

Claude Computer Use Demo

⚠ Security Alert: Never provide access to sensitive accounts or data, as malicious web content can hijack Claude's behavior

Chat HTTP Exchange Logs



INVOICE DATA EXTRACTION TASK

ENVIRONMENT:

- Web browser (Firefox) with invoice management system
- Text editor (gedit) with example.csv file open

OBJECTIVE: Extract and compile invoice information from multiple pages into a structured CSV format.

MULTI-PAGE DATA EXTRACTION PROCESS:

PHASE 1 - PAGE 1 PROCESSING: For each invoice record on page 1:

INPUT DATA (from web interface):

- Invoice ID (unique identifier)
- Due Date (DD-MM-YYYY format)

EXTRACTION STEPS:

1. Click invoice download button for each ID
2. Open the downloaded invoice document
3. Extract the following fields: • Invoice Number • Invoice Date (DD-MM-YYYY format) • Company Name • Total Amount Due
4. Enter data into CSV file (one line per invoice)

PHASE 2 - PAGE NAVIGATION: After completing all invoices on page 1:

1. Navigate to page 2 of the invoice system
2. Repeat the same extraction process for all invoices on page 2
3. Continue adding data to the same CSV file

PHASE 3 - FINAL PAGE PROCESSING: After completing page 2:

1. Navigate to page 3 of the invoice system
2. Repeat the same extraction process for all invoices on page 3
3. Continue adding data to the same CSV file

PROCESSING ORDER: Page 1 → Complete all invoices → Page 2 → Complete all invoices → Page 3 → Complete all invoices

OUTPUT FORMAT (CSV): ID,DueDate,InvoiceNo,InvoiceDate,CompanyName,TotalDue

SAMPLE DATA: 5jef1y8yx4t8yupbpo3fzg,25-02-2019,10021,13-02-2019,Sit Amet Corp.,1234.40
g11rzk7loegnitbclwnh8k,06-04-2019,223543,30-01-2019,Aenean LLC,976.00

IMPORTANT NOTES:

- Process pages sequentially (1, 2, then 3)
- All extracted data goes into the same CSV file
- Maintain consistent formatting across all pages
- Ensure complete processing of each page before moving to the next
- Final CSV should contain all invoices from pages 1, 2, and 3

Enter one complete record per line in the CSV file, continuing to add new lines as you process each subsequent page.

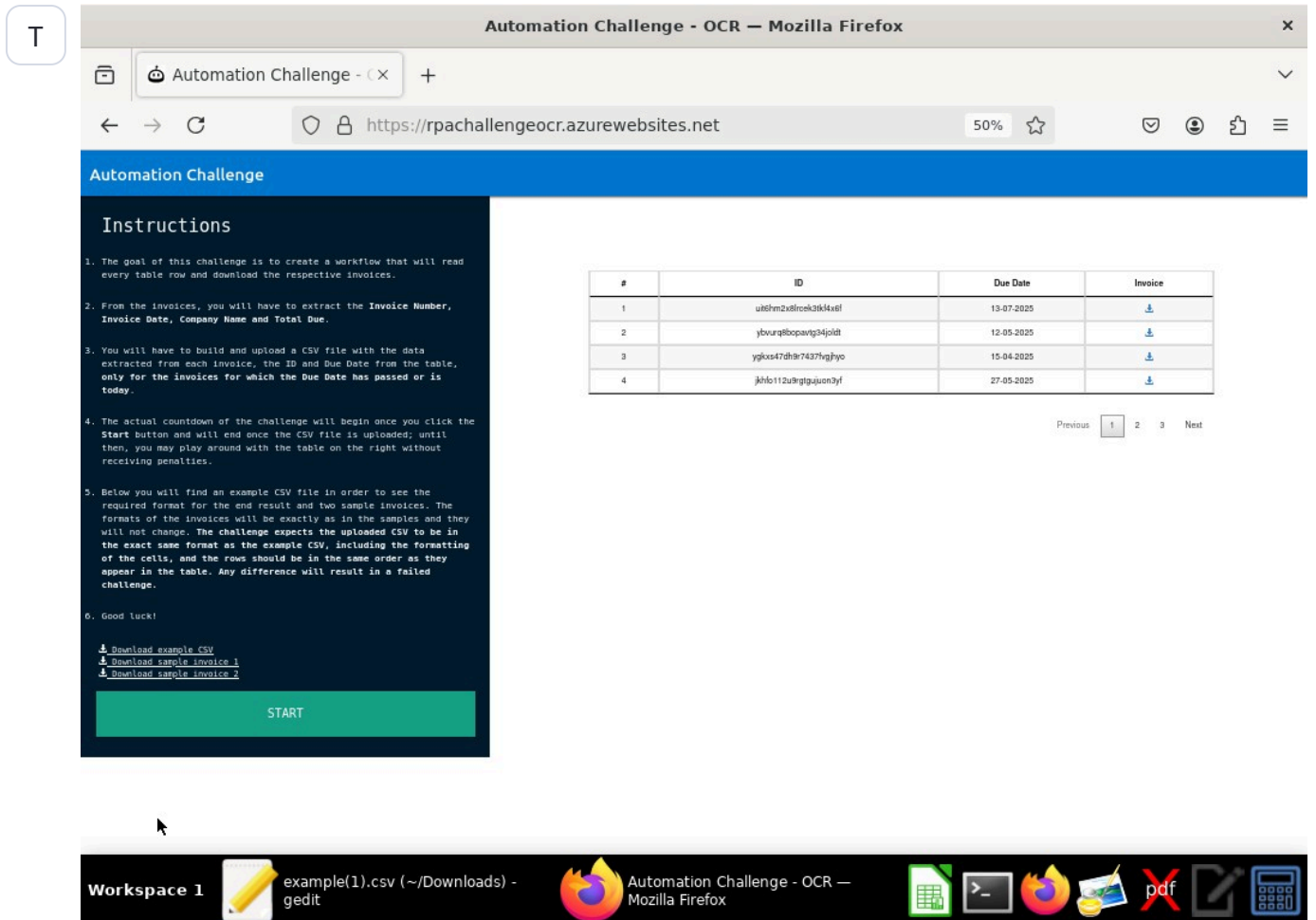



I'll help you extract invoice data from multiple pages and compile it into a CSV format. Let me start by taking a screenshot to see the current state of the system.



Tool Use: computer

Input: {'action': 'screenshot'}



 I can see the invoice management system with a table showing invoices. There's an example.csv file open in the taskbar. Let me first click the START button to begin the challenge, and then examine the CSV file structure.

Tool Use: computer
Input: {'action': 'left_click', 'coordinate': [170, 589]}

T

Automation Challenge - OCR — Mozilla Firefox

Automation Challenge - (X) +

← → ↻

🔒 🌐 🔗 https://rpachallengeocr.azurewebsites.net 50% ☆

🔍 👤 📄 ☰

Automation Challenge

Instructions

- The goal of this challenge is to create a workflow that will read every table row and download the respective invoices.
- From the invoices, you will have to extract the **Invoice Number**, **Invoice Date**, **Company Name** and **Total Due**.
- You will have to build and upload a CSV file with the data extracted from each invoice, the ID and Due Date from the table, only for the invoices for which the Due Date has passed or is today.
- The actual countdown of the challenge will begin once you click the **Start** button and will end once the CSV file is uploaded; until then, you may play around with the table on the right without receiving penalties.
- Below you will find an example CSV file in order to see the required format for the end result and two sample invoices. The formats of the invoices will be exactly as in the samples and they will not change. The challenge expects the uploaded CSV to be in the exact same format as the example CSV, including the formatting of the cells, and the rows should be in the same order as they appear in the table. Any difference will result in a failed challenge.
- Good luck!

[Download example CSV](#)
[Download sample invoice 1](#)
[Download sample invoice 2](#)

In Progress

#	ID	Due Date	Invoice
1	jpmorlp3en3demy61zb	05-06-2025	📄
2	j83wed8wlc3pvdrt1vdla03h	22-04-2025	📄
3	5gha45gx7m8yoox3unbp	24-04-2025	📄
4	8bihqdbwy26i3heuo07	19-05-2025	📄

Previous 1 2 3 Next

Workspace 1

example(1).csv (~/.Downloads) - gedit

Automation Challenge - OCR — Mozilla Firefox



Great! Now I can see the challenge is in progress with the invoice table displayed on Page 1. I can see:

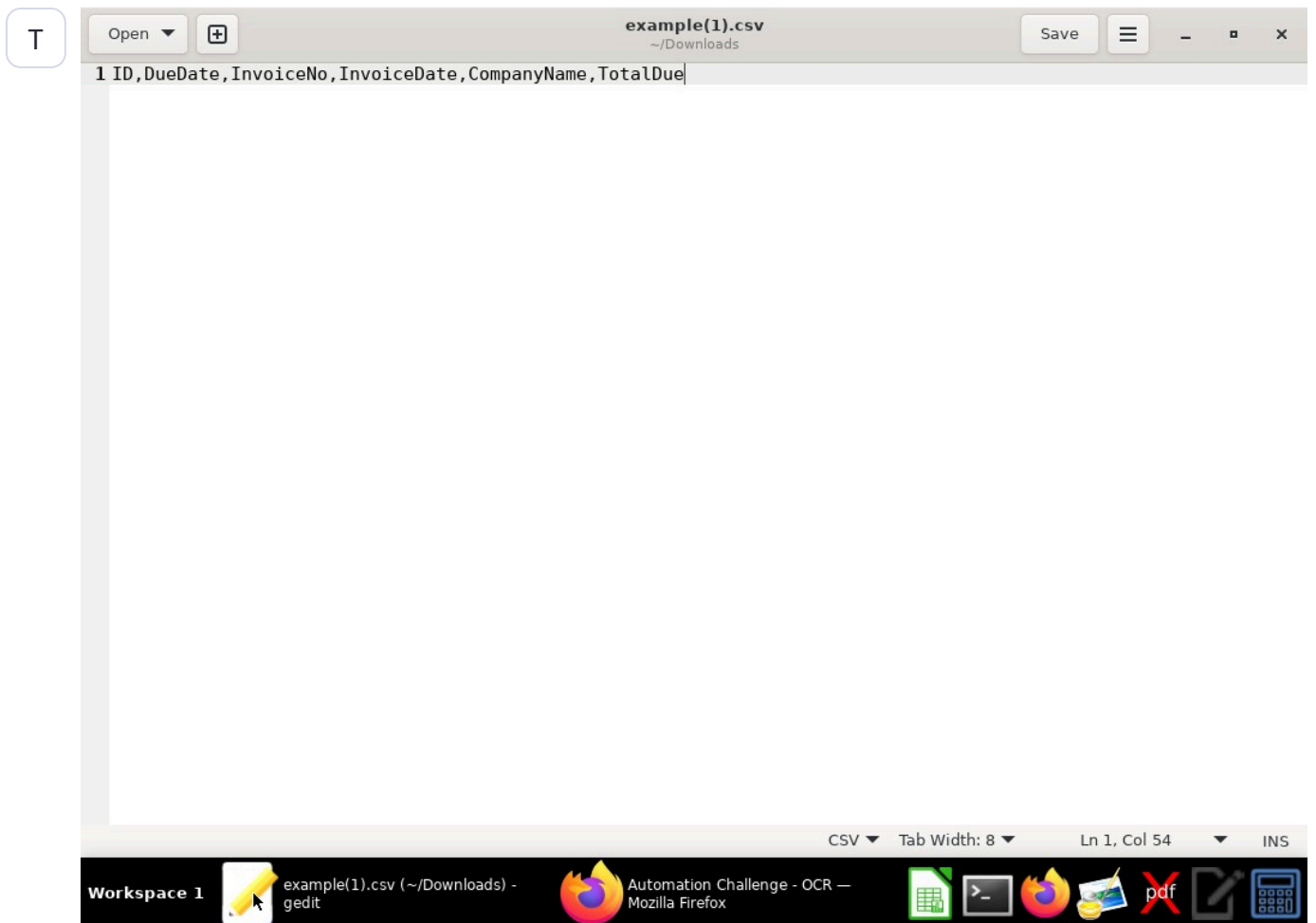
- 4 invoices on this page
- Each row has an ID, Due Date, and an Invoice download button
- Page navigation shows we're on page 1 with pages 2 and 3 available

Let me first check the example.csv file to understand the structure by clicking on it in the taskbar.



Tool Use: computer

Input: `{'action': 'left_click', 'coordinate': [144, 738]}`



Perfect! I can see the CSV file structure. It has the header:

```
ID, DueDate, InvoiceNo, InvoiceDate, CompanyName, TotalDue
```

Now I need to go back to Firefox to start processing the invoices. Let me click on the Firefox icon in the taskbar.



Tool Use: computer

```
Input: {'action': 'left_click', 'coordinate': [496, 738]}
```

T

Automation Challenge - OCR — Mozilla Firefox

Automation Challenge - (x) +

[←](#) [→](#) [↻](#)

[🔒](#) [🌐](#) [🔗](#) [📄](#) [🔍](#)

[🔗](#) [📄](#) [🔍](#)

[https://rpachallengeocr.azurewebsites.net](#)

50% [🌟](#)

Automation Challenge

Instructions

1. The goal of this challenge is to create a workflow that will read every table row and download the respective invoices.
2. From the invoices, you will have to extract the Invoice Number, Invoice Date, Company Name and Total Due.
3. You will have to build and upload a CSV file with the data extracted from each invoice, the ID and Due Date from the table, only for the invoices for which the Due Date has passed or is today.
4. The actual countdown of the challenge will begin once you click the Start button and will end once the CSV file is uploaded; until then, you may play around with the table on the right without receiving penalties.
5. Below you will find an example CSV file in order to see the required format for the end result and two sample invoices. The formats of the invoices will be exactly as in the samples and they will not change. The challenge expects the uploaded CSV to be in the exact same format as the example CSV, including the formatting of the cells, and the rows should be in the same order as they appear in the table. Any difference will result in a failed challenge.
6. Good luck!

[📄 Download example CSV](#)
[📄 Download sample invoice 1](#)
[📄 Download sample invoice 2](#)

In Progress

#	ID	Due Date	Invoice
1	jpmorlp3en3demy61zb	05-06-2025	📄
2	j8wdb9wlc3pvdrt1vda03h	22-04-2025	📄
3	5gha45gx7mtyqozkrRlx	24-04-2025	📄
4	8biqz8wdy26i0tweu07	19-05-2025	📄

Previous 1 2 3 Next

Workspace 1

example(1).csv (~/.Downloads) - gedit

Automation Challenge - OCR — Mozilla Firefox

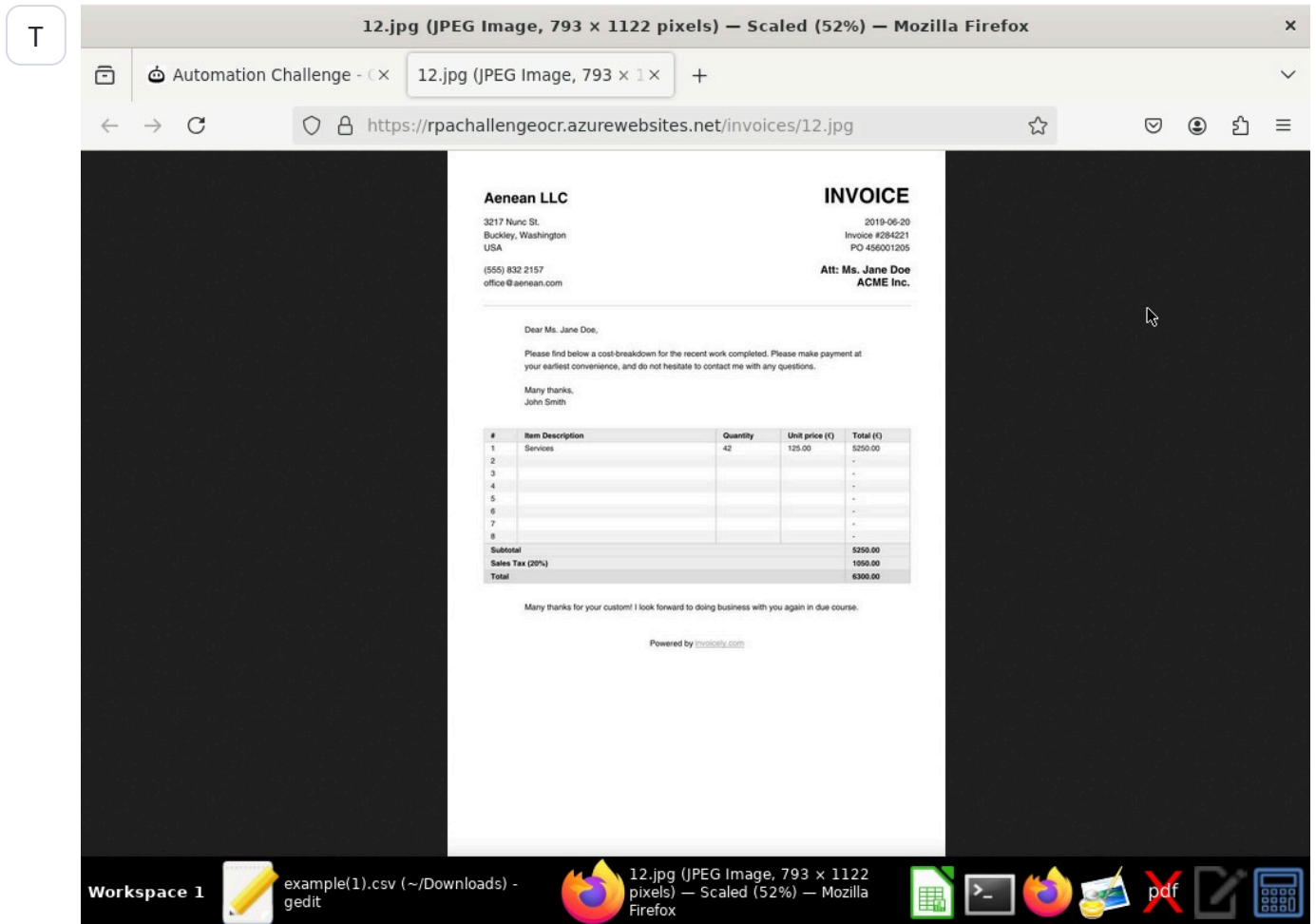
Great! I'm back to the Firefox window showing Page 1 with the invoice data. Now I'll start processing each invoice. Let me begin with the first invoice by clicking on its download button.

