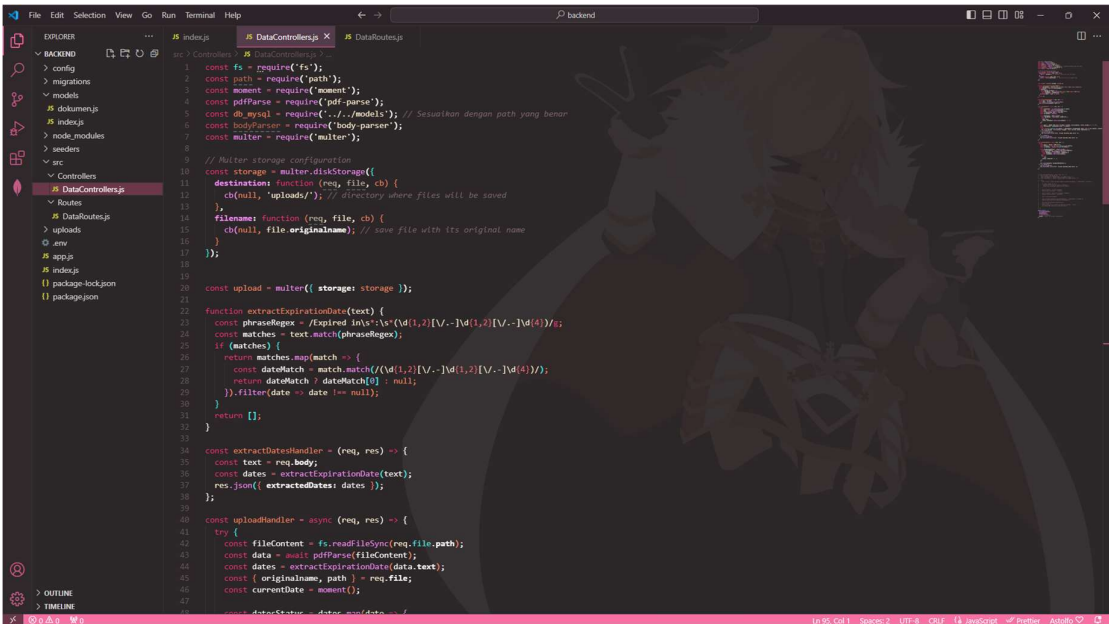


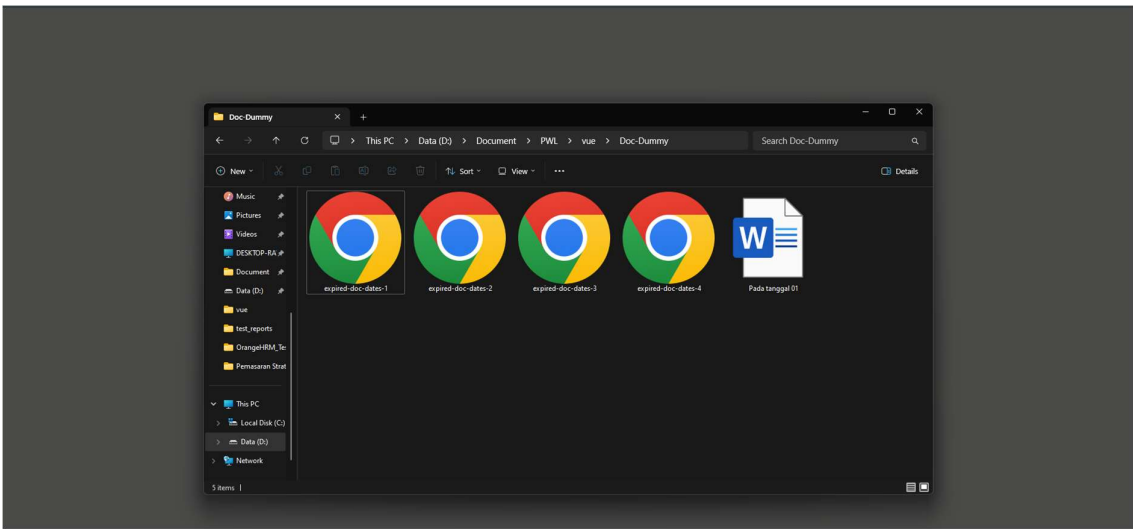
[illegible]

