PROJECT SUMMARY – TASK MANAGER WITH AI, VOICE NAVIGATION & AUTOMATION

==

1. INTRODUCTION

This project is a complete full■stack Task Management System built using React (frontend) and FastAPI (backend).

It includes authentication, task CRUD operations, analytics, AI-powered productivity insights, PDF/CSV export,

email notifications, cron-based reminders, voice navigation, and a clean modern UI.

The system is designed for high performance, automation, and easy user workflow management.

----

2. FRONTEND SUMMARY (React + Vite)

The frontend is implemented using:

• React 18

• Vite bundler

• TailwindCSS for styling

• Lottie animations for UI enhancement

• Axios for API communication

• React Router v6 for routing

• Lucide-React icons for UI icons

• SpeechRecognition API for voice navigation

Frontend Technical Concepts Used:

• Component-based architecture

• State management using React Hooks (useState, useEffect)

• Protected routes with authentication

• Dynamic UI updates using conditional rendering

• Reusable components (Sidebar, Dashboard, Cards)

• LocalStorage for theme persistence and token storage

• Responsive grid layouts

• Lottie integration for animated pages

User-Facing Pages:

• Landing page with animation

• Login, Register, Forgot Password

• Dashboard with upcoming + overdue tasks

• List View (CRUD interface)

• Kanban board (drag & drop using status)

• Analytics page (charts + insights)

• Activity Log page

• Profile settings

• AI Summary page

• Voice Navigation integrated into Sidebar

----

3. BACKEND SUMMARY (FastAPI)

Backend is implemented using:

• FastAPI framework

• JWT Authentication

• MongoDB database

• ReportLab for PDF export

• python-dotenv for environment variables

• APScheduler for CRON jobs

• Requests library for AI API integration

• CORS middleware for frontend communication

Backend Features:

• Authentication with JWT

• Secure password hashing (bcrypt)

• CRUD operations for tasks

• File upload support

• Role-based access for admin endpoints

• Analytics aggregation (weekly completions, status counts)

• Activity logs for every action

• CSV + PDF export

• Automatic email reminders using cron jobs

• AI summary API route using Groq LLaMA 3.1 model

----

4. PROJECT FOLDER STRUCTURE (BACKEND)

app/

■■■ main.py → App entry point

■■■ auth/

■ ■■■ jwt_handler.py → Generates & validates JWT

■ ■■■ jwt_bearer.py → FastAPI dependency for protected routes

■ ■■■ user_routes.py → Login, Register, Password change

■■■ models/

■ ■■■ user_model.py → User DB functions

■ ■■■ task_model.py → Task CRUD functions

■ ■■■ activity_model.py → Activity logs

■■■ routes/

■ ■■■ task_routes.py → Task APIs

■ ■■■ ai_routes.py → AI summary API

■ ■■■ voice_routes.py → Voice command functions

■ ■■■ admin_routes.py → Admin-only endpoints

■■■ utils/

■ ■■■ email_sender.py → OTP & reminders

■ ■■■ scheduler.py → Cron jobs

■■■ schemas/

■ ■■■ task_schema.py → Task validation models

■ ■■■ user_schema.py → User validation models

----

5. VOICE NAVIGATION SYSTEM

Technology Used:

• Browser SpeechRecognition API

• Continuous listening mode

• Automatic + Manual switching

• Backend voice API for command processing

How It Works:

1. User taps MIC button.

2. Browser starts capturing speech continuously.

3. User says “manual mode” → Navigation disabled.

4. User says “automatic mode” → Auto page navigation starts.

5. Commands like:

• “Open dashboard”

• “Open kanban”

• “Go to analytics”

• “Create task”

trigger navigation without manual clicks.

Task Creation Using Voice:

• Speech → API → Title & description are extracted

• Backend creates new task instantly.

----

6. AI SUMMARY ENGINE

AI model used:

• Groq LLaMA 3.1 8B Instant model (fast & free tier)

Processing Steps:

1. Backend fetches tasks for the user.

2. Converts tasks into structured text.

3. Sends text to Groq API.

4. AI generates:

• Short productivity summary (5 lines)

• Weakness summary (5 lines)

• Improvement suggestions (5 lines)

• AI Productivity Score (0–100)

Purpose:

• Helps user understand performance trends.

• Detects blocked, overdue, duplicate tasks.

• Provides personalized improvement tips.

----

7. EMAIL OTP + PASSWORD RESET

Features:

• Send OTP via email using SMTP

• Verify OTP for resetting password

• Secure hashing of new password

• OTP expiration handling

Uses:

• login → forgot password

----

8. AUTOMATED CRON REMINDERS

Tools Used:

• APScheduler (BackgroundScheduler)

Daily cron job:

• Checks all tasks of all users

• Identifies “due tomorrow”

• Sends automatic email reminder

Example:

“Your task ‘Finish API’ is due tomorrow. Please complete it.”

----

9. TASK ANALYTICS ENGINE

Shows:

• Status chart (To-do, In-progress, Completed, Blocked)

• Priority chart (High, Medium, Low)

• Weekly productivity chart

Generated from MongoDB aggregation logic.

----

10. EXPORT SYSTEM (CSV + PDF)

CSV Export:

• Uses Python csv library

• Exports all tasks with fields

PDF Export:

• Uses ReportLab (Platypus)

• Generates clean tabular PDF report

----

11. OVERALL SYSTEM SUMMARY

This project is a fully advanced personal productivity system combining:

• Task management

• AI automation

• Speech-based navigation

• PDF/CSV export

• Analytics dashboards

• Reminders and email automation

• Clean modern UI

It demonstrates strong skills in:

Frontend (React), Backend (FastAPI), AI integration, voice automation, UI/UX,

database architecture, cron jobs, and real-world product development.

END OF DOCUMENT