Product Requirements Document (PRD)

Product Requirements Document (PRD)

Product Name
Personal Daily Priority Tracker

Overview
The Personal Daily Priority Tracker is a mobile-first web app designed for individuals, particularly those with INTP-T personality traits and inattentive ADHD, to manage daily tasks with minimal cognitive load. It enables quick task creation, stack ranking, commenting, and structured review.

Goals
1. **Reduce Cognitive Load**: Minimize decision fatigue for daily planning.
2. **Centralized Task Management**: All daily priorities in one place.
3. **On-the-Go Access**: Optimized for mobile usage.
4. **Historical Tracking**: Maintain a searchable record of past tasks.

Core Features & CRUD Capabilities

1. Task Management (CRUD)
- **Create**: Add tasks with fields: Task Name, Why, ETA, Priority, Comments.
- **Read**: View current day's tasks (pending & completed).
- **Update**: Edit any field of an existing task.
- **Delete**: Remove tasks permanently or archive them.
- **Append Comments**: Add multiple comments without overwriting older ones (versioned comment log).

2. Priority Management
- Drag-and-drop or quick buttons to reorder tasks.
- Numeric ranking system (1 = highest priority).
- Auto-sort by priority on save.

3. Review & Reporting
- View pending vs completed tasks.
- Export task list (CSV/PDF).
- Task carry-over feature.

4. Search & History
- Search across historical tasks.
- Filters by date, priority, or keyword.

Cloud Storage & Data Persistence
- **Storage**: All task data stored in a secure cloud database (AWS DynamoDB or Firebase Firestore).
- **Media Storage**: Any attachments or files stored in AWS S3 or Firebase Storage.
- **Authentication**: User authentication using OAuth (Google, Slack, or Company SSO).
- **Offline-first**: Local caching via IndexedDB/PWA service workers, syncing when online.

Product Requirements Document (PRD)

Orchestration Layer
- **Purpose**: Manage all API requests, background syncing, and daily reset automation.
- **Option 1 (Serverless)**: AWS Lambda or Google Cloud Functions to handle CRUD API endpoints.
- **Option 2 (Containerized)**: Lightweight Node.js backend in AWS ECS/Fargate.
- **Scheduler**: CloudWatch (AWS) or Firebase Scheduled Functions for daily reset jobs.

Phased Implementation Plan

Phase 1 (High Priority - Core Functionality)
- Task CRUD operations
- Priority ranking system
- Comment append system
- Cloud-based persistent storage
- Mobile-first UI

Phase 2 (Medium Priority - Enhancements)
- Daily review dashboard
- Export options (CSV/PDF)
- Historical task search

Phase 3 (Low Priority - Automation & Personalization)
- AI-based task suggestions
- Recurring task automation
- Integration with Slack for quick task entry

Non-Functional Requirements
- **Performance**: Tasks should load in <200ms from cache.
- **Scalability**: Handle up to 1000 tasks/user/day.
- **Security**: Encrypted storage (AES-256), HTTPS for all requests.
- **Usability**: Minimalist design, large tap targets, accessible color palette.

Wireframes
(See the following pages for visual mockups.)

Product Requirements Document (PRD)

Mobile App Wireframes