

AWT

i. Prisma

I learned that Prisma is an ORM (Object Relational Mapping) that helps us to interact with the database using JS instead of writing raw MongoDB queries / or any other ~~raw~~ raw queries.

In my project, I used Prisma to define models. Quote
It made data access more structured & easier to maintain.

> I used Prisma methods like `prisma.quotes.create()` & `findMany()` to add & fetch data.

ii. Express & RESTful Routes

In my project:

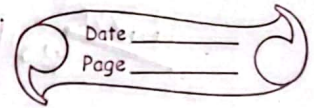
- GET /quotes : fetches all quotes from the database.
- POST /quotes : creates a new quote using data from the frontend.

iii. Middleware

Middleware fns like - `express.json()` (parses JSON data)
- `express.urlencoded()`

help Express understand the incoming request bodies.
`express.urlencoded({ extended: true })` parses data from HTML forms.

Harsh
Roll no: 28



iv. CORS

Cross Origin Resource Sharing

By default, browser block requests from different origins.

To allow the frontend to talk backend, I used cors package.

v. Prisma Studio

It is a visual editor for the db.

I launched it by running "npx prisma studio" in the terminal. This opened a browser window where I could:

- view all entries in the Quote table
- Add, edit, or data manually.

In my project,

I used .env file to store sensitive data i.e. my MongoDB connection string.

I in ~~prisma~~ schema.prisma

changed provider to mongodb before starting.

JSON APIs

JSON APIs send & receive data in JSON format, which is widely supported & easy to use.

```
app.get('/users', async (req, res) => {  
  const users = await prisma.user.findMany()  
  res.json(users)  
})
```

//

// returns a JSON array of users

Migration

Migration is the change to the database schema. It's a way to update the structure of the database to keep it up-to date.

To create a migration:

```
npx prisma migrate dev
```

The CLI will make a new migration file based on the changes we want to make to the database.