AWT

SQL VS NoSQL Datasets

SQL
Data model :- Relational (Tables with rows + columns, relationships via foreign keys)

Schema : Strict, requires, predefined structure (columns, datatypes)

Query Language : SQL

NoSQL
Data Model : Document - oriented (JSON like documents within collections).

Schema : Flexible
Query Language : MongoDB Query language (based on JS)

Prisma ORM:

↳ A modern ORM for Node.js / Typescript to interact with databases using TS / JS code instead of raw SQL.

Benefits :-
↳ Automatically generates migrations, reducing errors.
↳ Provides type safety, auto completion and simplified db interaction.

# Middleware:-

↳ Software that ~~sits~~ sits betw^n input and response, handling tasks like logging authentication, or request modification.

# Routing:

↳ Determines how the app responds to various URLs and HTTP Method

eg.

```
app.get ("/user", (req, res) => {
        res.send ("Retrieve all users")
})
```

# JSON APIs:-

↳ API's that communicate using JSON format (widely supported).

eg.
```
app.get('/quotes', async (req, res)=>{
      const ~~[illegible]~~ quotes = await prisma.
                                      quote.findMany()
      res.json(quotes)
})
```

Displaying Backend Data on HTML Page using fetch()

① Create a server that serves JSON data.
② In frontend use fetch() method.

e.g.

```
fetch ("http://localhost:7000/api/quote")
    .then (res => res.json())
    .then (~~quote~~ data => {

    ~~data.quote~~ data .forEach ((quote) => {
        const div = document.createElement
                                        ("div");
        div.className = "quote";
        div.innerHTML = `
              <p>${quote.text} </p>
           ~~quoteContainer~~

    });
});
```