

User Documentation Requirements

AI-Powered Personal Productivity System

6.1 User Interface Specifications

Design System

SCSS

// Core Design Tokens

\$colors: (

primary: #2563eb,
secondary: #7c3aed,
success: #10b981,
warning: #f59e0b,
error: #ef4444,

// Semantic colors

text-primary: #111827,
text-secondary: #6b7280,
text-disabled: #9ca3af,

background: #ffffff,
surface: #f9fafb,
border: #e5e7eb

);

\$typography: (

font-family: 'Inter, system-ui, sans-serif',

// Scale

xs: 0.75rem,
sm: 0.875rem,
base: 1rem,
lg: 1.125rem,
xl: 1.25rem,
2xl: 1.5rem,
3xl: 1.875rem,

// Weights

normal: 400,
medium: 500,
semibold: 600,
bold: 700

);

\$spacing: (

0: 0,
1: 0.25rem,
2: 0.5rem,
3: 0.75rem,

```
4: 1rem,  
6: 1.5rem,  
8: 2rem,  
12: 3rem,  
16: 4rem
```

```
);
```

```
$breakpoints: (
```

```
xs: 320px,  
sm: 640px,  
md: 768px,  
lg: 1024px,  
xl: 1280px,  
2xl: 1536px
```

```
);
```

Component Library

```
typescript
```

```
// Button variants
interface ButtonProps {
  variant: 'primary' | 'secondary' | 'outline' | 'ghost' | 'danger';
  size: 'xs' | 'sm' | 'md' | 'lg' | 'xl';
  disabled?: boolean;
  loading?: boolean;
  fullWidth?: boolean;
  leftIcon?: ReactNode;
  rightIcon?: ReactNode;
}
```

```
// Input components
interface InputProps {
  type: 'text' | 'email' | 'password' | 'number' | 'date' | 'time';
  size: 'sm' | 'md' | 'lg';
  error?: string;
  helperText?: string;
  prefix?: ReactNode;
  suffix?: ReactNode;
  clearable?: boolean;
}
```

```
// Card layouts
interface CardProps {
  variant: 'elevated' | 'outlined' | 'filled';
  padding: 'none' | 'sm' | 'md' | 'lg';
  interactive?: boolean;
  selected?: boolean;
}
```

Interaction Patterns

Gesture/Action	Desktop	Mobile	Result
Create Task	Cmd/Ctrl+N	Tap + button	Open task creator
Quick Add	Type anywhere	Long press	Inline task input
Complete Task	Click checkbox	Swipe right	Mark complete
Delete Task	Cmd/Ctrl+D	Swipe left	Move to trash
Edit Task	Double-click	Tap task	Open editor
Multi-select	Shift+click	Long press+tap	Select multiple
Drag reorder	Mouse drag	Touch hold+drag	Reorder tasks

Gesture/Action	Desktop	Mobile	Result
Search	Cmd/Ctrl+K	Pull down	Open search

Responsive Design

CSS

```
/* Mobile First Approach */
```

```
/* Base: Mobile (320px - 639px) */
```

```
.container {  
  padding: 1rem;  
  max-width: 100%;  
}
```

```
.task-grid {  
  display: flex;  
  flex-direction: column;  
  gap: 0.5rem;  
}
```

```
/* Tablet (640px - 1023px) */
```

```
@media (min-width: 640px) {  
  .container {  
    padding: 1.5rem;  
    max-width: 768px;  
    margin: 0 auto;  
  }  
}
```

```
.task-grid {  
  display: grid;  
  grid-template-columns: repeat(2, 1fr);  
  gap: 1rem;  
}  
}
```

```
/* Desktop (1024px+) */
```

```
@media (min-width: 1024px) {  
  .container {  
    padding: 2rem;  
    max-width: 1280px;  
  }  
}
```

```
.task-grid {  
  grid-template-columns: repeat(3, 1fr);  
  gap: 1.5rem;  
}
```

```
.sidebar {  
  display: block;
```

```
width: 280px;
```

```
}
```

```
}
```

Accessibility Requirements

```
html
```

<!-- Semantic HTML structure -->

<main role="main" aria-label="Task List">

<section aria-labelledby="today-tasks">

<h2 id="today-tasks">Today's Tasks</h2>

<ul role="list" aria-label="Task items">

<li role="listitem">

<article class="task-card" aria-label="Task: Review documentation">

<button

role="checkbox"

aria-checked="false"

aria-label="Mark task as complete"

tabindex="0"

