PDF SIGNING USING DIGITAL SIGNATURE

Overview:

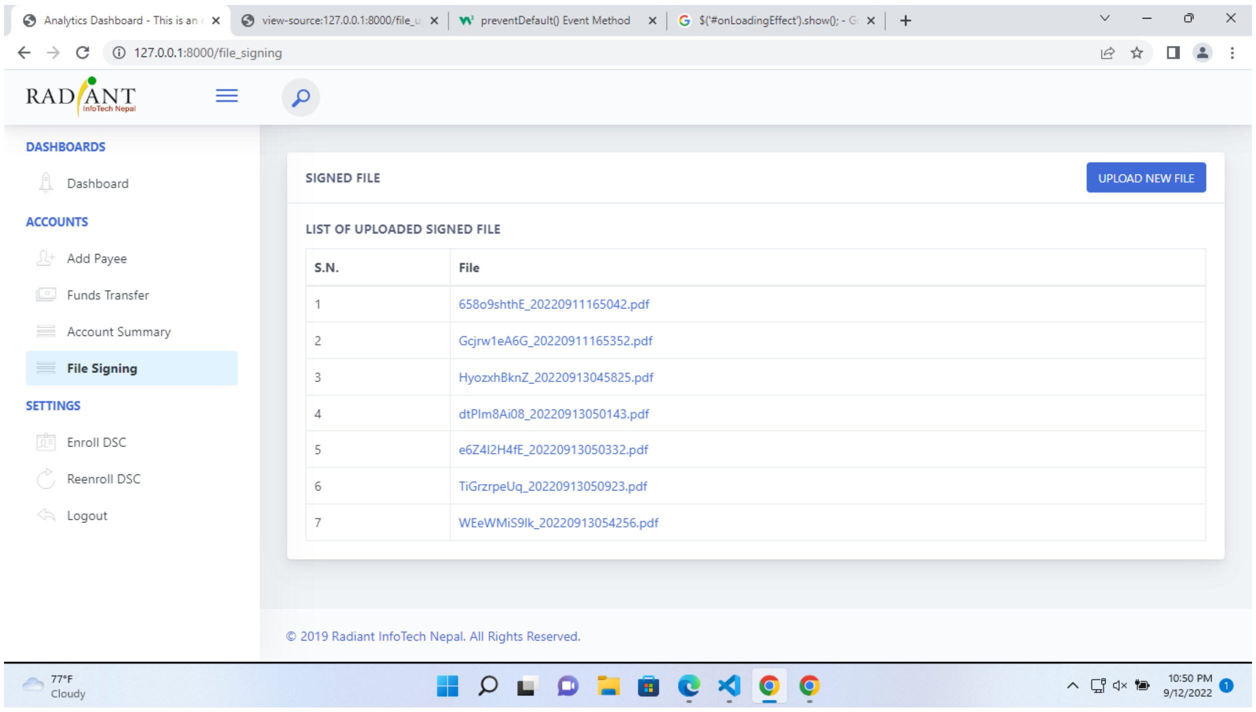
Introduction:

Problem Statement:

Prerequisite:

Steps for signing PDF using digital signature:

1. When user login then this type of page displayed in the screen



127.0.0.1:8000/file_signing

SIGNED FILE

LIST OF UPLOADED SIGNED FILE

S.N.	File
1	658o9shthE_20220911165042.pdf
2	Gcjrw1eA6G_20220911165352.pdf
3	HyozxhBknZ_20220913045825.pdf
4	dtPlm8Ai08_20220913050143.pdf
5	e6Z4I2H4fE_20220913050332.pdf
6	TiGrzrpeUq_20220913050923.pdf
7	WEeWMiS9lk_20220913054256.pdf

© 2019 Radiant InfoTech Nepal. All Rights Reserved.

