Secure File Sender Web Application - Project Plan

# 1. Deployment Context

- Environment: Linux server with cPanel (no Docker, no Kubernetes)  
- Deployment: Simple file-based deployment (via FTP or cPanel File Manager)  
- Database: SQLite (or PostgreSQL if preferred)  
- Email: SMTP (cPanel or third-party like SendGrid/Mailgun)

# 2. Tech Stack

## Frontend (Admin SPA)

- Framework: SvelteKit  
- Design Style: Professional, modern, elegant, slightly futuristic  
- Deployment: Static assets or bundled into backend wwwroot

## Backend

- Language/Framework: ASP.NET 8 Web API  
- Database: SQLite (or PostgreSQL)  
- File Handling: Filesystem-based with compression  
- Auth: JWT-based login for admins

# 3. Admin Portal Features

- Form with inputs for email, receiver details, files, expiration  
- Upload & compress files (non-ZIP)  
- Generate one-time link and secure passcode  
- Send email with link and passcode  
- Dashboard: client list, statuses, download count, cleanup tools  
- Role-based login (SuperAdmin, Admin, Viewer)  
- QR code generation, audit log view, theme/branding support

# 4. Customer Flow

- Open secure link: greeting + passcode prompt  
- Passcode validation with optional CAPTCHA  
- On success: decompressed file download, link is invalidated  
- Expiry is enforced  
- Mobile-ready, multilingual, with QR access

# 5. Security Features

- Hashed + salted passcodes (bcrypt/PBKDF2)  
- CAPTCHA integration (reCAPTCHA or hCaptcha)  
- Rate limiting and HTTPS enforced  
- Link expiration and single-use enforcement  
- Secure file paths and access control  
- Audit and IP logging

# 6. Compression / Decompression

- Each uploaded file is compressed (e.g., GZip or LZ4)  
- Files decompressed on-the-fly during download  
- ZIP archives are not used (business requirement)

# 7. Code Guidelines

## Backend

- Extremely clean, modular, and readable  
- Central config file (appsettings.json) with feature toggles  
- Layered architecture: Controllers → Services → Repos

## Frontend

- Modular SvelteKit components  
- Consistent styling and clean file structure  
- No config file, feature behavior derived from backend flags