Final System Workflow - EduPrajna Employee Reporting App

1. Employee Login Page

Fields:

- Employee ID

- Password

Validation:

- Check against credentials stored in the database (you can hash passwords for basic security).

On Success:

- Redirect to form page

- Auto-fill Employee Name + ID

2. Daily Report Submission Form

Fields:

1. Number of Dials

2. Connected Calls

3. Positive Prospect

4. Dead Calls

5. Demos

6. Admission

7. Client Visit

8. Client Closing - with software/digital marketing 9. Backdoor Calls - numbers & textbook follow-ups 10. Posters Done (optional)

Submit Button:

- On click -> Save to database -> Show message: "Thank you, have a nice day!"

3. Backend/API Layer

Use Next.js API routes for:

- Login authentication

- Storing form submissions

- Admin queries (daily/weekly/monthly exports)

ORM: Drizzle ORM

DB: Neon PostgreSQL

4. Admin Dashboard

Admin login with secure credentials

Features:

- View all submissions by employee

- Filter by:

- Date range (daily/weekly/monthly)

- Employee ID or name

- Export as CSV or Excel (day report,weekly report, monthly report)

- (Optional) Visual reports: charts, tables Folder Structure (Next.js App Router)

/app

/login # Employee login

/form # Daily report form (protected) /admin/login # Admin login

/admin/dashboard # Admin dashboard

/lib

db.ts # Neon connection

schema.ts # Drizzle schema

/pages/api

login.ts # API for login auth

submit.ts # API for form submission report.ts # API for admin report queries

/public

logo.png

.env.local # Neon DB credentials, secret keys

Next Steps:

Would you like me to now generate a starter project zip or give:

- Auth system (employee + admin)

- Daily report form UI (Tailwind + Framer Motion)

- API routes

- Drizzle DB schema

- Download button for reports

Say: "Yes, generate the starter project" and I'll build the full base code ready to run on Replit + Vercel + Neon.