>

Complete task

</button>

<div class="task-content">

<h3>Review documentation</h3>

<time datetime="2024-01-15T15:00">3:00 PM</time>

</div>

<button

aria-label="Task options"

aria-haspopup="menu"

aria-expanded="false"

>

More options

</button>

</article>

</section>

</main>

<!-- Skip links -->

Skip to main content

Skip to navigation

Skip to search

6.2 User Onboarding Flow

First-Time User Experience

typescript

```
interface OnboardingStep {
```

```
  id: string;
```

```
  title: string;
```

```
  description: string;
```

```
  action: () => void;
```

```
  skipable: boolean;
```

```
}
```

```
const onboardingSteps: OnboardingStep[] = [
```

```
{
```

```
  id: 'welcome',
```

```
  title: 'Welcome to Your Productivity Hub',
```

```
  description: 'Let\'s get you set up in under 2 minutes',
```

```
  action: () => {},
```

```
  skipable: false
```

```
},
```

```
{
```

```
  id: 'create-first-task',
```

```
  title: 'Create Your First Task',
```

```
  description: 'Try typing "Meeting tomorrow at 2pm"',
```

```
  action: () => openTaskCreator(),
```

```
  skipable: false
```

```
},
```

```
{
```

```
  id: 'enable-ai',
```

```
  title: 'Enable AI Assistant',
```

```
  description: 'Let AI help parse and schedule your tasks',
```

```
  action: () => enableAI(),
```

```
  skipable: true
```

```
},
```

```
{
```

```
  id: 'sync-setup',
```

```
  title: 'Enable Cross-Device Sync',
```

```
  description: 'Access your tasks from any device',
```

```
  action: () => setupSync(),
```

```
  skipable: true
```

```
},
```

```
{
```

```
  id: 'customize',
```

```
  title: 'Customize Your Workflow',
```

```
  description: 'Choose your preferred view and theme',
```

```
  action: () => openSettings(),
```

```
  skipable: true
```

```
}  
];
```

Interactive Tutorial

```
javascript  
  
// Tooltip-based tutorial system  
const tutorialSteps = [  
  {  
    target: '.create-task-button',  
    content: 'Click here to create a new task',  
    position: 'bottom'  
  },  
  {  
    target: '.task-input',  
    content: 'Type naturally - we\'ll parse dates and priorities',  
    position: 'top'  
  },  
  {  
    target: '.view-selector',  
    content: 'Switch between List, Kanban, and Calendar views',  
    position: 'left'  
  },  
  {  
    target: '.claude-connector',  
    content: 'Connect to Claude for advanced AI features',  
    position: 'right'  
  }  
];
```

6.3 Feature Documentation

Task Management Guide

```
markdown
```

Task Management

Creating Tasks

Natural Language Input

Type tasks naturally and our AI will parse them:

- "Call John tomorrow at 3pm" → Creates task with due date
- "High priority: Review proposal" → Sets high priority
- "Weekly team meeting #work" → Adds recurring pattern and tag

Rich Text Editor

Use Notion-like blocks for detailed descriptions:

- `/heading` - Add headings
- `/list` - Create lists
- `/toggle` - Add collapsible sections
- `/code` - Insert code blocks
- `/table` - Add tables

Keyboard Shortcuts

- `Cmd/Ctrl + N` - New task
- `Cmd/Ctrl + Enter` - Save task
- `Cmd/Ctrl + D` - Delete task
- `Tab` - Indent subtask
- `Shift + Tab` - Outdent subtask

Organizing Tasks

Tags and Categories

- Use `#tag` syntax for quick tagging
- Create custom categories in settings
- Filter by multiple tags simultaneously

Dependencies

- Link related tasks
- Auto-update dependent tasks
- Visualize task chains

Templates

- Save common workflows
- Apply templates with variables
- Share templates with team

AI Features Guide

markdown

AI Assistant Features

Local AI Processing

Your data never leaves your device:

- Task parsing happens locally
- Priority scoring runs offline
- Schedule optimization works without internet

Claude Integration

Connect with Claude for advanced features:

Setup

1. Open Claude Connector (Settings → AI)
2. Copy your task data
3. Paste in Claude chat
4. Copy Claude's commands back

Available Commands

- ``ANALYZE`` - Get insights on your productivity
- ``OPTIMIZE`` - Reorganize schedule for efficiency
- ``SUGGEST`` - Get task recommendations
- ``AUTOMATE`` - Create automation rules

Natural Language Queries

Ask questions like:

- "What should I work on today?"
- "Show me overdue tasks"
- "Schedule my day"
- "Find tasks about project X"