2. When click in file signing then ,

   Url is:- http://127.0.0.1:8000/file_signing

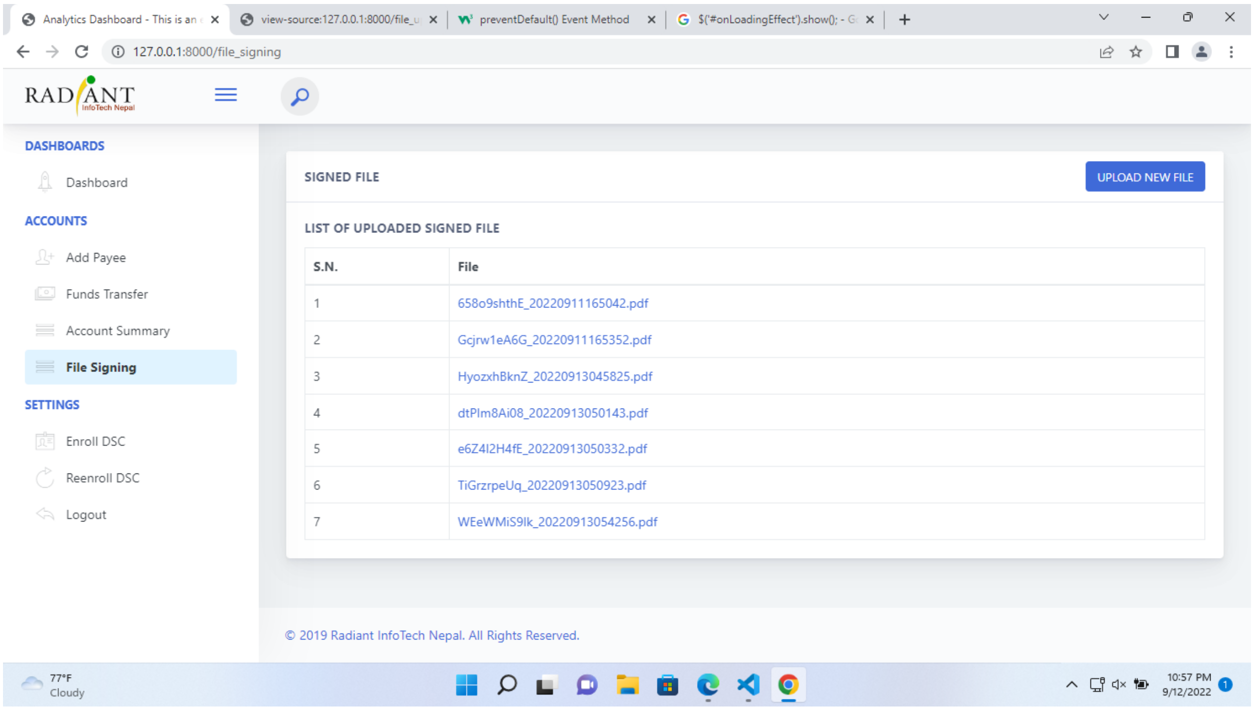
```
Route::get('/file_signing','UploadController@fileSignList')->name('file_sign_list');
```

Above route shows controller name `UploadController and its function name fileSignList`

```
    public function fileSignList()
    {
        $this->_data['uploads'] = Upload::all();
        return view($this->_page.'list',$this->_data);
    }
```

`fileSignList function render list.blade.php`

After it display page like this



RADIANT InfoTech Nepal

DASHBOARDS
- Dashboard

ACCOUNTS
- Add Payee
- Funds Transfer
- Account Summary
- File Signing

SETTINGS
- Enroll DSC
- Reenroll DSC
- Logout

SIGNED FILE

UPLOAD NEW FILE

LIST OF UPLOADED SIGNED FILE

S.N.	File
1	658o9shthE_20220911165042.pdf
2	Gcjrw1eA6G_20220911165352.pdf
3	HyozxhBknZ_20220913045825.pdf
4	dtPlm8Ai08_20220913050143.pdf
5	e6Z4I2H4fE_20220913050332.pdf
6	TiGrzrpeUq_20220913050923.pdf
7	WEeWMiS9lk_20220913054256.pdf

© 2019 Radiant InfoTech Nepal. All Rights Reserved.

