Project Timeline & Milestones

AI-Powered Personal Productivity System

8.1 Development Phases

Phase 1: Foundation

Core Infrastructure Setup

Task	Owner	Dependencies	Priority
Project initialization & repository setup	DevOps	-	P0
Development environment configuration	Lead Dev	Project setup	P0
CI/CD pipeline setup	DevOps	Repository	P0
Vercel deployment configuration	DevOps	CI/CD	P0
Supabase project initialization	Backend	-	P0
Database schema implementation	Backend	Supabase	P0
Authentication system setup	Backend	Database	P0
Basic project structure	Frontend	Project setup	P0

Core Functionality Implementation

Task	Owner	Dependencies	Priority
Task CRUD operations API	Backend	Database	P0
Local storage implementation	Frontend	-	P0
Basic UI components	Frontend	Project structure	P0
Task list view	Frontend	UI components	P0
Create/Edit task forms	Frontend	UI components	P0
Basic state management	Frontend	-	P0
Unit tests for core functions	QA	All above	P0

Phase 1 Deliverables:

- ✓ Working authentication system
- ✓ Basic task CRUD operations
- ✓ Local storage functionality

- ✓ Deployment pipeline active

Phase 2: Core Features

Rich Content & Scheduling

Task	Owner	Dependencies	Priority
Tiptap editor integration	Frontend	-	P0
Rich text storage format	Backend	Database	P0
Task scheduling system	Backend	Task CRUD	P0
Recurring task patterns	Backend	Scheduling	P0
Calendar view component	Frontend	Task list	P1
Drag-drop functionality	Frontend	UI components	P1

PWA & Sync Foundation

Task	Owner	Dependencies	Priority
Service worker setup	Frontend	-	P0
PWA manifest configuration	Frontend	Service worker	P0
Offline mode implementation	Frontend	Service worker	P0
Basic sync engine	Backend	Database	P0
Conflict detection logic	Backend	Sync engine	P0
Real-time subscriptions	Backend	Supabase	P1
Integration tests	QA	All above	P0

Phase 2 Deliverables:

- ✓ Rich text editor functional
- ✓ Recurring tasks supported
- ✓ Basic offline mode working
- ✓ Initial sync capabilities

Phase 3: AI Integration

Local LLM Setup

Task	Owner	Dependencies	Priority
Ollama integration setup	AI/ML	-	P0
Model selection & download	AI/ML	Ollama	P0
Web Worker configuration	Frontend	-	P0
Task parsing implementation	AI/ML	Model setup	P0
Priority scoring algorithm	AI/ML	Task parsing	P0
Natural language input UI	Frontend	Task parsing	P0
LLM performance optimization	AI/ML	All above	P1

Claude Connector & AI Features

Task	Owner	Dependencies	Priority
Claude connector architecture	Lead Dev	-	P0
Export/Import mechanisms	Frontend	-	P0
Command parser development	AI/ML	-	P0
AI suggestions system	AI/ML	LLM setup	P1
Schedule optimization	AI/ML	Priority scoring	P1
AI feature UI integration	Frontend	All AI features	P0
AI integration tests	QA	All above	P0

Phase 3 Deliverables:

- ✓ Local LLM operational
- ✓ Task parsing working
- ✓ Claude connector functional
- ✓ AI-powered scheduling

Phase 4: Advanced Features

Real-time Sync & Conflict Resolution

Task	Owner	Dependencies	Priority
CRDT implementation	Backend	Sync engine	P0
Vector clock system	Backend	CRDT	P0
Conflict resolution UI	Frontend	CRDT	P0
Real-time sync optimization	Backend	Subscriptions	P1
Multi-device testing setup	QA	Sync system	P0
Sync performance optimization	Backend	All sync	P1

Automation & Enhancement

Task	Owner	Dependencies	Priority
Automation rules engine	Backend	Task system	P1
Rule builder UI	Frontend	Rules engine	P1
Advanced scheduling algorithms	AI/ML	Scheduling	P1
Bulk operations optimization	Backend	Task CRUD	P1
Performance optimizations	Full Stack	All features	P0
Security audit	Security	All features	P0