Script code untuk back-end menggunakan framework Express.JS



```
1 const fs = require('fs');
2 const path = require('path');
3 const moment = require('moment');
4 const pdfParse = require('pdf-parse');
5 const db_mysql = require('../models'); // Sesuaikan dengan path yang benar
6 const bodyParser = require('body-parser');
7 const multer = require('multer');
8
9 // Multer storage configuration
10 const storage = multer.diskStorage({
11   destination: function (req, file, cb) {
12     cb(null, 'uploads/'); // directory where files will be saved
13   },
14   filename: function (req, file, cb) {
15     cb(null, file.originalname); // save file with its original name
16   }
17 });
18
19 const upload = multer({ storage: storage });
20
21
22 function extractExpirationDate(text) {
23   const phraseRegex = /Expired in(\d{1,2}[V-]\d{1,2}[V-]\d{4})/g;
24   const matches = text.match(phraseRegex);
25   if (matches) {
26     return matches.map(match => {
27       const dateMatch = match.match(/(\d{1,2}[V-]\d{1,2}[V-]\d{4})/);
28       return dateMatch ? dateMatch[0] : null;
29     }).filter(date => date !== null);
30   }
31   return [];
32 }
33
34 const extractDatesHandler = (req, res) => {
35   const text = req.body;
36   const dates = extractExpirationDate(text);
37   res.json({ extractedDates: dates });
38 };
39
40 const uploadHandler = async (req, res) => {
41   try {
42     const fileContent = fs.readFileSync(req.file.path);
43     const data = await pdfParse(fileContent);
44     const dates = extractExpirationDate(data.text);
45     const { originalname, path } = req.file;
46     const currentDate = moment();
47     const dataObj = {
48       originalname: originalname,
49       path: path,
50       dates: dates,
51       currentDate: currentDate.toISOString()
52     };
53     db_mysql.insert('documents', dataObj, (err, result) => {
54       if (err) {
55         res.status(500).json({ message: 'Error saving document' });
56       } else {
57         res.status(201).json({ message: 'Document saved successfully' });
58       }
59     });
60   } catch (error) {
61     res.status(500).json({ message: 'Error uploading document' });
62   }
63 }
```

Data Dummy untuk menentukan tanggal expired pada dokumen



Tampilan table files pada database yang menggunakan MySQL

data\_arsip

New

dokumens

files

sequelizemeta

information\_schema

mysql

performance\_schema

phpmyadmin

pwl\_chat\_db

test

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

FileID

FileName

FilePath

ExtractedDates

Status

☐

Edit

Copy

Delete

76

expired-doc-dates-1.pdf

uploads\expired-doc-dates-1.pdf

["31-12-2024"]

0

☐

Edit

Copy

Delete

77

expired-doc-dates-2.pdf

uploads\expired-doc-dates-2.pdf

["24-11-2024"]

0

☐

Edit

Copy

Delete

78

expired-doc-dates-3.pdf

uploads\expired-doc-dates-3.pdf

["28-02-2024"]

1

☐

Edit

Copy

Delete

79

expired-doc-dates-4.pdf

uploads\expired-doc-dates-4.pdf

["25-11-2024"]

0

Hasil tampilan web beserta data yang sudah diupload dan di ekstrak tanggalnya, lalu memunculkan informasi apakah dokumen tersebut sudah kadaluwarsa atau belum

Upload Dokumen

Pilih Dokumen

Choose File

expired-doc-dates-4.pdf

Upload

File ID	File Name	File Path	Extracted Dates	Status
76	expired-doc-dates-1.pdf	uploads\expired-doc-dates-1.pdf	["31-12-2024"]	Not Expired
77	expired-doc-dates-2.pdf	uploads\expired-doc-dates-2.pdf	["24-11-2024"]	Not Expired
78	expired-doc-dates-3.pdf	uploads\expired-doc-dates-3.pdf	["28-02-2024"]	Expired
79	expired-doc-dates-4.pdf	uploads\expired-doc-dates-4.pdf	["25-11-2024"]	Not Expired