3) when click in upload new file button then

url is : http://127.0.0.1:8000/file_upload

Route is :

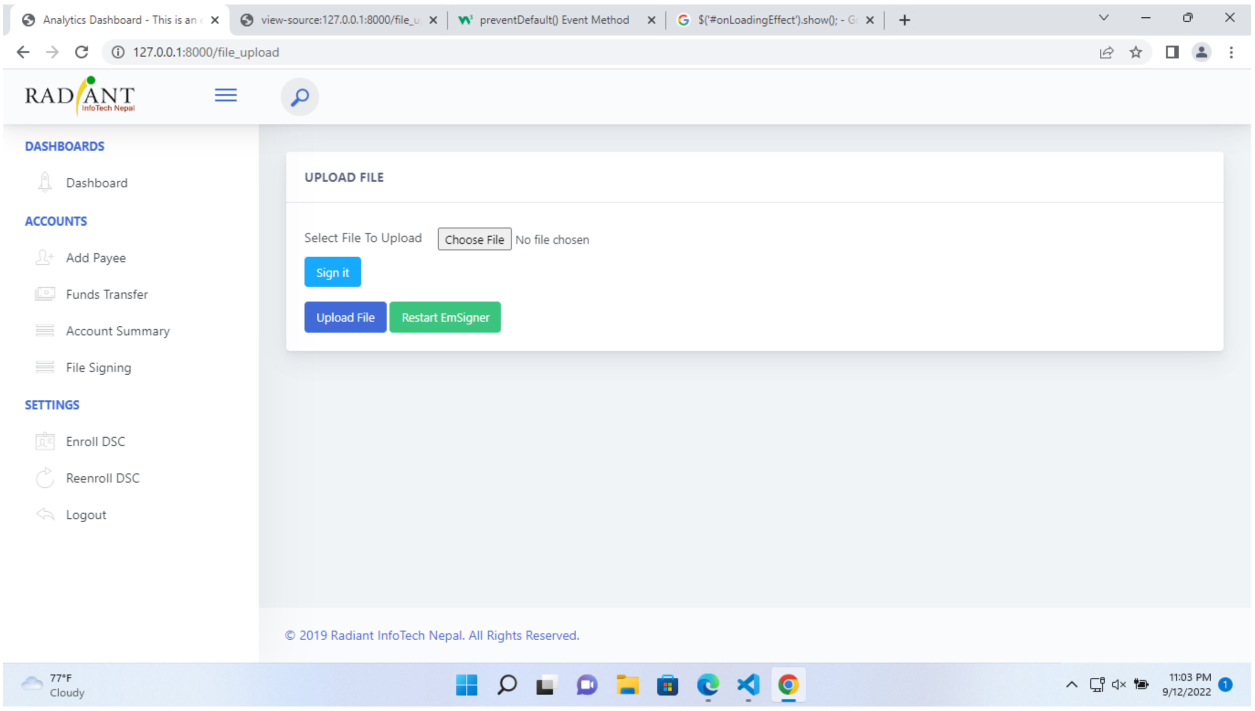
`Route::get('/file_upload','UploadController@fileUpload')->name('file_upload');`

Above route shows controller name **`UploadController and its function name fileUpload`**

```
    public function fileUpload()
    {

        return view($this->_page.'add',$this->_data);
    }
```

