Understanding Sessions in Web Development

What is a Session?

A session is a way to store information about a user across multiple requests to the server. It allows the server to remember the user without asking them to re-authenticate each time.

How Sessions Work (Traditional Flow)

1. User logs in by submitting credentials.
2. Server verifies and creates a session object (e.g., { sessionId: 'abc123', userId: 'nikhil123' }).
3. Server sends a session ID to the client in a cookie.
4. Client includes the session ID in future requests.
5. Server identifies the user using the session ID.

Where is Session Data Stored?

- On Server: In memory or in a database (e.g., Redis).
- On Client: Only the session ID (usually in cookies).

Alternative: JWT-Based Authentication

JWTs (JSON Web Tokens) are stateless and don't require storing session data on the server. The server issues a signed token after login, which the client includes in request headers.

When to Use What?

- Sessions (stateful): Simple apps, secure login/logout handling.
- JWT (stateless): Scalable systems, APIs, mobile + web apps.

Session Example in Express (Node.js)

```
const session = require('express-session');

app.use(session({
  secret: 'my-secret',
```

```
  resave: false,
  saveUninitialized: true
}));
```

Accessing session: req.session.user = { id: 'nikhil123' };