6.4 API Documentation

REST API Reference

yaml

Authentication

All API requests require authentication via Bearer token:

Authorization: Bearer <token>

Base URL

https://app.vercel.app/api

Rate Limiting

- 1000 requests per hour per user
- 100 requests per minute per IP
- Headers: X-RateLimit-Limit, X-RateLimit-Remaining

Endpoints

Tasks

GET /api/tasks

Retrieve tasks with filtering and pagination

Query Parameters:

- status (string): pending|completed|archived
- priority (number): 0-10
- tags (string): comma-separated tags
- search (string): search query
- limit (number): results per page (default: 50, max: 100)
- offset (number): pagination offset
- sort (string): field:direction (e.g., created_at:desc)

Response:

```
{
  "data": [Task],
  "meta": {
    "total": 150,
    "limit": 50,
    "offset": 0
  }
}
```

POST /api/tasks

Create a new task

Request Body:

```
{
```

```
"title": "string (required)",
"description": "string",
"priority": "number (0-10)",
"due_date": "ISO 8601 string",
"tags": ["string"],
"parent_id": "UUID"
}
```

Response: Task object

PUT /api/tasks/:id

Update an existing task

Request Body: Partial task object

Response: Updated task object

DELETE /api/tasks/:id

Delete a task

Response: { "success": true }

Sync

POST /api/sync/push

Push local changes to server

Request Body:

```
{
  "device_id": "string",
  "changes": [Delta],
  "vector_clock": {}
}
```

Response:

```
{
  "accepted": [UUID],
  "rejected": [UUID],
  "conflicts": [Conflict]
}
```

GET /api/sync/pull

Pull changes from server

Query Parameters:

- **device_id** (string): unique device identifier
- **since** (timestamp): changes after this time

Response:

```
{  
  "changes": [Delta],  
  "vector_clock": {},  
  "has_more": boolean  
}
```

WebSocket API

javascript

// Connection

```
const ws = new WebSocket('wss://app.vercel.app/ws');
```

// Authentication

```
ws.send(JSON.stringify({  
  type: 'auth',  
  token: 'Bearer <token>'  
}));
```

// Subscribe to real-time updates

```
ws.send(JSON.stringify({  
  type: 'subscribe',  
  channels: ['tasks', 'sync']  
}));
```

// Receive updates

```
ws.on('message', (data) => {  
  const event = JSON.parse(data);
```

```
  switch(event.type) {  
    case 'task:created':  
      // New task created  
      break;  
    case 'task:updated':  
      // Task updated  
      break;  
    case 'task:deleted':  
      // Task deleted  
      break;  
    case 'sync:required':  
      // Sync needed  
      break;  
  }  
});
```

// Send real-time updates

```
ws.send(JSON.stringify({  
  type: 'task:update',  
  data: {  
    id: 'task-uuid',  
    changes: { status: 'completed' }  
  }  
}));
```

```
}
```

```
}});
```

6.5 Troubleshooting Guide

Common Issues

Issue	Cause	Solution
Tasks not syncing	Network issues	Check connection, manual sync
AI not responding	Model not loaded	Restart app, check memory
Claude commands fail	Invalid format	Check command syntax
Lost data	Sync conflict	Check conflict resolver
Slow performance	Too many tasks	Archive old tasks
Can't login	Token expired	Re-authenticate

Error Messages

typescript

```
const errorMessages = {  
  'NETWORK_ERROR': 'Unable to connect. Working offline.',  
  'SYNC_CONFLICT': 'Sync conflict detected. Review changes.',  
  'AI_TIMEOUT': 'AI processing timed out. Try again.',  
  'STORAGE_FULL': 'Local storage full. Archive old tasks.',  
  'AUTH_EXPIRED': 'Session expired. Please login again.',  
  'INVALID_INPUT': 'Invalid input format. Check syntax.',  
  'RATE_LIMITED': 'Too many requests. Please wait.',  
  'ENCRYPTION_ERROR': 'Unable to decrypt data. Check password.'  
};
```

Debug Mode

javascript

```
// Enable debug mode in console
localStorage.setItem('debug', 'true');

// Debug information available
{
  performance: {
    renderTime: 'ms per frame',
    memoryUsage: 'MB used',
    apiLatency: 'ms per request'
  },
  sync: {
    lastSync: 'timestamp',
    pendingChanges: 'count',
    conflicts: 'array'
  },
  ai: {
    modelLoaded: 'boolean',
    inferenceTime: 'ms',
    cacheHits: 'percentage'
  }
}
```

Document Version: 1.0.0

Last Updated: January 2024

Next Review: February 2024