The above function render add.blade.php
After this page display like this



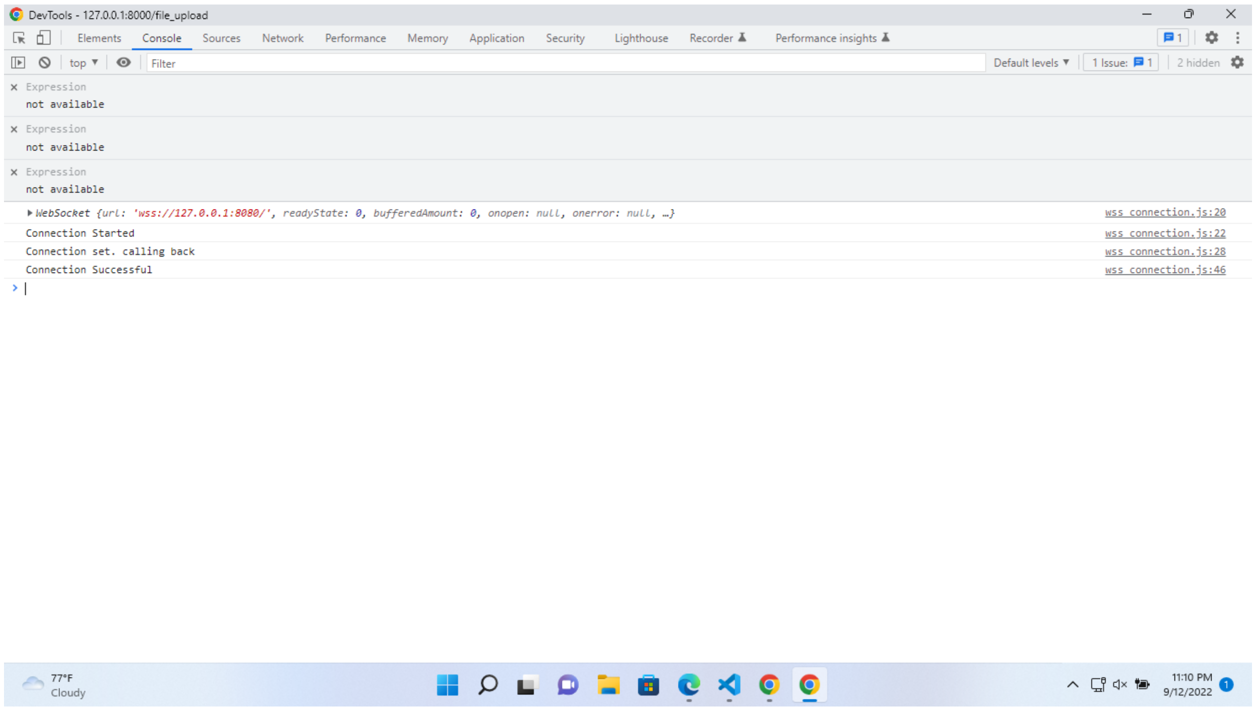
In this page javascript is including using section

```
@section('js_scripts')
<script type="text/javascript">
```

So above code including foot.blade.php

```
</div>
    <script type="text/javascript" src="{{ asset('js/main.js')
}}"></script>
    <script src="{{ asset('js/wss_connection.js') }}"></script>
    <script src="{{ asset('js/sign_data.js') }}"></script>
    @yield('js_scripts')
</body>
</html>
```

So we show here two javascript including `js/wss_connection.js and js/sign_data.js.`
`js/wss_connection.js  javascript function start websocket. You can see in the console like this`



```
Expression
not available
Expression
not available
Expression
not available
▶WebSocket {url: 'wss://127.0.0.1:8080/', readyState: 0, bufferedAmount: 0, onopen: null, onerror: null, …}    wss_connection.js:20
Connection Started                                                                                            wss_connection.js:22
Connection set. calling back                                                                                  wss_connection.js:28
Connection Successful                                                                                         wss_connection.js:46
```

4. In step 3 there is a form to upload pdf.if you select pdf and press sign it. Then javascript click event is occur. Code show below from add.blade.php

```
@section('js_scripts')
 <script type="text/javascript">
   var CSRF_TOKEN = "{{csrf_token()}}";
   var uploadUrl = "{!! route('temp.upload') !!}";
   var baseURL = "{{ url('/') }}";
   $('#sign-doc').click(function (e) {
       e.preventDefault();
       e.stopPropagation();
       if ($('#document').val() == null) {
           alert('Please select the document first!');
           Return;
 }
       $('#onLoadingEffect').show();

       //upload pdf to templocation
       var doc = document.getElementById('document').files[0];
       console.log("start")
       var form_data = new FormData();
       console.log("end")
       form_data.append('_token', CSRF_TOKEN);
       form_data.append("file", doc);
       $.ajax({
           url: uploadUrl,
           method: 'POST',
           data: form_data,
           contentType: false,
           cache: false,
           processData: false,
           beforSend: function () {
             $('#docinfo').text('Please Wait!! Starting Signing process...');
           },
           success: function (data) {
             $('#tempdoc').val(data.filepath);
             console.log(baseURL + data.filepath);
             signPdf(baseURL + data.filepath, pdfSigned, failedToSignPdf);
           },
           error: function (error) {
           }
       })
   });
```

Note:- all pdf signing process belongs to above function

The above function have Two process .

Process A):- when user select file and press sign it then above function submit form using AJAX.

