Q-1. You have a Laravel application with a form that submits user information using a POST request. Write the code to retrieve the 'name' input field value from the request and store it in a variable called $name.

Answer: To retrieve the 'name' input field value from the request and store it in a variable called $name we need to write the following code:

```
$name = $request->input('name');
```

Q-2. In your Laravel application, you want to retrieve the value of the 'User-Agent' header from the current request. Write the code to accomplish this and store the value in a variable called $userAgent.

Answer: To retrieve the value of the 'User-Agent' header from the current request in my Laravel application and store it in a variable called $userAgent, the following code can be used:

```
$userAgent = $request->header('User-Agent');
```

Q-3. You are building an API endpoint in Laravel that accepts a GET request with a 'page' query parameter. Write the code to retrieve the value of the 'page' parameter from the current request and store it in a variable called $page. If the parameter is not present, set $page to null.

Answer: To retrieve the value of the 'User-Agent' header from the current request in my Laravel application and store it in a variable called $userAgent, the following code can be used:

```
$page = $request->query('page', null);
```

After executing this code, the value of the 'page' query parameter will be stored in the $page variable. If the 'page' parameter is not present in the request, $page will be set to null as specified in the default value in the second parameter.

Q-4. Create a JSON response in Laravel with the following data:

```
{
  "message": "Success",
  "data": {
    "name": "John Doe",
    "age": 25
  }
}
```

Answer:

```
return response()->json([
  'message' => 'Success',
  'data' => [
    'name' => 'John Doe',
    'age' => 25
  ]
]);
```

Q-5. You are implementing a file upload feature in your Laravel application. Write the code to handle a file upload named 'avatar' in the current request and store the uploaded file in the 'public/uploads' directory. Use the original filename for the uploaded file.

Answer: To handle a file upload named 'avatar' in a Laravel application and store the uploaded file in the 'public/uploads' directory using the original filename, you can use the following code:

```
if ($request->hasFile('avatar') && $request->file('avatar')->isValid()) {
    $file = $request->file('avatar');
    $filename = $file->getClientOriginalName();
    $path = $file->storeAs('uploads', $filename, 'public');
    return response()->json([
        'message' => 'File uploaded successfully',
        'path' => $path
    ]);
} else {
    return response()->json([
        'message' => 'Invalid file or no file uploaded'
    ], 400);
}
```

Q-6. Retrieve the value of the 'remember_token' cookie from the current request in Laravel and store it in a variable called $rememberToken. If the cookie is not present, set $rememberToken to null.

Answer: To retrieve the value of the 'remember_token' cookie from the current request in Laravel and store it in a variable called $rememberToken, with the fallback of setting $rememberToken to null if the cookie is not present, you can use the following code:

```
$rememberToken = $request->cookie('remember_token', null);
```

After executing this code, the value of the 'remember_token' cookie will be stored in the $rememberToken variable. If the cookie is not present, $rememberToken will be set to null as specified in the default value that is specified in the second parameter.

Q-7. Create a route in Laravel that handles a POST request to the '/submit' URL. Inside the route closure, retrieve the 'email' input parameter from the request and store it in a variable called $email. Return a JSON response with the following data:

```
{
  "success": true,
  "message": "Form submitted successfully."
}
```

Answer: To create a route in Laravel that handles a POST request to the '/submit' URL, retrieves the 'email' input parameter from the request, stores it in a variable called $email, and returns a JSON response with the provided data, you can use the following code:

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
Route::post('/submit', function (Request $request) {
  $email = $request->input('email');
  return response()->json([
    'success' => true,
    'message' => 'Form submitted successfully.'
  ]);
});
```