Commands:

npx create-next-app@latest

npm install prisma --save-dev

npm install @prisma/client

npx prisma init --datasource-provider postgresql

Dependencies:

TypeScript, ESLint, TailwindCSS, App Router, Turbopack

Day 1:

1. Created the prisma DB models inside prisma>schema.prisma, and pushed it to the DB.
2. npx prisma migrate dev --name first push   
   Error: can’t reach server => put prisma folder in the inner folder “blog”/
3. npm run dev
4. npx shadcn@latest init
5. npx shadcn@latest add button
6. Started by making page.tsx under (home) folder, which will import the navbar and other components.
7. Created a “component” folder, under which home and ui folders were made. Under home, we had header.tsx under which navbar.tsx was first designed, followed by search-input and the toggle mode.
8. npm install next-themes
9. layot.tsx has {children} which is wrapped with ThemeProvider component of shadcn
10. npx shadcn@latest add dropdown-menu
11. npx shadcn@latest add input label
12. The Navbar was completed, but alignment issues persist.
13. Work was done on the menu bar and making it for mobile applications.

Day 2:

1. npm install @clerk/nextjs; documentation for clerk was followed.
2. Integrating clerk user creation into the database: either by webhook method or simply storing credentials on signup.
3. Issue arose when @primsa/client was not resolved through npx prsima generate => npm install prisma --save-dev (folder structure was also revised to fit into the git structure) then npx prisma generate
4. npx shadcn@latest add table badge sheet card
5. Worked on dashboard and create article page.
6. Worked on actions (which is an alternative to apis)
7. Worked on implementing cloudinary and Zod validation for the create articles, sending the data to database; common issues encountered is name of cloudinary env variable name not correct, and another was a silent error that occurred due to submitting without an image.
8. Had minor issues with TypeScript, like having a return for every case, type mismatch, etc.
9. Should learn about array buffers and TS as well.

Day 3:

1. Issue of comment not being posted => comments.map was not working due to parenthesis order issue
2. Used optimistic updating logic for fast user UI updates.
3. Like dislike issue fixed by using state variables to keep track and keeping track of the updatedLikeCount locally in like-dislike.ts action, preferring a if-else block instead of two await blocks; also returned the likeCount and isLiked status to the frontend