Some code is `console.log("end")`

```
       var form_data = new FormData();
       form_data.append('_token', CSRF_TOKEN);
        form_data.append("file", doc);
        $.ajax({
            url: uploadUrl,
            method: 'POST',
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            beforSend: function () {
                $('#docinfo').text('Please Wait!! Starting Signing 
process...');
            },
            success: function (data) {
                $('#tempdoc').val(data.filepath);
                console.log(baseURL + data.filepath);
                signPdf(baseURL + data.filepath, pdfSigned, 
failedToSignPdf); //(this is second process)
            },
            error: function (error) {

            }
        })
    });
```

its.url is 127.0.0.1:8000/`temp/upload`

`And route is :- Route::post('temp/upload',['as' => 'temp.upload','uses'=>'UploadController@tempUpload']);`

`Note:-second process B is occurred when form is submitted to database and response is passing to javascript function  signPdf(baseURL + data.filepath, pdfSigned, failedToSignPdf);`

Lets move it again in process A. Form is submitted to database through php function public function tempUpload(Request $request) which is in controller UploadController.php .

tempUpload function described below.

```
public function tempUpload(Request $request){
        $file = $request->file('file');
        if(null == $file){
            return response()->json(['status'=>'failed','msg'=>'No files
found on request.','error-code'=>400],400);
        }
        $filename=
str_random(10).'_'.date('YmdHis').'.'.$file->getClientOriginalExtension();
        $tempDir = '/tempUpload/';
        $result = $file->move(public_path().$tempDir,$filename);
        if($result){
            return
response()->json(['status'=>'success','filepath'=>$tempDir.$filename],200)
;

        }else {
            return response()->json(['status'=>'failed','msg'=>'Failed to
save file.','error-code'=>500],500);

        }
    }
```

In process A(above function) do nothing it just upload file and save to database . when form is successfully submitted then response is send back to javascript function signPdf(baseURL + data.filepath, pdfSigned, failedToSignPdf);

Means form is submitted and back to again in success ajax code.where signPdf(baseURL + data.filepath, pdfSigned, failedToSignPdf); function is calling.

PROCESS B :- in process B submitted form response is send as parameter signPdf(baseURL + data.filepath, pdfSigned, failedToSignPdf) . Signpdf function is in included in sign_data.js .

Signpdf function looks like this.

```
function signPdf(inputFileUrl, successCallback, failureCallback) {
    var signParams = "emsigneraction=pdfsign\n" +
        "tbs=" + inputFileUrl +
        "\noutputpath=" +
        "\nsignaction=3\n" +
        "certtype=ALL\n" +
        "expirycheck=true\n" +
        "issuername=\n" +
        "signtype=detached\n" +
        "coordinate=400,100,500,150\n" +
        "pageno=all\n" +
        "reason=test\n" +
        "location=Kathmandu";
    callApplet(signParams, successCallback, failureCallback);
}
```

