Universidade Federal de Uberlândia
Faculdade de Computação – Prof. Daniel A. Furtado
7º Trabalho de Desenvolvimento Web II
Requisições Cross-Origin, CORS

Instruções Gerais

- Esta atividade deve ser realizada **individualmente**;
- Tecnologias permitidas: HTML5, CSS, JavaScript, Bootstrap, PHP, MySQL, API Fetch com async/await. O objeto XMLHttpRequest **não é permitido** neste trabalho. O método .**then** não é permitido.
- Aspectos de segurança devem ser considerados para evitar ataques XSS e SQL Injection;
- Sintaxe da XHTML como <img/> ou
 não é permitida (anulará o trabalho);
- O website deve ser hospedado e disponibilizado online, conforme orientações disponíveis no final deste documento;
- Ao construir o website, utilize dados fictícios. **Jamais utilize** dados pessoais como seu nome, CPF, endereço, e-mail etc.;
- Esteja atento às **observações sobre plágio** apresentadas no final desde documento;
- Trabalhos com implementações utilizando trechos de códigos retirados de sites da Internet ou de trabalhos de semestres anteriores serão anulados;
- As páginas web não devem conter qualquer conteúdo de caráter imoral, desrespeitoso, pornográfico, discurso de ódio, desacato etc.;
- O website deve ser validado utilizando as ferramentas disponíveis nos endereços **validator.w3.org** e **jigsaw.w3.org/css-validator** (não deve conter nenhum erro ou *warning*);
- O trabalho deve ser entregue até a data/hora definida pelo professor. Não deixe para enviar o trabalho nos últimos instantes, pois eventuais problemas relacionados à eventos adversos como instabilidade de conexão, congestionamento de rede etc., não serão aceitos como motivos para entrega da atividade por outras formas ou em outras datas;
- Este trabalho deve ser feito **mantendo os trabalhos anteriores intactos**, ou seja, os trabalhos anteriores devem permanecer online conforme foram entregues;
- Trabalhos enviados por e-mail ou pelo MS Teams **não serão considerados**.

Material de Apoio

https://furtado.prof.ufu.br/site/teaching/DW2/DW2-Modulo4-HTTP-Ajax.pdf

https://furtado.prof.ufu.br/site/teaching/PPI/PPI-Modulo4-JavaScript.pdf

Exercício 1

Faça uma cópia do **Exercício 1** do trabalho anterior e coloque a cópia online em pasta específica deste trabalho. Teste o exemplo no navegador utilizando apenas **HTTP** (sem HTTPS) na barra de endereços (o Chrome deve apresentar a mensagem **Not secure** ao lado da barra). Em seguida, altere o código JavaScript para que a requisição Ajax seja realizada utilizando o endereço completo, com **HTTPS** (fetch(https://seusite.infinityfree/trab7/ex1/busca-endereco.php...). Acesse novamente o formulário de endereço online, **sem** utilizar HTTPS, e observe se a funcionalidade de preenchimento de endereço continua ativa. Investigue a causa analisando as requisições no

ambiente de desenvolvimento do navegador. Explique o motivo do problema inserindo comentários no código.

Exercício 2

Faça uma nova cópia do **Exercício 1** do trabalho anterior. Crie um novo subdomínio no infinityfree e mova apenas o arquivo **busca-endereco.php** para o novo subdomínio. Faça os ajustes no código JavaScript para que os dados de endereço do site anterior sejam buscados utilizando o **busca-endereco.php** do novo site. Teste a funcionalidade abrindo o site anterior no navegador. A funcionalidade de preenchimento de endereço funcionou? Investigue as requisições no **Dev Tools** do navegador e responda (crie um arquivo .txt para as respostas):

a) A requisição Ajax disparada é simples ou complexa?
b) A requisição Ajax disparou um preflight? Por quê?
c) Qual cabeçalho deveria estar na resposta HTTP da requisição Ajax para que o serviço funcionasse?
d) Altere a requisição Ajax para que uma requisição **preflight** seja disparada pelo navegador quando o usuário preencher o CEP. Registre um print da tela mostrando detalhes da requisição **preflight** no Dev Tools. Disponibilize a imagem do print no final da página principal do exercício. Explique por que a requisição modificada disparou o **preflight**.
e) Experimente modificar a requisição para que o modo 'no-cors' seja utilizado. Analise novamente a resposta HTTP no navegador. Resolveu o problema? Explique.
f) Altere o código JavaScript para que a API **ViaCep** seja utilizada ao invés de **busca-endereco.php**. Analise novamente as requisições no navegador. Verifique a presença dos cabeçalhos relacionados ao CORS na resposta HTTP.
g) Crie um script PHP para realizar a contagem de acessos do website e hospede o script utilizando o novo domínio. Quando o script for acessado, ele deve incrementar o contador (armazenado em arquivo), e apresentar a contagem atual utilizando o construtor **echo**. Crie uma página HTML simples (hello.html) no **domínio antigo** que envie uma requisição Ajax para o script contador do **domínio novo** assim que a página for aberta pelo usuário. A contagem não precisa aparecer na página **hello.html**.

Entrega

Compacte os arquivos e envie pelo Sistema Acadêmico de Aplicação de Testes (SAAT) até a data limite indicada pelo professor em sala de aula.

Sobre Eventuais Plágios

Este é um trabalho individual. Os alunos envolvidos em qualquer tipo de plágio, total ou parcial, seja entre equipes ou de trabalhos de semestres anteriores ou de materiais disponíveis na Internet (exceto os materiais de aula disponibilizados pelo professor), serão duramente penalizados (art. 196 do Regimento Geral da UFU). Todos os alunos envolvidos terão seus **trabalhos anulados** e receberão **nota zero**.