From the table, I can see:

- Row 1: ID = jpmme2dnxz8amvjr1st, Due Date = 05-04-2025
- Row 2: ID = phwxBaOoqmrjVztGbeh, Due Date = 22-04-2025
- Row 3: ID = 5ghe3mFpv7mtyqozkrRlx, Due Date = 24-04-2025
- Row 4: ID = aohqz8wdyztbn5RwaO7, Due Date = 16-05-2025

Let me click on the first invoice download button:

Tool Use: computer
Input: `{'action': 'left_click', 'coordinate': [887, 253]}`



Excellent! The first invoice has opened as an image. I can extract the following information:

Invoice 1 Details:

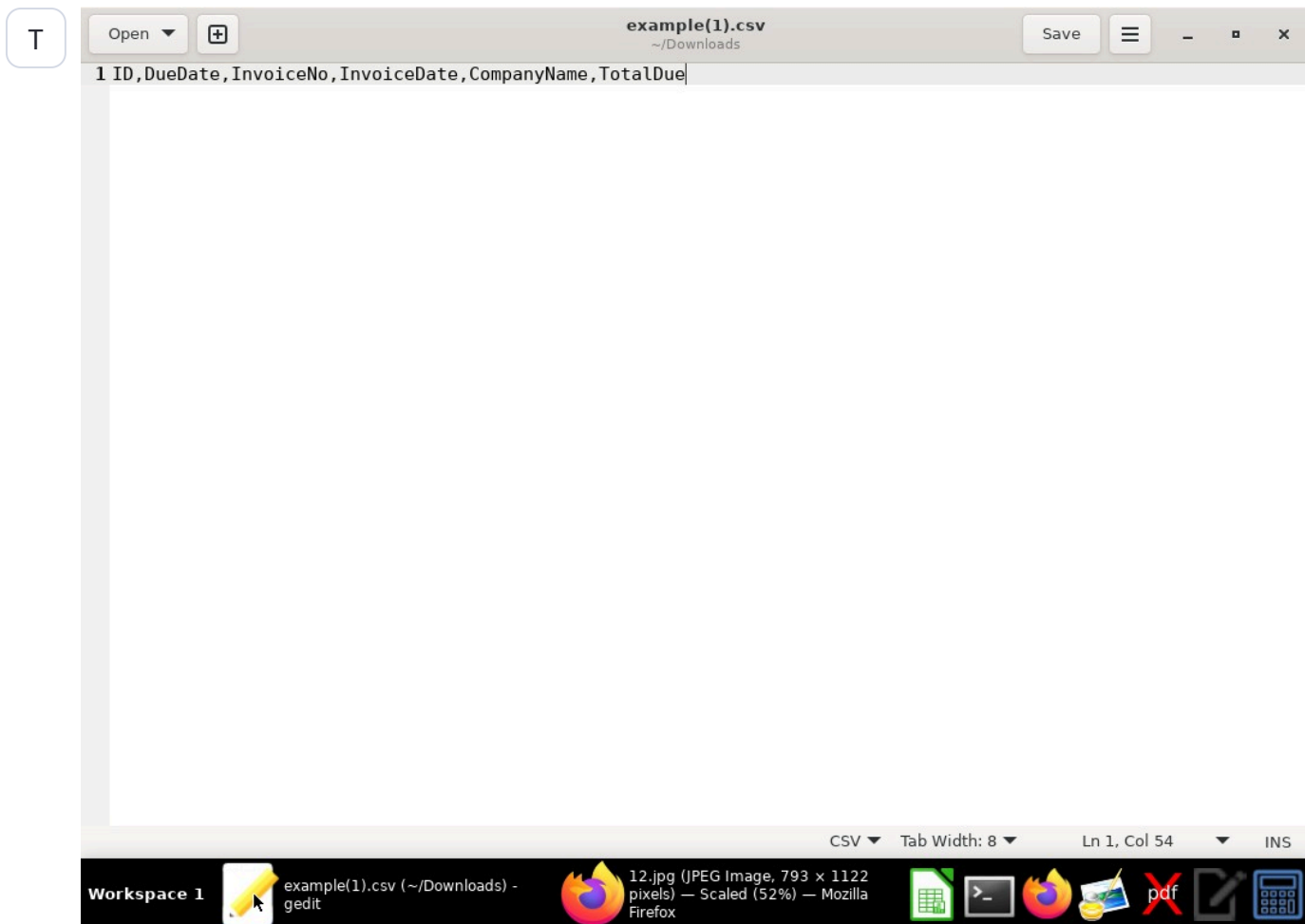
- Company Name: Aenean LLC
- Invoice Number: #984223
- Invoice Date: 2018-06-20
- Total Amount: \$280.00
- ID: jpmme2dnxz8amvjr1st
- Due Date: 05-04-2025


Now let me go back to the CSV file to add this data. Let me click on the CSV file in the taskbar:



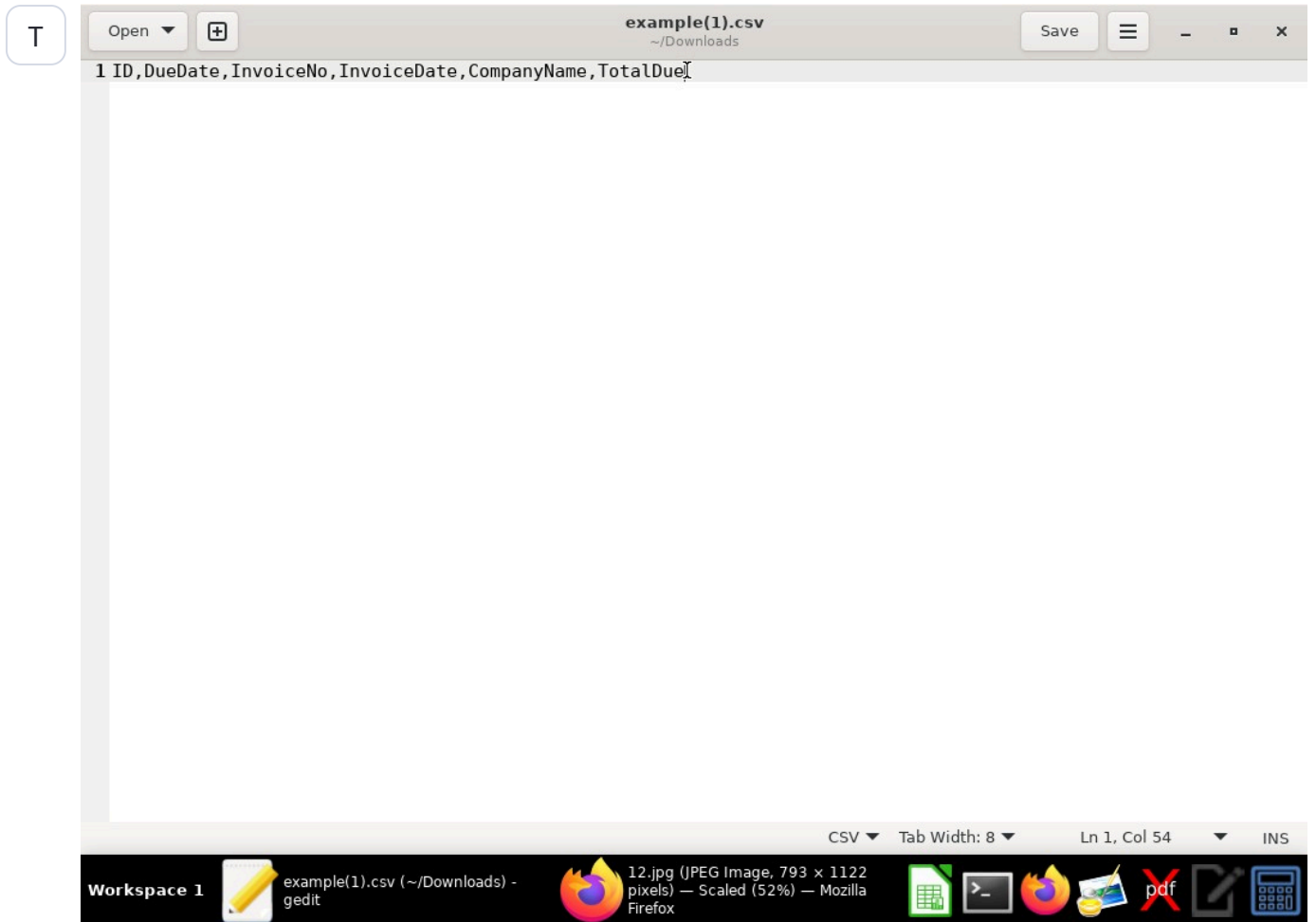
Tool Use: computer


Input: `{'action': 'left_click', 'coordinate': [144, 738]}`




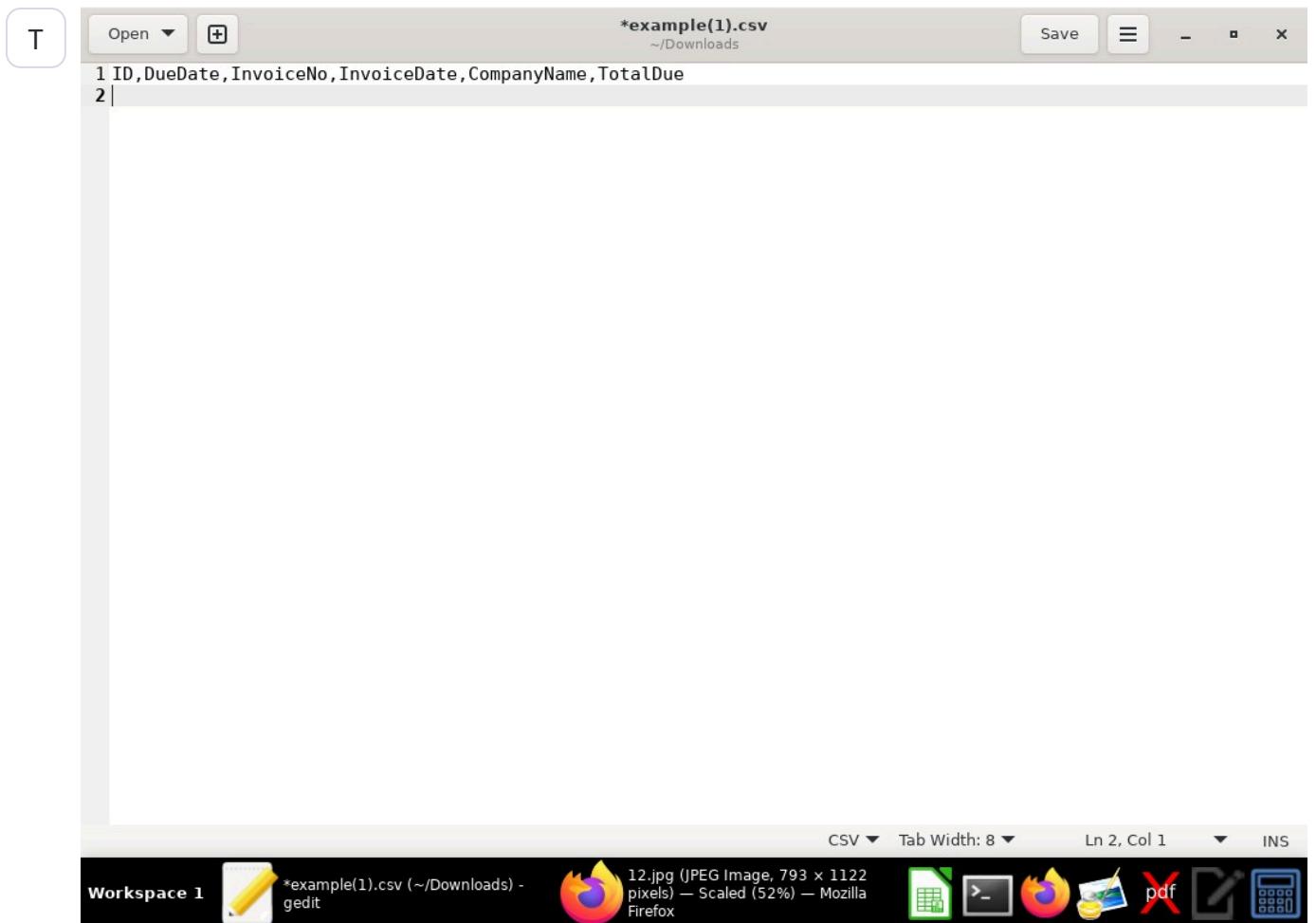
 Now I need to add the first invoice data to the CSV file. I'll click at the end of the header line and add a new line. Let me click after the header line:

 Tool Use: computer
Input: {'action': 'left_click', 'coordinate': [505, 55]}



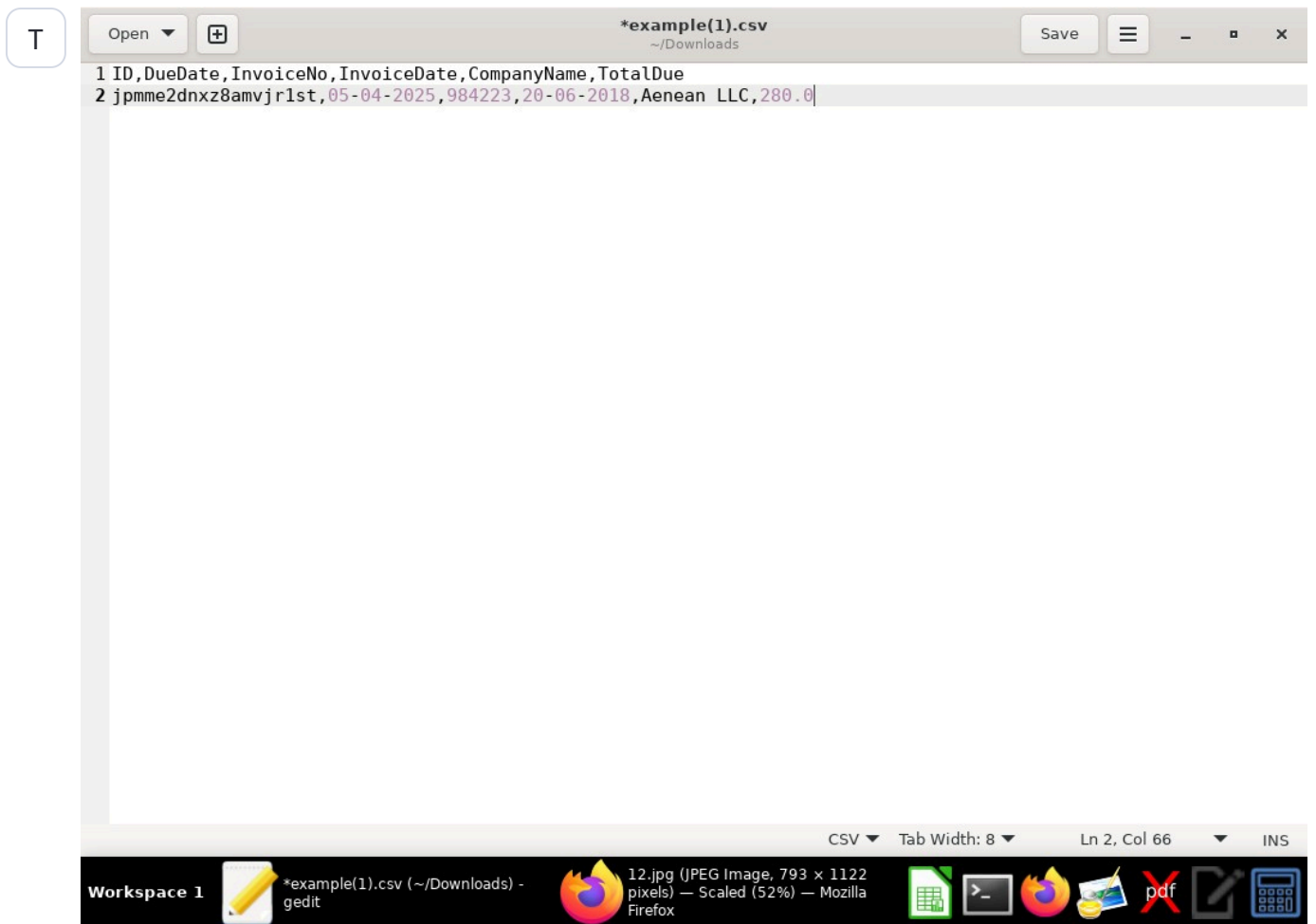
 Good, the cursor is positioned at the end of the header line. Now I'll press Enter and add the first invoice data:

 Tool Use: computer
Input: {'action': 'key', 'text': 'Return'}



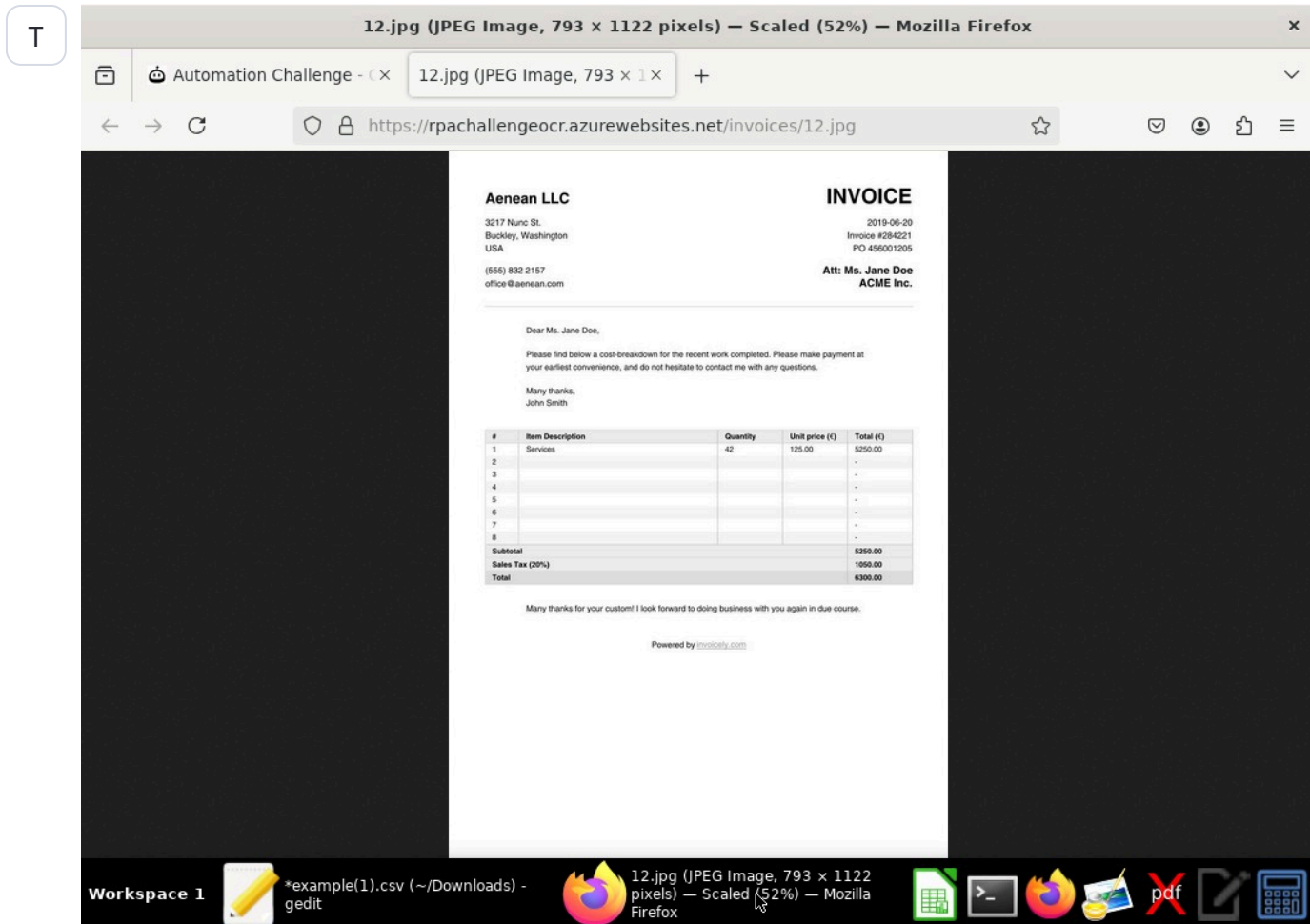
Perfect! I'm now on line 2. I need to convert the invoice date format from 2018-06-20 to DD-MM-YYYY format (20-06-2018) to match the required format. Now I'll type the first invoice data:

Tool Use: computer
Input: {'action': 'type', 'text': 'jpmme2dnxz8amvjr1st,05-04-2025,984223,20



Great! The first invoice data has been added to the CSV. Now I need to go back to Firefox to process the second invoice. Let me click on the Firefox browser:

Tool Use: computer
Input: {'action': 'left_click', 'coordinate': [560, 738]}



I need to go back to the main page with the invoice list. Let me click on the first tab (Automation Challenge - OCR):

Tool Use: computer
Input: {'action': 'left_click', 'coordinate': [155, 58]}

T
Automation Challenge - OCR — Mozilla Firefox
✕

AutomationChallenge - x 12.jpg (JPEG Image, 793 x 1 x +)
✕

← → ↻
🔒 https://rpachallengeocr.azurewebsites.net 50% ☆
🔍 📄 📁 ☰

Automation Challenge

Instructions

- The goal of this challenge is to create a workflow that will read every table row and download the respective invoices.
- From the invoices, you will have to extract the **Invoice Number**, **Invoice Date**, **Company Name** and **Total Due**.
- You will have to build and upload a CSV file with the data extracted from each invoice, the ID and Due Date from the table, only for the invoices for which the Due Date has passed or is today.
- The actual countdown of the challenge will begin once you click the **Start** button and will end once the CSV file is uploaded; until then, you may play around with the table on the right without receiving penalties.
- Below you will find an example CSV file in order to see the required format for the end result and two sample invoices. The formats of the invoices will be exactly as in the samples and they will not change. The challenge expects the uploaded CSV to be in the exact same format as the example CSV, including the formatting of the cells, and the rows should be in the same order as they appear in the table. Any difference will result in a failed challenge.
- Good luck!

