6 principais erros:

1-

```
<!-- O atributo action do formulário está apontando para "idex.php", deveria ser "index.php",
 para que o arquivo processe o formulário. -->
<form action="idex.php" method="post">
```

Quando um usuário preenche o formulário e clica no botão de enviar, o navegador tenta acessar o arquivo especificado no atributo action. Se o arquivo não existe, o servidor web retorna uma mensagem de erro, e os dados submetidos não são processados como esperado. Nessa ocasião o arquivo deveria ser index.php

2-

```
$cep = $_POST['cep'];


// Dentro da função get_address,
// você se refere a ($cp) que não foi definida anteriormente. O correto seria ($cep), que é o parâmetro recebido pela função.
$address = (get_address($cp));
```

O uso de $cp resulta em um erro de "variável não definida" porque o script tenta acessar uma variável que não foi inicializada ou declarada antes de ser usada. Pois o nome da variável inicializada foi $cep logo essa deveria ser a variável usada ao chamar o método get_address

3-

```php
$address = (get_address($cp));



echo "<br><br>CEP Informado: $cep<br>";


// A variável deveria ser $address, não $address
echo "Rua: $addres->logradoro<br>";
```

A variável $addres está incorreta. O nome correto, conforme definido anteriormente no script, é $address. Usar uma variável não definida pode resultar em um aviso de "variável não definida"

4-

```php
// A propriedade deveria ser logradouro, não logradoro
echo "Rua: $addres->logradoro<br>";
```

A propriedade logradoro é um erro de digitação para logradouro. O uso incorreto da propriedade de um objeto resulta em um erro de "propriedade não encontrada". Pois o PHP está tentando acessar um elemento que não existe no objeto retornado da função

5-

```php
// a variável está errada e deveria ser $address
echo "Estado: $adress->uf<br>";
```

A variável $adress não foi definida em qualquer lugar do script anteriormente. Portanto, quando o PHP tenta acessar essa variável, ele não consegue encontrar uma definição para ela, retornando um erro de função indefinida

6-

```php
//  a URL está mal formada. Deveria ter uma barra (/) entre ws e $cep, assim: "http://viacep.com.br/ws/$cep/xml/"
$url = "http://viacep.com.br/ws$cep/xml/";
```

Nesta linha, a concatenação do CEP diretamente após ws não inclui uma barra (/) entre ws e o valor da variável $cep. A falta dessa barra resulta em uma URL que não segue a estrutura esperada pelo serviço ViaCEP, o que levará a uma resposta de erro ao envio de dados.

Alguns pontos que reparei:

```html
<!-- A tag <title> deve estar dentro da tag <head>   -->
<title> MEU CEP </title>
```

A tag <title> é utilizada diretamente no corpo do documento ou fora da tag <head>. Isso é considerado uma prática inadequada porque a tag <head> é destinada a conter metadados (dados sobre dados) que definem configurações da página e informações que não são diretamente exibidas no conteúdo visível ao usuário.

```
<body>
    <!-- O atributo action do formulário está apontando para "idex.php", deveria ser "index.php",
     para que o arquivo processe o formulário. -->
    <form action="idex.php" method="post">
    <label> Insira o CEP: </label>
    <input type="text" name="cep">
    <input type="submit" value="Enviar">


    <!-- A tag <form> não está fechada. -->
</body>
```

Quando você cria um formulário HTML usando a tag <form>, é fundamental fechá-la adequadamente com uma tag de fechamento </form>. Isso indica ao navegador onde o formulário termina. A falta dessa tag de fechamento pode causar comportamentos inesperados

```
// A linha $address = (get_address($cp)); possui parênteses desnecessários
$address = (get_address($cp));
```

Embora os parênteses extras não afetem a funcionalidade, eles podem adicionar uma camada desnecessária de complexidade visual ao código. Isso pode potencialmente confundir outros desenvolvedores ou até mesmo você mesmo ao revisitar o código, fazendo com que se pergunte se há uma razão especial para esses parênteses.

O que eu fiz:

Realizei correções no código para garantir o funcionamento adequado do projeto e também aprimorei o design visual para melhorar a experiência do usuário.

O que eu poderia melhorar:

Para melhorar a legibilidade do código e facilitar futuras melhorias, eu poderia separar o HTML e o PHP em arquivos distintos. Também deveria assegurar o formato do CEP; embora eu já esteja removendo caracteres não numéricos com

preg_replace, é uma boa prática validar ainda mais esses dados para garantir que estão dentro de um formato esperado, por exemplo, checando o comprimento do CEP. Validando o CEP também pelo lado do usuário, eu posso fazer com que ele corrija o erro antes de enviá-lo e implementar tratamento de erro em vários casos, como por exemplo, caso o CEP não seja encontrado.