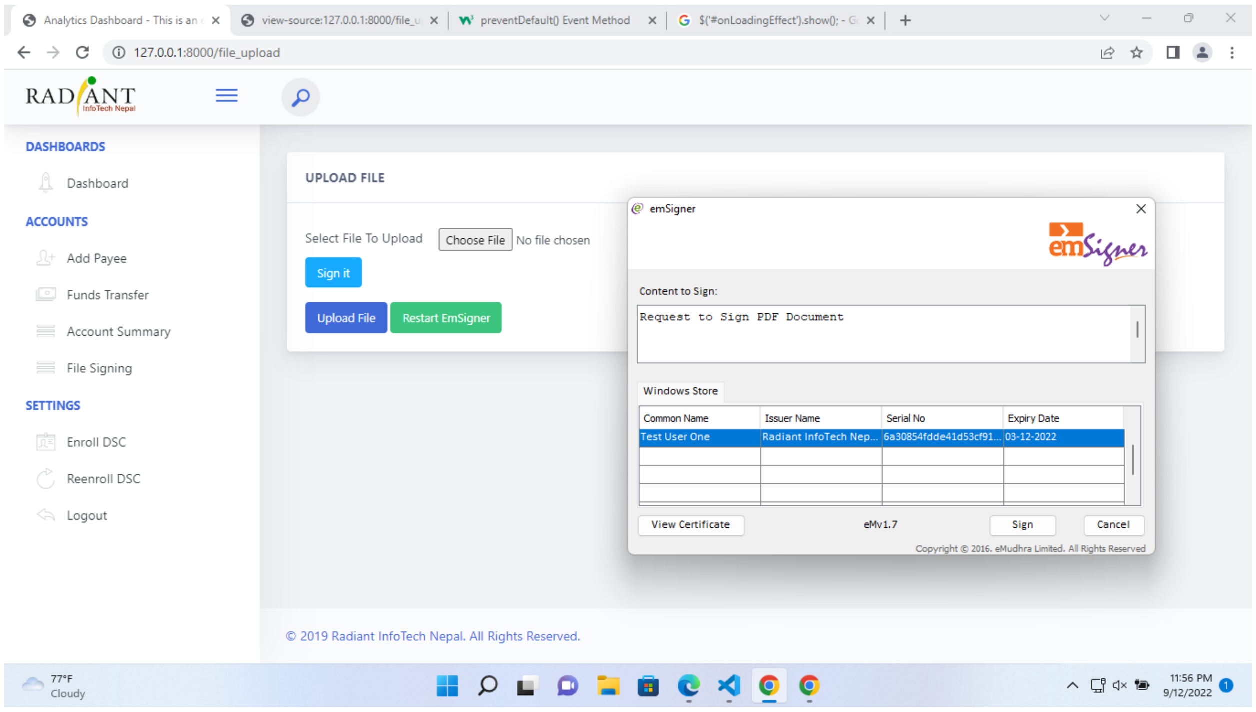
In signPdf , calling of callApplet with some parameter like signparams and successcallback(which is response of pdf upload to database) etc.

Note:- callApplet(signParams, successCallback, failureCallback) function is used to call client application emsigner with some parameter.

callApplet function described below.

```
function callApplet(msgText,successCallBack,randomNo)
{   if (connection == null) {
        alert('Error , Try Again');
    }
    alert(msgText)
    connection.send(msgText);
    connection.onerror = (error) => {
        alert('Please check the server connection : ' + error);
        return failureCallBack(error);
    }
    connection.onmessage = (e) => {
        if (e.data.indexOf("subProtocol") == -1) {
            var respData = e.data;
            return successCallBack(respData);
        }
    }  }
```

Note:- callapplet function main statement is **connection.send(msgText)**
Which is coming from js/wss_connection.js.
Finally **callApplet send request to client emsigner and popup using (connection.send(msgText))**



Content to Sign:

```
Request to Sign PDF Document
```

Common Name	Issuer Name	Serial No	Expiry Date
Test User One	Radiant InfoTech Nep...	6a30854fdde41d53cf91...	03-12-2022