[Download example CSV](#)
[Download sample invoice 1](#)
[Download sample invoice 2](#)

In Progress

#	ID	Due Date	Invoice
1	jpmorpr3en3demy61zb	05-06-2025	Download
2	j83wed8wlc3pvdrt1vdlao3h	22-04-2025	Download
3	5gha45gx7m8yoon3unbp	24-04-2025	Download
4	8biqdbwly26i3heulo7	19-05-2025	Download

Previous 1 2 3 Next

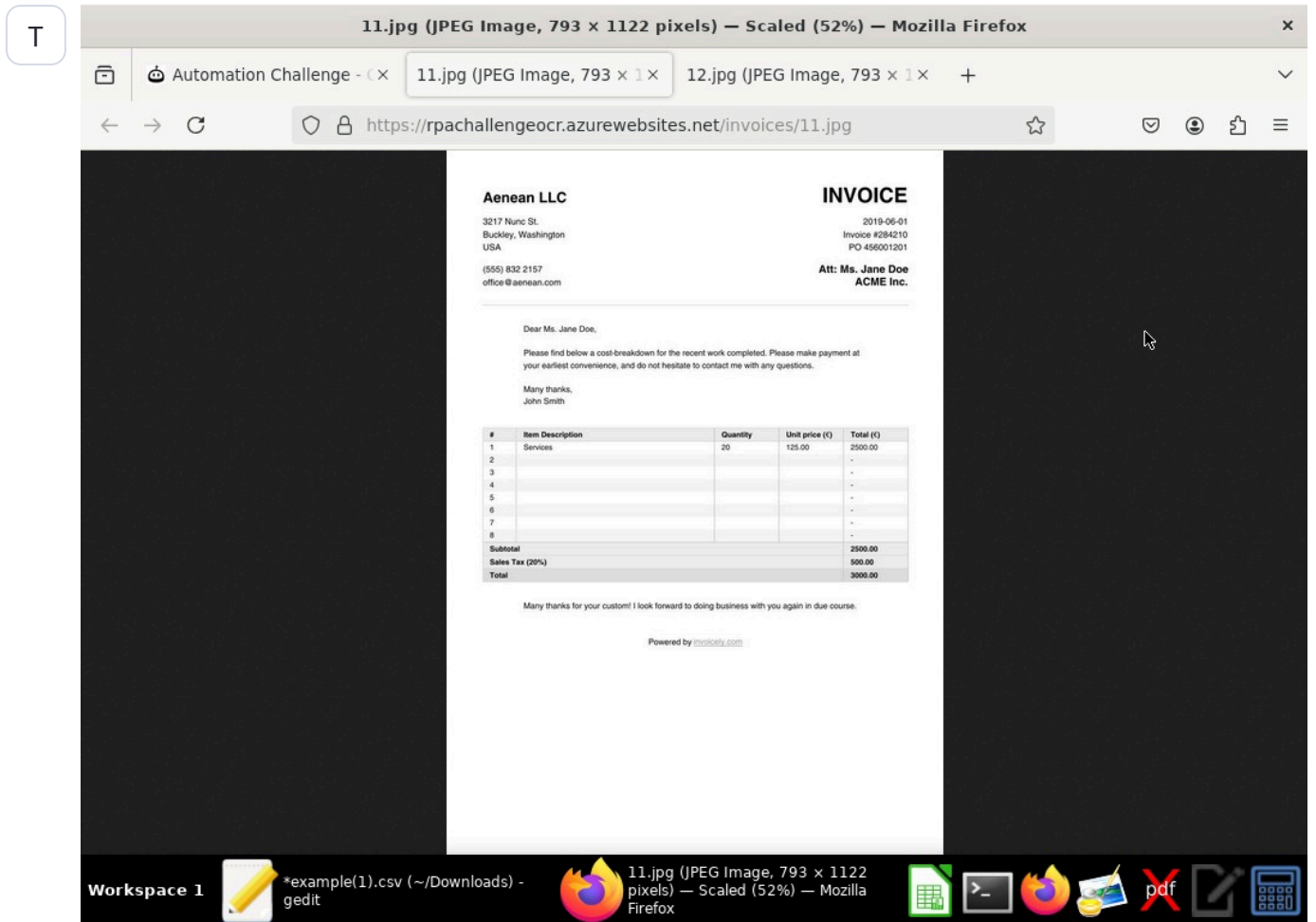
Workspace 1

*example(1).csv (~/.Downloads) - gedit

Automation Challenge - OCR — Mozilla Firefox

Perfect! I'm back to the main invoice list. Now I need to process the second invoice. Let me click on the download button for the second row (phwxBaOoqmrjVztGbeh):

Tool Use: computer
 Input: {'action': 'left_click', 'coordinate': [887, 273]}



Great! The second invoice has opened. I can extract the following information:

Invoice 2 Details:

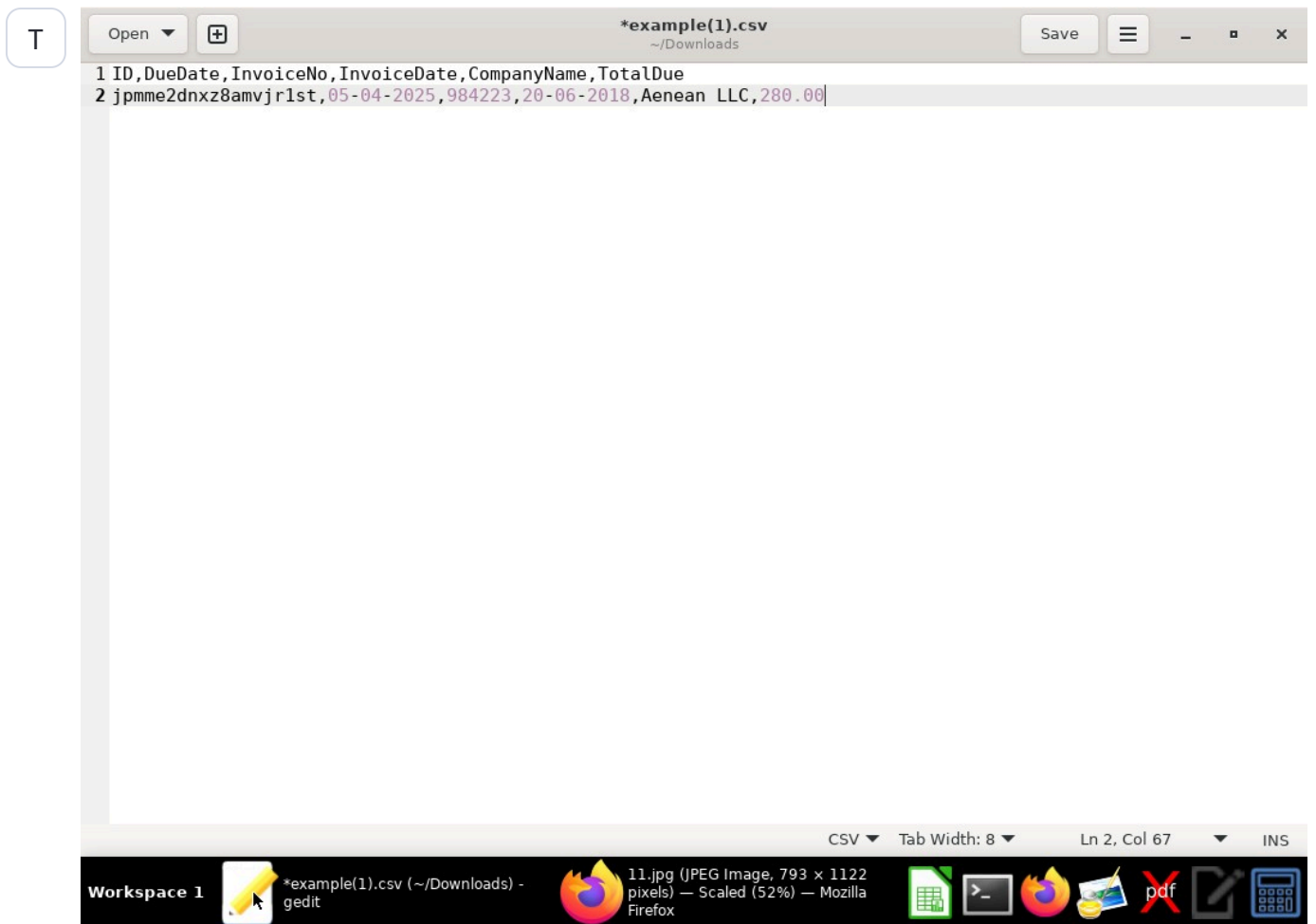
- Company Name: Aenean LLC
- Invoice Number: #983470
- Invoice Date: 2018-06-01
- Total Amount: \$3000.00
- ID: phwxBaOoqmrjVztGbeh
- Due Date: 22-04-2025

Now let me go back to the CSV file to add this data:



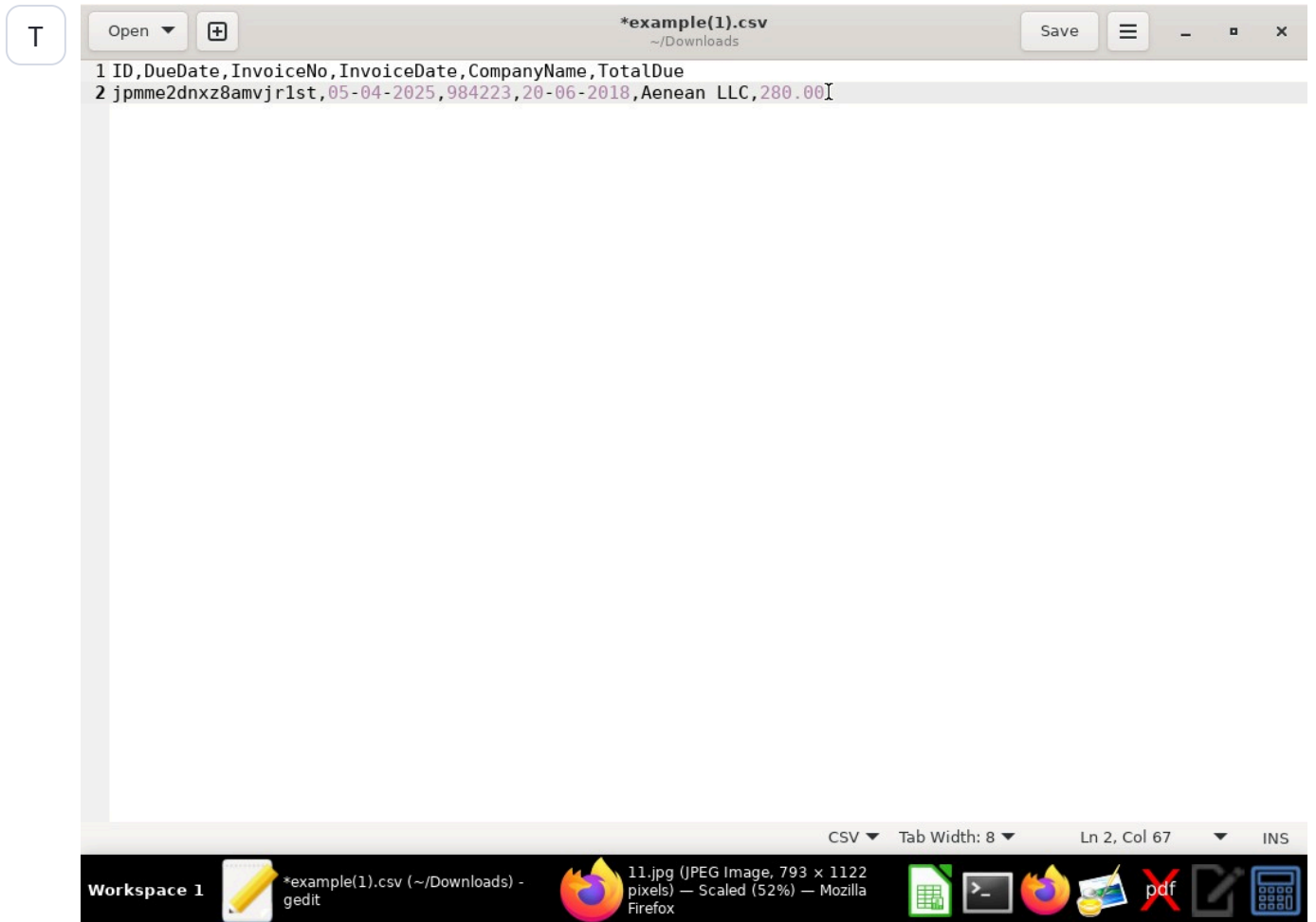
Tool Use: computer


Input: {'action': 'left_click', 'coordinate': [144, 738]}




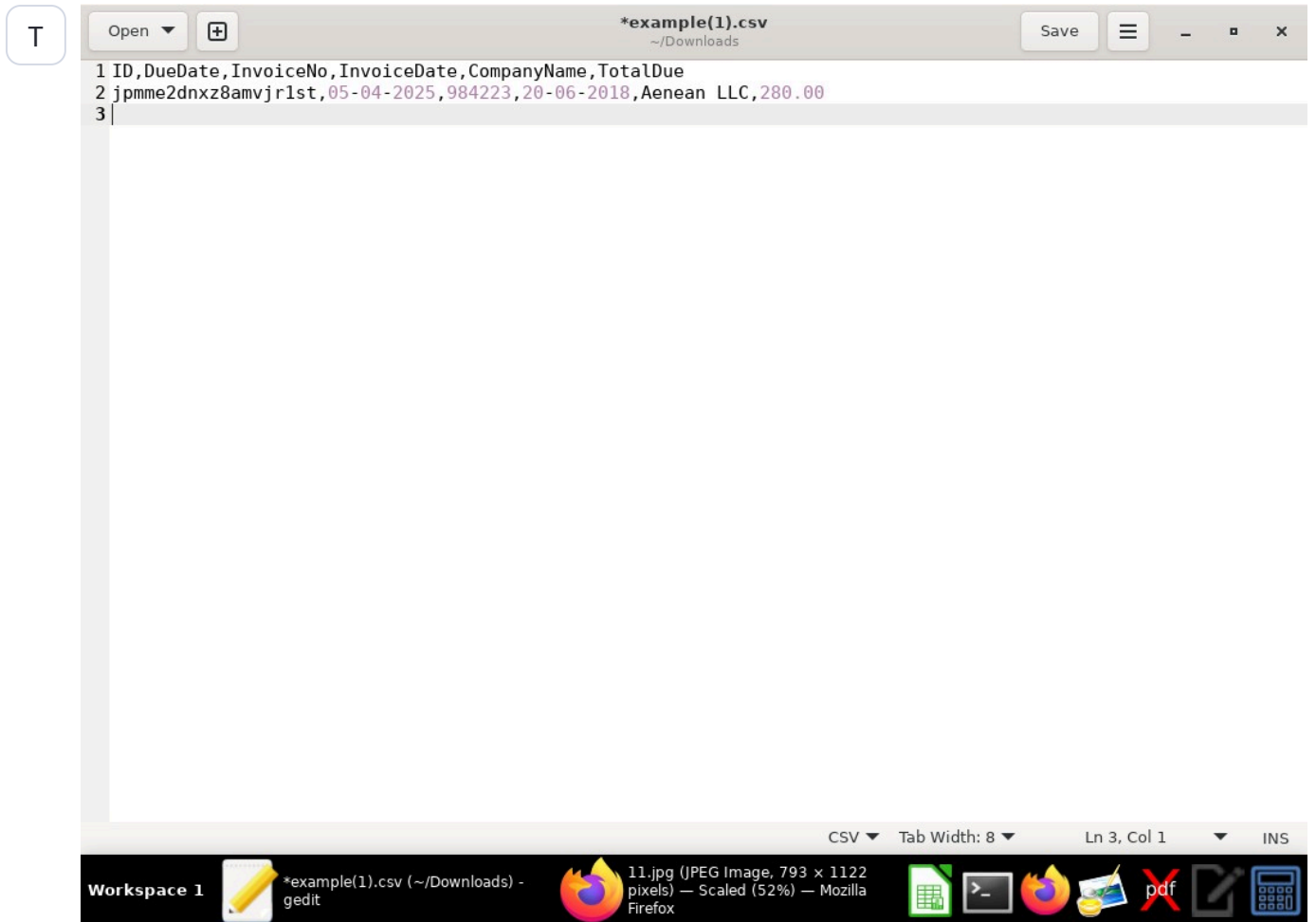
I need to position the cursor at the end of line 2 and add a new line. Let me click at the end of the first invoice entry:


Tool Use: computer
Input: `{'action': 'left_click', 'coordinate': [623, 73]}`




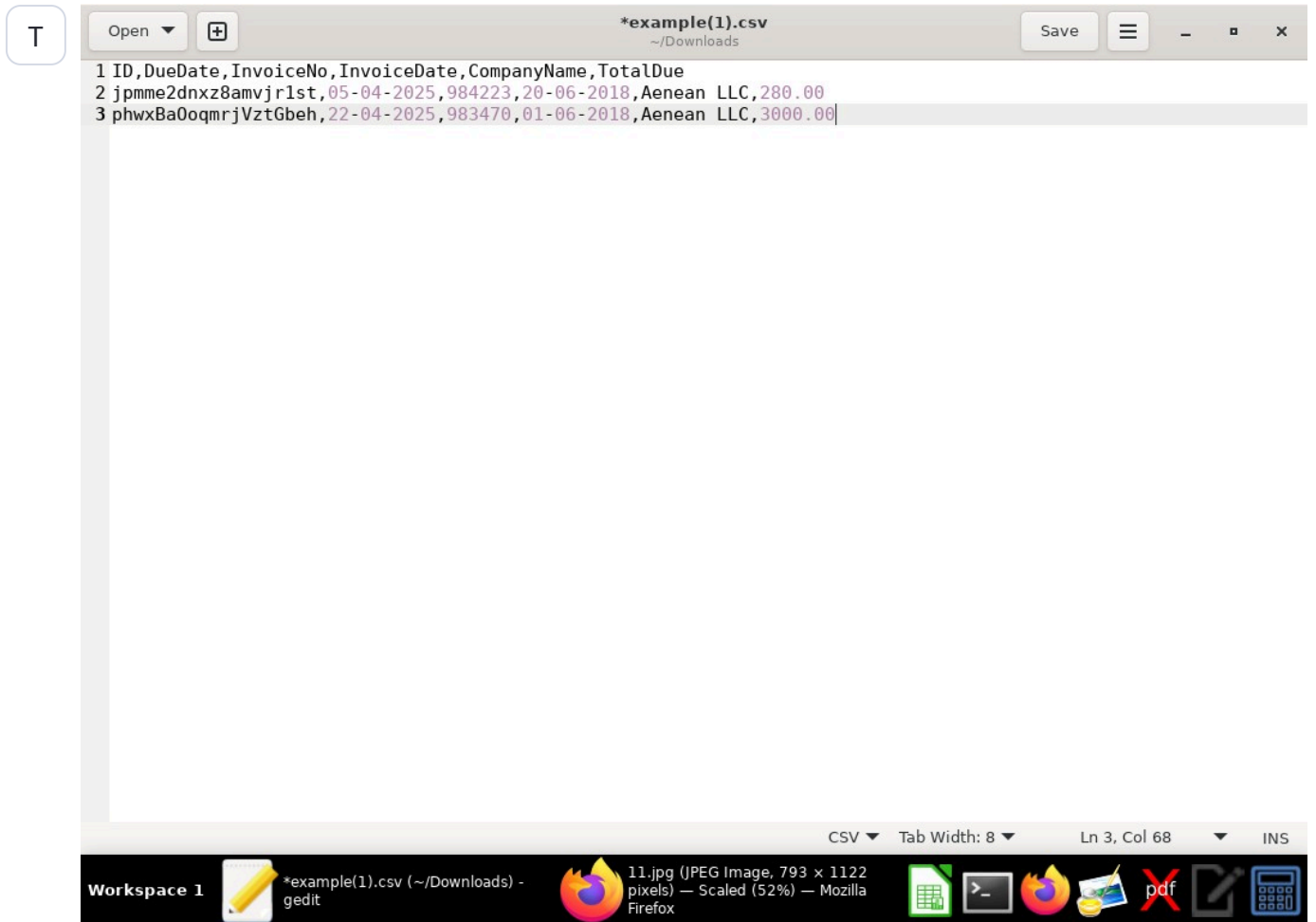
 Good, now I'll press Enter to create a new line and add the second invoice data:


 Tool Use: computer
Input: {'action': 'key', 'text': 'Return'}



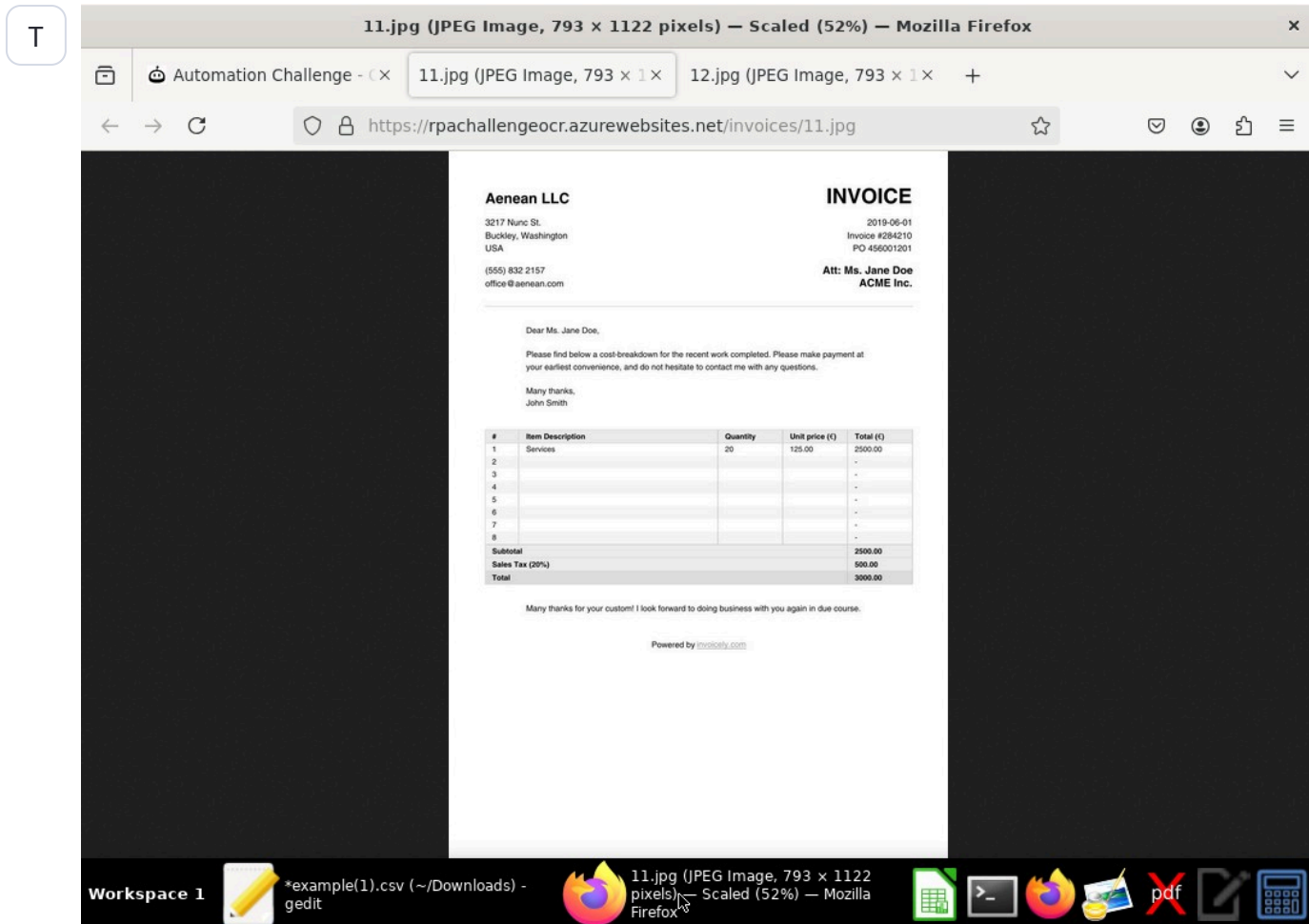
 Perfect! Now I'll type the second invoice data, converting the date from 2018-06-01 to 01-06-2018:

 Tool Use: computer
Input: {'action': 'type', 'text': 'phwxBa0oqmrjVztGbeh,22-04-2025,983470,01'}



 Excellent! Now I need to continue with the third invoice. Let me go back to Firefox to process the third invoice:

 Tool Use: computer
Input: `{'action': 'left_click', 'coordinate': [496, 738]}`



I need to go back to the main invoice list. Let me click on the first tab:

Tool Use: computer
Input: {'action': 'left_click', 'coordinate': [155, 58]}

T

Automation Challenge - OCR — Mozilla Firefox

AutomationChallenge - x 11.jpg (JPEG Image, 793 x 1x 12.jpg (JPEG Image, 793 x 1x +

← → ↺ https://rpachallengeocr.azurewebsites.net 50% ☆

Automation Challenge

Instructions

- The goal of this challenge is to create a workflow that will read every table row and download the respective invoices.
- From the invoices, you will have to extract the **Invoice Number**, **Invoice Date**, **Company Name** and **Total Due**.
- You will have to build and upload a CSV file with the data extracted from each invoice, the ID and Due Date from the table, **only for the invoices for which the Due Date has passed or is today**.
- The actual countdown of the challenge will begin once you click the **Start** button and will end once the CSV file is uploaded; until then, you may play around with the table on the right without receiving penalties.
- Below you will find an example CSV file in order to see the required format for the end result and two sample invoices. The formats of the invoices will be exactly as in the samples and they will not change. The challenge expects the uploaded CSV to be in the exact same format as the example CSV, including the formatting of the cells, and the rows should be in the same order as they appear in the table. Any difference will result in a failed challenge.
- Good luck!

[Download example CSV](#)
[Download sample invoice 1](#)
[Download sample invoice 2](#)

In Progress

#	ID	Due Date	Invoice
1	jpmorlp3enk3demy61zb	05-06-2025	Download Invoice
2	fj3wd8xl03pdrivdlas0h	22-04-2025	Download Invoice
3	5gja45jpc7mtyooxdunbp	24-04-2025	Download Invoice
4	8bhqdwty26ilhwu07	19-05-2025	Download Invoice