Phase 4 Deliverables:

- ✓ Full sync system operational
- ✓ Automation rules working
- ✓ Advanced scheduling active
- ✓ Performance optimized

Phase 5: Polish & Deploy

UI/UX Refinement

Task	Owner	Dependencies	Priority
UI consistency review	Frontend	-	P0
Responsive design fixes	Frontend	UI review	P0
Animation & transitions	Frontend	-	P2
Dark mode refinement	Frontend	Theme system	P1
Accessibility improvements	Frontend	All UI	P0
User onboarding flow	Frontend	All features	P0
Documentation writing	Tech Writer	All features	P0

Testing & Deployment

Task	Owner	Dependencies	Priority
Full E2E test suite	QA	All features	P0
Cross-platform testing	QA	E2E tests	P0
Performance testing	QA	All features	P0
Security testing	Security	All features	P0
Bug fixes	Full Stack	Testing	P0
Production deployment	DevOps	All tests pass	P0
Monitoring setup	DevOps	Deployment	P0

Phase 5 Deliverables:

- ✓ All tests passing
- ✓ Documentation complete
- ✓ Production deployment
- ✓ Monitoring active

8.2 Resource Allocation

Team Structure

Role	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
Lead Developer	Architecture, Code Review	Integration, Review	Claude Connector	Performance	Final Review
Frontend Developer	Basic UI	Rich Editor, PWA	AI UI Integration	Sync UI	Polish & Responsive

Role	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
Backend Developer	Database, API	Sync Engine	API Support	CRDT, Automation	Optimization
AI/ML Engineer	-	-	LLM Integration	Advanced AI	Fine-tuning
DevOps Engineer	Setup, CI/CD	Deployment	Infrastructure	Scaling	Production
QA Engineer	Test Planning	Integration Tests	AI Testing	Full Testing	Final Testing
Security Engineer	Initial Review	-	-	Security Audit	Final Audit

Task Dependencies Graph

```
Foundation
    ├── Database Schema
    │   ├── Task CRUD API
    │   │   ├── Rich Text Storage
    │   │   ├── Scheduling System
    │   │   └── Sync Engine
    │   │       ├── CRDT Implementation
    │   │       └── Real-time Sync
    │   └── Authentication
    │       └── User Management
    ├── Basic UI
    │   ├── Task List View
    │   ├── Rich Editor
    │   └── Calendar View
    └── Local Storage
        ├── Service Worker
        └── Offline Mode
            └── PWA Features

AI Integration
    ├── Local LLM
    │   ├── Task Parsing
    │   └── Priority Scoring
    └── Claude Connector
        ├── Export/Import
        └── Command Parser
```

8.3 Risk Management

Risk Matrix

Risk	Probability	Impact	Mitigation Strategy
LLM performance issues	Medium	High	Use smaller models, optimize caching
Sync conflicts	High	Medium	Implement robust CRDT system
Browser compatibility	Low	High	Progressive enhancement, polyfills
Storage limits	Medium	Medium	Implement archiving, compression
Security vulnerabilities	Low	Critical	Regular audits, pen testing
Scope creep	High	Medium	Strict phase planning, MVP focus
Third-party API changes	Low	Medium	Abstract integrations, fallbacks

Contingency Plans

yaml

```
performance_issues:
  detection: Automated performance monitoring
  response:
    - Immediate: Disable heavy features
    - Short-term: Optimize critical paths
    - Long-term: Refactor architecture

data_loss:
  detection: Integrity checks, user reports
  response:
    - Immediate: Stop writes, preserve state
    - Short-term: Restore from backups
    - Long-term: Improve backup strategy

security_breach:
  detection: Security monitoring, audits
  response:
    - Immediate: Isolate affected systems
    - Short-term: Patch vulnerability
    - Long-term: Security review

major_bug:
  detection: Error tracking, user reports
  response:
    - Immediate: Rollback if critical
    - Short-term: Hotfix deployment
    - Long-term: Improve testing
```

