

# Success Planner Backend – Full & Extended Development Plan (2025)

---

## 📌 Project Overview

Success Planner is a productivity app for personal users and admin. It includes authentication, user/admin roles, planners (yearly, monthly, weekly, todo), modules, and future-ready modules. It also adds notifications, file uploads, and activity logs.

## 📌 Technologies Used

- Node.js – Backend runtime
- Express.js – Routing & server
- MongoDB + Mongoose – NoSQL Database
- JWT + cookie-parser – Authentication via tokens
- bcryptjs – Password encryption
- Zod – Input validation
- Nodemailer – OTP/Email for password reset
- Cloudinary – Image uploads
- Helmet, CORS, Rate Limiting – Security
- dotenv – Config management

## 📌 Modules Breakdown

- Auth: Register, Login, Logout, OTP-based Reset, Email Verification
- Role Management: User can switch to/from Admin
- Dashboard: Yearly, Monthly, Weekly planner + To-Do List (Doing/Done)
- Admin Panel: Total users, stats, role control (only admin access)
- User Profile: Avatar update, profile data, switch role
- Notifications: Task reminders via Email
- Task History & Logs: User actions log for review
- File Uploads: Upload personal notes or planner-related documents

## 📌 Development Steps

1. Project initialize (npm init)
2. Folder structure setup (as per plan)
3. MongoDB connect (config/db.js)
4. user.model.js (role, password, avatar)

5. 5. Auth: register/login/logout (JWT cookie-based)
6. 6. Middleware ☐☐☐☐: auth, role, error
7. 7. switchRole API ☐☐☐☐ (PATCH /user/switch-role)
8. 8. Admin route (/admin/data) ☐☐☐☐
9. 9. Dashboard planner: model, controller, routes
10. 10. Notifications system: reminders via email
11. 11. File Uploads setup using Cloudinary
12. 12. Task Logs module for tracking
13. 13. Zod validation ☐☐ input ☐☐
14. 14. Security: Helmet, Rate Limit, JWT secure
15. 15. Postman ☐☐ test ☐☐☐☐
16. 16. Deploy ☐☐☐☐

## Security Checklist

- JWT in secure, HTTPOnly cookies
- Zod validations for all user input
- Bcrypt hashed passwords
- Role switch only by authenticated users
- Role-based route protection (restrictTo("admin"))
- CORS + Helmet + Rate Limit enabled
- Email verification before role-switch
- Cloudinary upload security using signed URLs

## Final Notes

☐☐ ☐☐ ☐☐ services/controller ☐☐ ☐☐-☐☐ ☐☐☐ User ☐☐ manually admin  
☐☐☐☐ ☐☐ switch ☐☐☐☐ smart ☐☐, ☐☐☐☐ protected ☐☐☐☐☐☐ ☐☐ UI  
☐☐ ☐☐☐ authorized sections ☐☐☐☐ based on role ☐☐ ☐☐☐ testing ☐☐☐☐:  
 auth → middleware → role → dashboard ☐ Notifications, file uploads, ☐☐ history logs future  
 SaaS feature ☐☐ ☐☐ ☐☐ ☐☐☐☐☐☐