Copyright © 2016. eMudhra Limited. All Rights Reserved

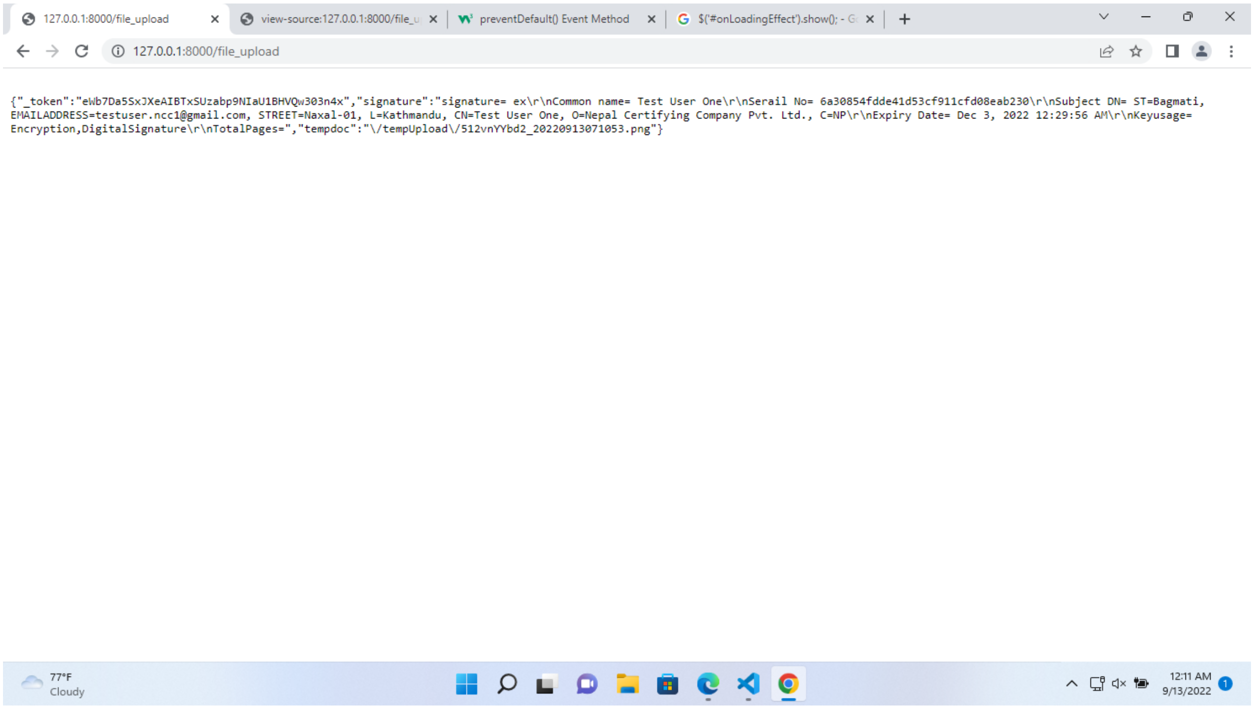
When client select test user one .if no error then it return back with client emsigner data.

```
connection.onmessage = (e) => {
 if (e.data.indexOf("subProtocol") == -1) {
            var respData = e.data;
            return successCallBack(respData);
        }
    }
```

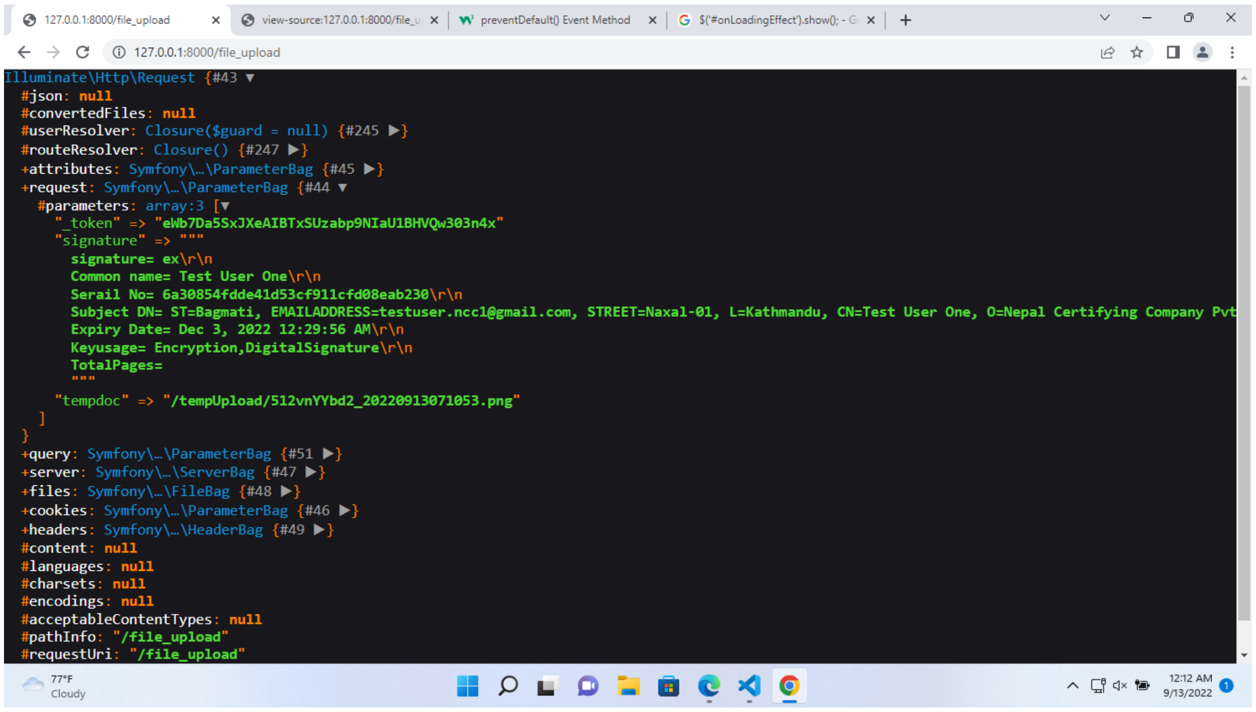
5) when click in upload file button then it submit data to database with clint pdf signature data.

```
public function fileUploadAction(Request $request)
    {
        $doc_path = "signedfile/";
        $filename = str_random(10) . '_' . date('YmdHis') . '.pdf';
        $signature = $this->grabSignatureOnly($request->signature);
       //
Storage::disk('public')->put($doc_path.$filename,base64_decode($signature)
);
        //Storage::disk('public')->put(public_path()
.$request->tempdoc,base64_decode($signature));
        Storage::disk('public')->put($doc_path.$filename,
base64_decode($signature));
        if (isset($request->tempdoc)) {
            File::delete(public_path() .$request->tempdoc);
        }

        $upload = new Upload();
        $upload->file_name = $filename;
        $upload->signature = $signature;
        if ($upload->save()) {
            return redirect()->route('file_sign_list')->with('success',
'File has been successfully uploaded .');
        }
        return redirect()->back()->with('danger', 'File could not be
uploaded .');
    }
```



