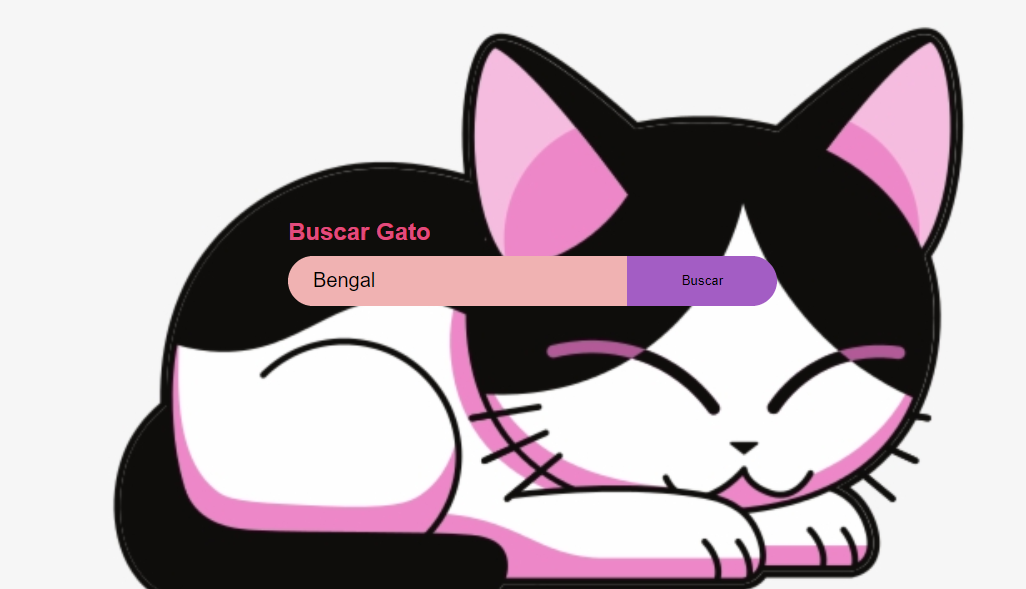
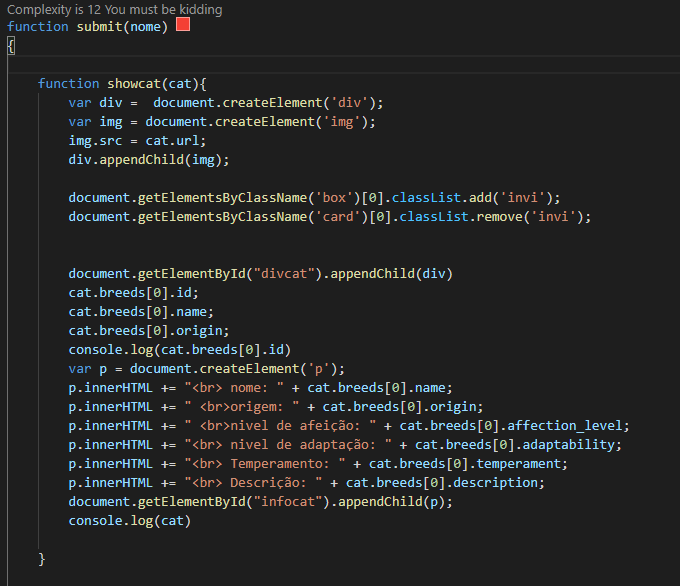
A pesquisa é feita digitando a raça do gato no buscador, se o gato existir vai aparecer as informações dele num card. Senão existir vai aparecer um erro.



Nessa função, após clicar no botão buscar, irá aparecer as informações do gato, que foram pegas pelo cat.breeds[0] e a informação que quero que apareça para o usuário. Também temos o img que é para aparecer a imagem do gato.



Nessa parte ele vai buscar o gato, e verificar se ele existe, se existir vai mostrar a imagem e os dados, senão vai retornar erro. O nome.preventDefault() foi usado para não recarregar a página após clicar no botão buscar.



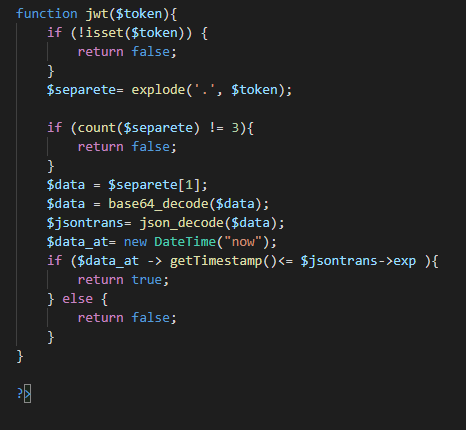
Aqui é a autenticação jwt no arquivo ligação.php, gerando a o token, e no main.php ele irá fazer a validação.



Na função “api” é pego a api do gato, verificado se tiver o nome do gato ele busca. Também é passado a chave da api, e no foreach a gente pega a imagem referente ao gato ao conseguir a imagem dos gatos buscados, é feito um array com elas, juntando as informações da imagem e da raça do gato. por fim, a função retorna um json contendo "breeds" (as raças dos gatos) e dentro de cada uma, todas as informações pertinentes, incluindo a imagem a ser mostrada ao usuário



Aqui é a validação do JWT, se não existir o $token vai retornar false. O separate é pra separar o $token, pois ele vem em 3 partes, e precisamos pegar a segunda parte do token, por isso $separate[1], e se diferente de 3 retorna falso. E também como no arquivo ligação demos a validade de 1 mês, se no data\_at ele pegar a data e for maior que a validade, irá ser preciso gerar outro token



No inicio do arquivo fizemos o caching , que verifica se o arquivo cache existe e se é válido, após isso lê o arquivo cacheado, se o arquivo não existir ele acessa a versão dinâmica da api e cria o cache

