1. Explique o que é JWT e quais são suas vantagens.
JWT (JSON Web Token) é um formato de token usado para autenticação e autorização. Ele armazena dados codificados e pode ser verificado sem consultar o banco. É leve, seguro e funciona bem com APIs.

2. Qual a diferença entre sessões e cookies?
Cookies armazenam dados no navegador do usuário, como identificadores. Sessões guardam dados no servidor, ligados ao cookie do usuário. Sessões são mais seguras para informações sensíveis.

3. Defina CORS e por que é importante configurá-lo corretamente.
CORS (Cross-Origin Resource Sharing) define quais domínios externos podem acessar recursos da sua API. Sem uma configuração correta, você pode permitir acessos não autorizados.

4. O que é CSRF e como prevenir este tipo de ataque?
CSRF (Cross-Site Request Forgery) engana o navegador para enviar ações indesejadas autenticadas. Pode ser evitado com tokens CSRF, validação de origem e uso de métodos seguros como POST.

5. Explique o conceito de SQL Injection e seus riscos.
SQL Injection ocorre quando comandos maliciosos são inseridos em consultas SQL. Isso pode expor, modificar ou excluir dados do banco. É uma falha grave de segurança.

6. Como o atributo HttpOnly ajuda a proteger cookies?
O atributo HttpOnly impede que o cookie seja acessado por scripts JavaScript, protegendo contra ataques XSS que tentam roubar tokens de sessão.

7. Quais são as principais partes de um token JWT?
Um JWT possui três partes:

- **Header**: tipo do token e algoritmo usado
- **Payload**: dados (como ID do usuário)
- **Signature**: assinatura digital para validar a autenticidade

8. Liste três boas práticas para evitar SQL Injection em aplicações Node.js.

1. Usar prepared statements com parâmetros.
2. Validar e sanitizar entradas do usuário.
3. Nunca montar consultas SQL com concatenação de strings.

9. O que é autenticação baseada em roles e quais são seus benefícios?
É uma forma de controlar permissões com base em funções (roles) dos usuários, como “admin” ou “cliente”. Garante acesso restrito e facilita a organização do sistema.

10. Como o middleware csrf auxilia na proteção de aplicações web?
Esse middleware gera tokens únicos para cada sessão e valida se o token está presente nas requisições POST. Isso bloqueia ações maliciosas externas (CSRF) que tentam enganar o usuário logado.