{"_token":"eWb7Da5SxJXeAIBTxSUzabp9NIaU1BHVQw303n4x","signature":"signature= ex\r\nCommon name= Test User One\r\nSerail No= 6a30854fdde41d53cf911cfd08eab230\r\nSubject DN= ST=Bagmati, EMAILADDRESS=testuser.ncc1@gmail.com, STREET=Naxal-01, L=Kathmandu, CN=Test User One, O=Nepal Certifying Company Pvt. Ltd., C=NP\r\nExpiry Date= Dec 3, 2022 12:29:56 AM\r\nKeyusage= Encryption,DigitalSignature\r\nTotalPages=","tempdoc":"\/tempUpload\/512vnYYbd2_20220913071053.png"}



```
Illuminate\Http\Request {#43 ▼
  #json: null
  #convertedFiles: null
  #userResolver: Closure($guard = null) {#245 ▶}
  #routeResolver: Closure() {#247 ▶}
  +attributes: Symfony\…\ParameterBag {#45 ▶}
  +request: Symfony\…\ParameterBag {#44 ▼
    #parameters: array:3 [▼
      "_token" => "eWb7Da5SxJXeAIBTxSUzabp9NIaU1BHVQw303n4x"
      "signature" => """
        signature= ex\r\n
        Common name= Test User One\r\n
        Serail No= 6a30854fdde41d53cf911cfd08eab230\r\n
        Subject DN= ST=Bagmati, EMAILADDRESS=testuser.ncc1@gmail.com, STREET=Naxal-01, L=Kathmandu, CN=Test User One, O=Nepal Certifying Company Pvt
        Expiry Date= Dec 3, 2022 12:29:56 AM\r\n
        Keyusage= Encryption,DigitalSignature\r\n
        TotalPages=
        """
      "tempdoc" => "/tempUpload/512vnYYbd2_20220913071053.png"
    ]
  }
  +query: Symfony\…\ParameterBag {#51 ▶}
  +server: Symfony\…\ServerBag {#47 ▶}
  +files: Symfony\…\FileBag {#48 ▶}
  +cookies: Symfony\…\ParameterBag {#46 ▶}
  +headers: Symfony\…\HeaderBag {#49 ▶}
  #content: null
  #languages: null
  #charsets: null
  #encodings: null
  #acceptableContentTypes: null
  #pathInfo: "/file_upload"
  #requestUri: "/file_upload"
```

Conclusion:-

User select pdf and press sign it. Using ajax, upload file and save to database.if it successfully saved then return response back to ajax and

send response to `signPdf(baseURL + data.filepath, pdfSigned, failedToSignPdf);`

signPdf call callApplet function which send request to client emsigner and popup it. When client select test user one , if success (no error) then it's pdf signature return again in add.blade.php form where hidden input field exists. After click upload file then it send file detail with signature data and store in database.

Thanku!!!