8.4 Success Metrics

Launch Criteria

Technical Requirements

- ✓ All P0 features implemented
- ✓ 90% test coverage achieved
- ✓ Performance budgets met
- ✓ Security audit passed
- ✓ Accessibility AA compliant
- ✓ Documentation complete

Quality Gates

Metric	Target	Status
Unit Test Coverage	>90%	Pending
Integration Test Coverage	>85%	Pending
E2E Test Coverage	>70%	Pending
Lighthouse Score	>90	Pending
Bundle Size	<200KB	Pending
Time to Interactive	<3s	Pending
Critical Bugs	0	Pending
High Priority Bugs	<5	Pending

Post-Launch KPIs

Initial Metrics

- User sign-ups tracking
- Active users monitoring
- Task creation volume
- Crash rate: Target <0.1%
- Error rate: Target <1%

Growth Metrics

- User retention: Target >80%
- Task completion rate: Target >70%
- Sync success rate: Target >99.9%
- AI parse accuracy: Target >95%
- User satisfaction: Target >4.5/5

Feature Adoption

- AI features usage rate
- Automation rules created
- Multi-device usage
- Claude connector adoption
- PWA installations

8.5 Documentation Deliverables

Document	Owner	Status	Priority
Technical Specification	Lead Dev	Complete	P0
API Documentation	Backend	In Progress	P0
User Guide	Tech Writer	Planned	P0
Admin Guide	DevOps	Planned	P1
Security Report	Security	Planned	P0
Testing Documentation	QA	In Progress	P0

8.6 Budget Considerations

Infrastructure Costs (All Free)

yaml

```yaml
infrastructure:
  hosting: $0 (Vercel free tier)
  database: $0 (Supabase free tier)
  cdn: $0 (Vercel CDN included)
  monitoring: $0 (Basic monitoring free)

tools:
  version_control: $0 (GitHub free)
  ci_cd: $0 (GitHub Actions free)
  project_management: $0 (GitHub Projects)

licenses:
  all_components: $0 (Open source)

total_monthly_cost: $0
```

Resource Requirements

yaml

```
development_team:
  lead_developer: 1
  frontend_developer: 1-2
  backend_developer: 1
  ai_ml_engineer: 1
  devops_engineer: 1 (part-time)
  qa_engineer: 1
  security_engineer: 1 (consulting)

minimum_viable_team: 3-4 developers
recommended_team: 5-6 developers
```

Appendices

A. Definition of Done

```yaml
code_complete:
  - Feature implemented
  - Unit tests written
  - Integration tests passed
  - Code reviewed
  - Documentation updated

ready_for_qa:
  - Deployed to staging
  - Test cases defined
  - Accessibility checked
  - Performance validated

production_ready:
  - QA approved
  - Security reviewed
  - Documentation complete
  - Monitoring configured
  - Rollback plan ready
```

B. Technology Decisions Log

Decision	Phase	Rationale	Alternatives Considered
React for frontend	Phase 1	Best ecosystem for Notion-like components	Vue, Svelte
Vercel for hosting	Phase 1	Best free tier, built for Next.js	Netlify, Railway
Supabase for database	Phase 1	Generous free tier, real-time built-in	Firebase, Appwrite
Mistral for LLM	Phase 3	Best performance/size ratio	Llama 2, Phi-2
Tiptap for editor	Phase 2	Most flexible, Notion-like	Slate, Draft.js

C. Phase Completion Checklist

```markdown
## Phase [X] Completion Checklist

### Required for Phase Completion
- [ ] All P0 tasks completed
- [ ] Tests passing (unit, integration)
- [ ] Documentation updated
- [ ] Code reviewed and approved
- [ ] Performance metrics met
- [ ] Security review passed

### Deliverables Verified
- [ ] Features working as specified
- [ ] No critical bugs
- [ ] Accessibility standards met
- [ ] Cross-platform compatibility confirmed

### Ready for Next Phase
- [ ] Dependencies resolved
- [ ] Team briefed on next phase
- [ ] Resources allocated
- [ ] Risks identified and mitigated
```

Document Version: 1.0.0

Last Updated: January 2024

Next Review: February 2024