Build & Deploy a Notes App | Next.js 15, React, Shadcn, Better Auth

Build & Deploy a Notes App | Next.js 15, React, Shadcn, Better Auth

https://www.youtube.com/watch?v=oSe_HilcQmM

Better Auth Tutorial – Email & Password Authentication with Email Verification

https://www.youtube.com/watch?v=-qJ4-H00e0g

```
npx create-next-app@latest
```

```
cd <>
git remote add origin https://github.com/ck-h/noteforge.git
pnpm dev

# shadcn -> docs -> nextjs
pnpm dlx shadcn@latest init
npx shadcn@latest init
# color: stone
pnpm dlx shadcn@latest add button
npx shadcn@latest add button

# -> https://tailark.com/ -> Explore blocks
pnpm dlx shadcn@latest add https://tailark.com/r/hero-section-2.json
pnpm dlx shadcn@latest add https://tailark.com/r/features-1.json
pnpm dlx shadcn@latest add https://tailark.com/r/call-to-action-1.json
pnpm dlx shadcn@latest add https://tailark.com/r/footer-1.json
pnpm add next-themes
# Create a theme provider -> create the file at components and copy
the text
# follow the rest of the steps in that page
# add file mode-toggle.tsx to components and copy the code
# add the dropdown-menu needed to the toggle:
pnpm dlx shadcn@latest add dropdown-menu
# then place the toggle where applicable. I put it in the Header
component
```

Better Auth

```
# -> better auth -> docs -> installation
pnpm add better-auth

# generate secret key -> copy it
# in the root of the project, create .env file and paste the key
# also for base url
# and the rest of the instructions
# there are some option for location/ we'll go with `lib` dir
# create `lib` at the root of the project -> auth.ts -> paste

# step 4 configure db -> replace existing code in auth.ts with the
drizzle code

# Setup the DB
# drizzle -> documentation -> tutorials -> Drizzle with Neon Postgres
(or better yet: Todo App with Neon Postgres)
pnpm add drizzle-orm
pnpm add -D drizzle-kit
pnpm add @neondatabase/serverless
# we don't need: pnpm add dotenv -> we already use node.js
# in npm round I installed:
npm i dotenv

# Create NEON project (login w/GH)
# connect -> connection string -> copy
# in .env add: DATABASE_URL=<paste> (without ' or ")

# connect drizzle to the DB:
# @root Create folder: db, create drizzle.ts in db -> paste
# In `auth.ts` update the path to import db to "@/db/drizzle"
# Create a drizzle.config.ts file in the root of your project and paste
# !!! Remove `src/` from the path to `schema.ts`
# create `schema.ts` at db folder

# Back to better auth docs, step 5:
npx @better-auth/cli generate
# yes to install, yes to create `auth-schema.ts`. this creates the file at root.
# Copy all the code to `schema.ts`. Remove Integer - it was never used
# We can delete `auth-scheme.ts`
# at the bottom of `scheme.ts` export the tables:
export const schema = {
user,
session,
account,
verification,
};
# In auth.ts, after provider, add:
schema: schema, // import your schema from db/schema.ts

# Push the tables to NEON
npx drizzle-kit push

# step 6 in better auth, only for mail:
emailAndPassword: {
enabled: true,
},

# step 7: create catch-all route
# In next-js we use the app router, and not the pages router

# step 8
# In `lib` create auth-client.ts -> paste
# authClient contains the methods to login, logout, and check authentication status

# better-auth docs -> Integrations -> Next -> Server Action Cookies -> add to auth.ts:
plugins: [nextCookies()] // make sure this is the last plugin in the array

# Middleware -> create `middleware.ts` in the root -> paste
# נועד לאמת על פי עוגיות, במקום לבצע קריאה בכל פעם לממשק לוודא הרשאות
# בסוף בהתאמות צריך לרשום את כל הראוטים שרוצים שיחול עליהם
```

כעת צריך ליישם את הממשקים למשתמש
ניתן דרך השרת או דרך הלקוח
בפרודקשן מעדיפים דרך השרת עם API
Under `app` we created `(client)` an `(server)` and the appropriate folders and pages for routes. Now we move to the `components`. Again. client and server, and 2 components for each.

```
# -> betterauth -> concepts -> API
https://www.better-auth.com/docs/concepts/api

# on the root we create `server` folder, and inside `users.ts`
```

Send verification email

sendEmail verification: create folder `email` - logic for sending the mail, `mail` - for the template

Send email using Resend

react.email/docs/integrations/resend

```
pnpm add resend @react-email/components
npm install resend @react-email/components
```