

# Next-Gen To-Do Application

## Full Web Pages Specification & Development Order Guide

This document outlines **all required pages**, their **purposes**, **functionalities**, **development priority order**, and **important development instructions**. It is designed to act as a complete blueprint for frontend and backend implementation.

---



## 1. DEVELOPMENT ORDER (HIGH → LOW PRIORITY)

A logical, efficient order to build the entire application.

### PHASE 1 — Core Foundation (Must Build First)

1. Authentication Pages
2. Dashboard
3. All Tasks Page
4. Task Create/Edit Modal
5. Task Details Page

### PHASE 2 — Supporting Features

1. Tags Management
2. Reminders Overview
3. Recurring Schedules Page
4. Subtasks Page (optional)

### PHASE 3 — User Account & Preferences

1. Profile Page
2. Settings Page
3. Attachments Page

### PHASE 4 — Admin Panel

1. Admin Dashboard
2. Admin User Management
3. Admin Events Viewer
4. Admin Notifications Monitor

## PHASE 5 — System Pages

1. **404 Page**
  2. **Error Page (500)**
  3. **Privacy Policy**
  4. **Terms & Conditions**
- 



## 2. COMPLETE PAGE LIST WITH DETAILS

Below is the **full explanation** of all required web pages.

---



## AUTHENTICATION & ONBOARDING

### 1. Login Page

**Importance:** Critical

**Key Features:** - Email/password login - "Forgot password" link - Option for social login (Google)

**Modules Used:** `users`, Supabase Auth

---

### 2. Register / Sign-up Page

**Importance:** Critical

**Key Features:** - Create user account - Email verification (optional) - Redirects to onboarding

---

### 3. Forgot Password Page

**Importance:** Medium

**Key Features:** - Enter email → send reset link

---

### 4. Onboarding Page

**Importance:** Medium-High

**Key Features:** - Choose theme, timezone, productivity preference - Auto-create default tags (Work, Personal)

---



## MAIN APPLICATION PAGES

### 5. Dashboard

**Importance:** Highest

**Key Features:** - Today's tasks - Overdue tasks - Quick-add task - Productivity stats

**DB Usage:** `tasks` , `events`

---

### 6. All Tasks Page

**Importance:** Highest

**Key Features:** - List all tasks - Filtering (status, tags, priority, due date) - Search bar - Bulk actions

---

### 7. Task Details Page

**Importance:** Highest

**Key Features:** - Full task info - Edit/delete task - Manage reminders - Manage subtasks - Attachments viewer - View task activity log

---

### 8. Task Create/Edit Modal

**Importance:** Highest

**Key Features:** - Title, description - Tags - Due date/time - Priority - Recurring schedule - Add reminder - File attachments

This modal is used in Dashboard, All Tasks, Tags Page, etc.

---

### 9. Subtasks Page (Optional)

**Importance:** Medium

**Key Features:** - List all subtasks of a parent task - Mark complete - Add/edit/delete subtasks

---



## TAGS & CATEGORIES

### 10. Tags Management Page

**Importance:** Medium

**Key Features:** - Add/edit/delete tags - Assign colors - Show task counts per tag

**DB Tables:** `tags`, `task_tags`

---



## REMINDERS & NOTIFICATIONS

### 11. Reminders Overview Page

**Importance:** Medium

**Key Features:** - list all reminders across tasks - toggle reminders on/off

**DB Table:** `reminders`

---

### 12. Notification History Page

**Importance:** Low-Medium

**Key Features:** - View notifications (pending/sent/failed)

**DB Table:** `notifications`

---



## RECURRING TASKS

### 13. Recurring Schedules Page

**Importance:** Medium

**Key Features:** - create recurring rules (daily/weekly/monthly) - link/unlink tasks

**DB Table:** recurring\_schedules

---

## ACCOUNT SETTINGS

### 14. Profile Page

**Importance:** Medium

**Key Features:** - Edit display name - Change password - Update email - Delete account

---

### 15. Settings Page

**Importance:** Medium

**Key Features:** - Theme mode - Notification preferences - Language - Productivity settings

---

## FILE MANAGEMENT

### 16. Attachments Page

**Importance:** Low-Medium

**Key Features:** - List all uploaded files - Preview - Delete

**DB Table:** attachments

---

## ADMIN PANEL

### 17. Admin Dashboard

**Importance:** High (for admin users)

**Key Features:** - Total users - DAU (daily active users) - Tasks created today - Failed notifications

**DB Tables:** users, tasks, events, notifications

---

## 18. Admin User Management Page

**Importance:** Medium

**Key Features:** - List all users - Ban/delete user - Toggle admin flag

---

## 19. Admin Events Viewer

**Importance:** Low-Medium

**Key Features:** - View raw events from `events` table - Filter by type/date

---

## 20. Admin Notifications Monitor

**Importance:** Low-Medium

**Key Features:** - View status of notifications - Retry failed ones (manual)

---



## SYSTEM PAGES

### 21. 404 Page

Simple "Not Found" screen.

---

### 22. 500 Error Page

Show error debug info (in dev mode only).

---

### 23. Privacy Policy Page

Required for production.

---

### 24. Terms & Conditions Page

Required for production.

---



## 3. IMPORTANT DEVELOPMENT INSTRUCTIONS

Use this as your rulebook while building.

---

### A. Architecture & Code Quality

- Use component-driven architecture (React recommended).
  - Keep pages lightweight; move logic to hooks/services.
  - Always separate UI, Logic, and DB interactions.
- 

### B. UX Guidelines

- Every page should support dark & light mode.
  - Use modals for creation/editing, not separate pages.
  - Keep primary actions visible on top.
  - Maintain consistent spacing, colors, fonts.
- 

### C. API & Database Best Practices

- All CRUD operations should map directly to your SQL schema.
  - Use optimistic updates for task interactions.
  - Always return meaningful error messages.
  - Validate input before saving to database.
- 

### D. Performance Considerations

- Paginate large lists (tasks, notifications).
  - Use indexing in Supabase (already supported).
  - Cache user profile & tags.
  - Use lazy loading for Task Details page.
- 

### E. Supabase-Specific Notes

- Enable RLS and test every resource access.
  - Use `auth.uid()` for filtering user-owned rows.
  - Store files in Supabase Storage, not in DB.
  - Use triggers/events for analytics if needed.
-

## F. UI Consistency Rules

- All forms must use the same style of input & submit button.
  - Task cards must look identical across Dashboard & All Tasks.
  - Use the same modal component everywhere.
- 

## G. Future-Proofing Notes

- Don't hardcode recurring rules; keep them database-driven.
  - Notifications should be modular for future push/email providers.
  - Events table should be extendable for analytics.
- 



## Final Note

This document serves as your **official development blueprint**. It gives you: - All screens needed - Their purpose - Their functionality - Development sequence - Key instructions & best practices

You can now move into UI/UX design or component creation confidently.