: AI-powered habit tracking with Flask web version first
- **Methodology Source**: Girasoul Business Dashboard patterns
- **AI Approach**: Local models (Ollama) for budget/privacy
- **Database Choice**: PostgreSQL for complex relationships
- **Modularity Rule**: 30 small files better than 1 large file
- **Development Focus**: Step-by-step completion criteria, not time-based
- **User Requirements**: Task management, habit tracking, goal setting, journaling, calendar integration

Next Steps for New Conversations

1. **Reference All Artifacts**: Use this plan and methodology documents

2. **State Specific Goals**: Which sprint/feature to work on
3. **Declare Skill Levels**: Confirm understanding of technologies
4. **Plan for Context Limits**: Use artifact-based approach for complex development
5. **Follow Methodology**: Always analyze and plan before coding

Success Metrics for Planning Phase

- [ ] Complete project structure defined
- [ ] All technical decisions documented
- [ ] Development roadmap with clear completion criteria
- [ ] AI learning plan established
- [ ] Quality standards defined
- [ ] Context transfer strategy prepared

Immediate Action Items (Next 3 Conversations)

Conversation 1: Environment Setup

- Set up project structure
- Configure PostgreSQL and Flask
- Create basic models
- Initialize Git repository

Conversation 2: Sprint 1 Implementation

- Build quick capture functionality
- Implement basic task management
- Create first working interface
- Establish development workflow

Conversation 3: Sprint 2 & Testing

- Implement daily task management
- Add time tracking
- Set up testing framework
- Validate development approach

This plan provides the complete roadmap for Project Kage development while maintaining your modular architecture principles and AI learning requirements. Each step has clear completion criteria and builds toward the ultimate goal of an AI-enhanced personal productivity system.