Previous 1 2 3 Next

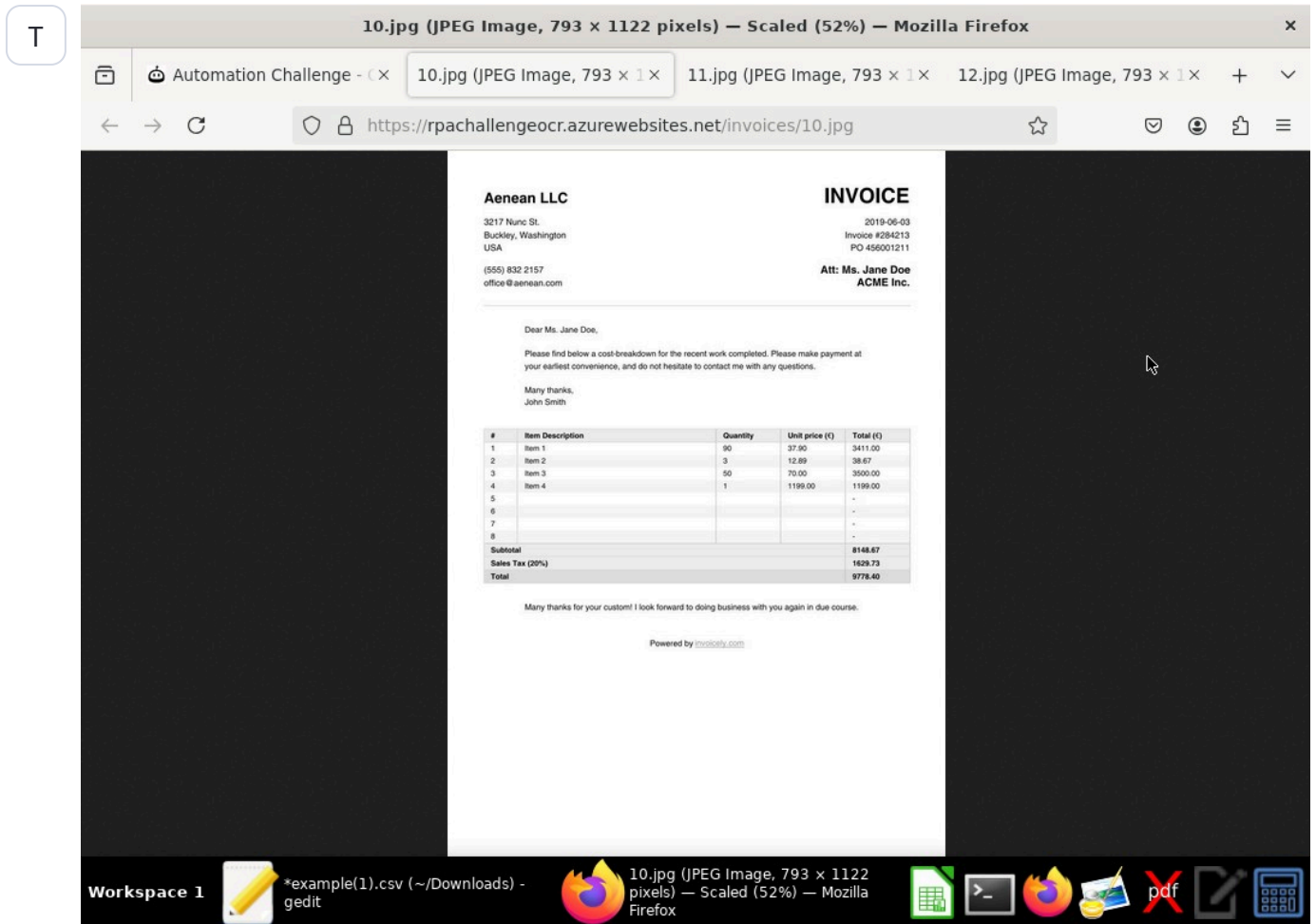


Great! Now I'll click on the third invoice download button (ID: 5ghe3mFpv7mtygozkrRx):



Tool Use: computer

```
Input: {'action': 'left_click', 'coordinate': [887, 293]}
```



Excellent! The third invoice has opened. I can extract the following information:

Invoice 3 Details:

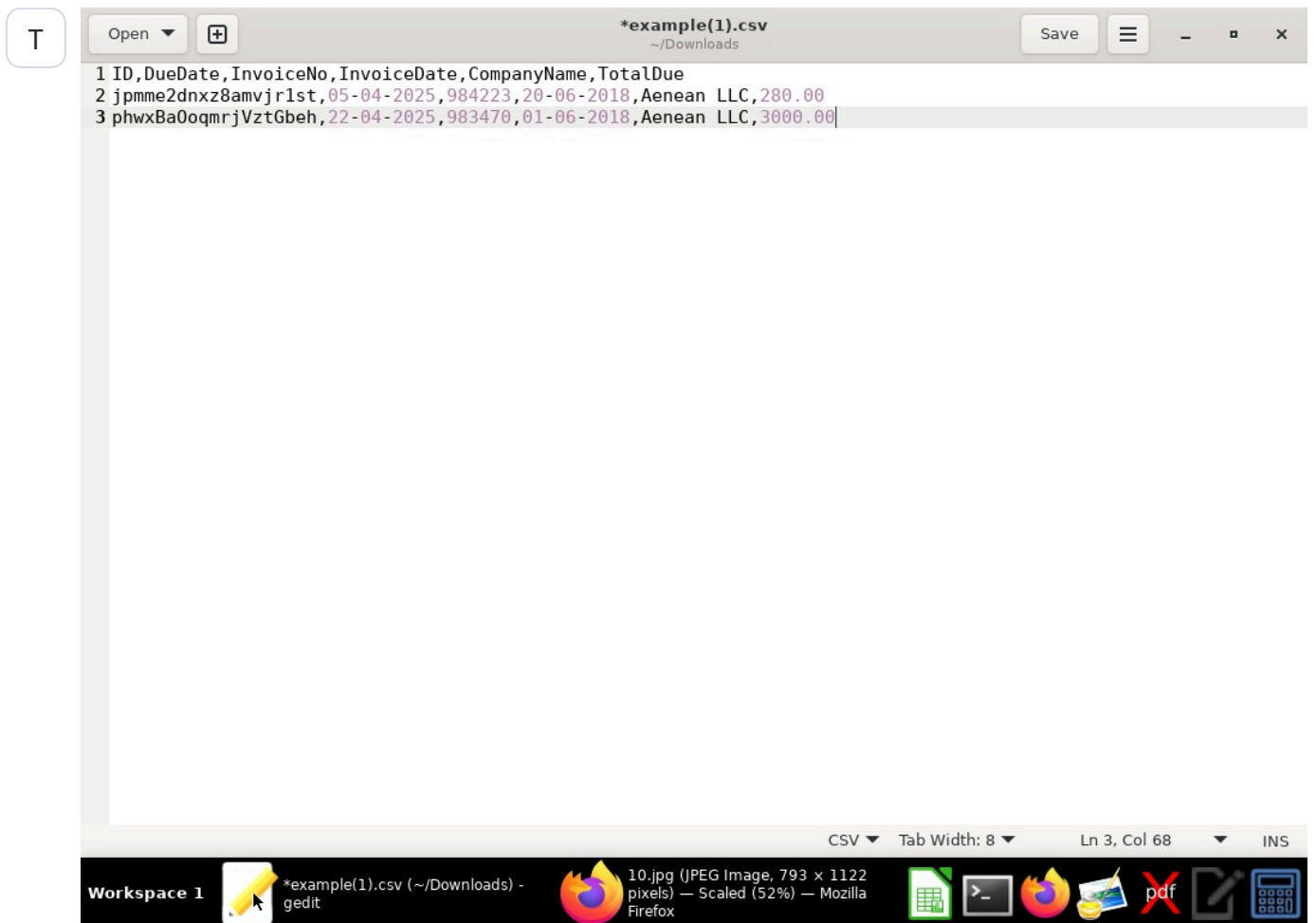
- Company Name: Aenean LLC
- Invoice Number: #983473
- Invoice Date: 2018-06-03
- Total Amount: \$975.40
- ID: 5ghe3mFpv7mtyqozkrRlx
- Due Date: 24-04-2025

Now let me go back to the CSV file to add this data:



Tool Use: computer

Input: `{'action': 'left_click', 'coordinate': [144, 738]}`

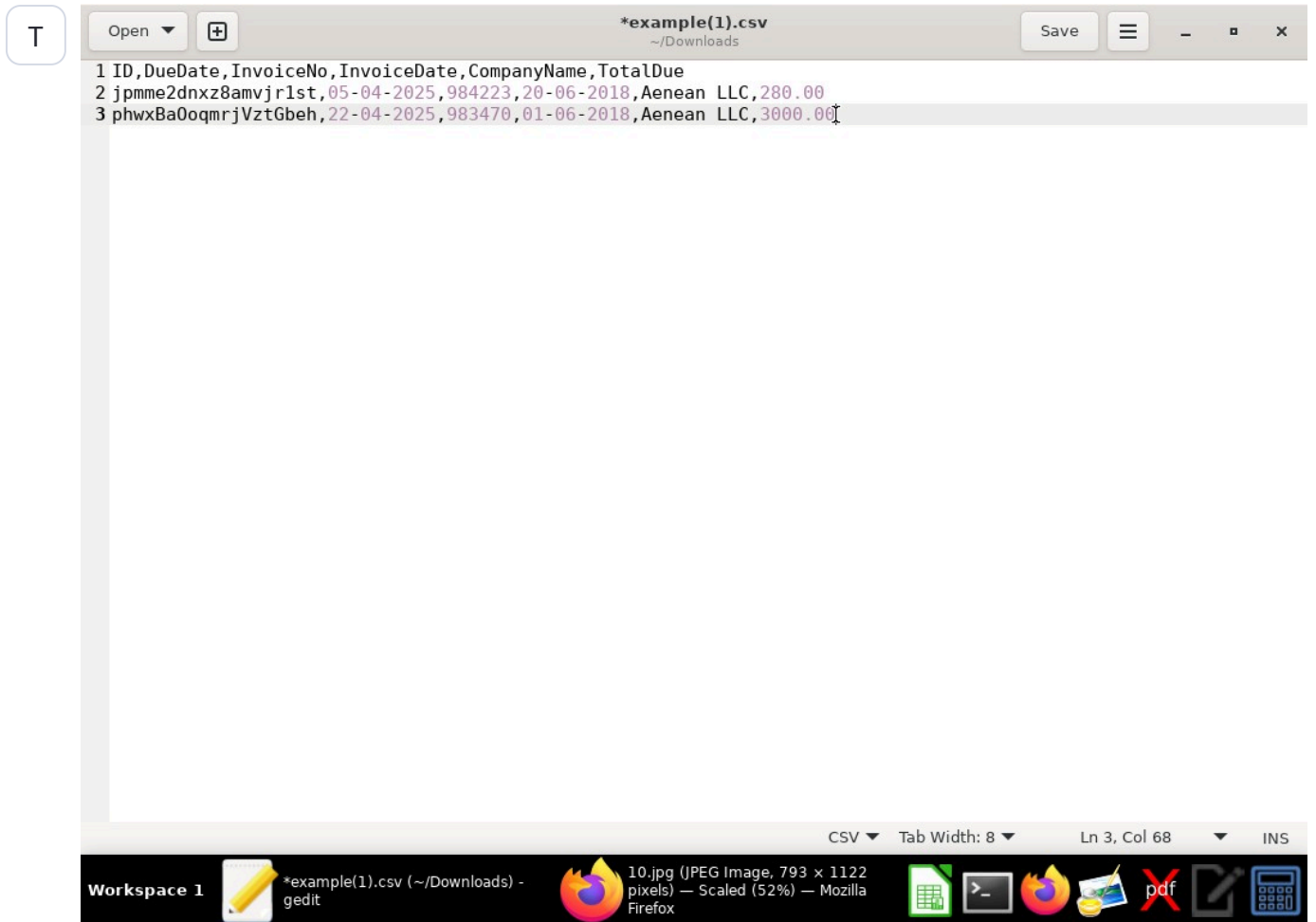



I need to position the cursor at the end of line 3 and add a new line. Let me click at the end of the second invoice entry:




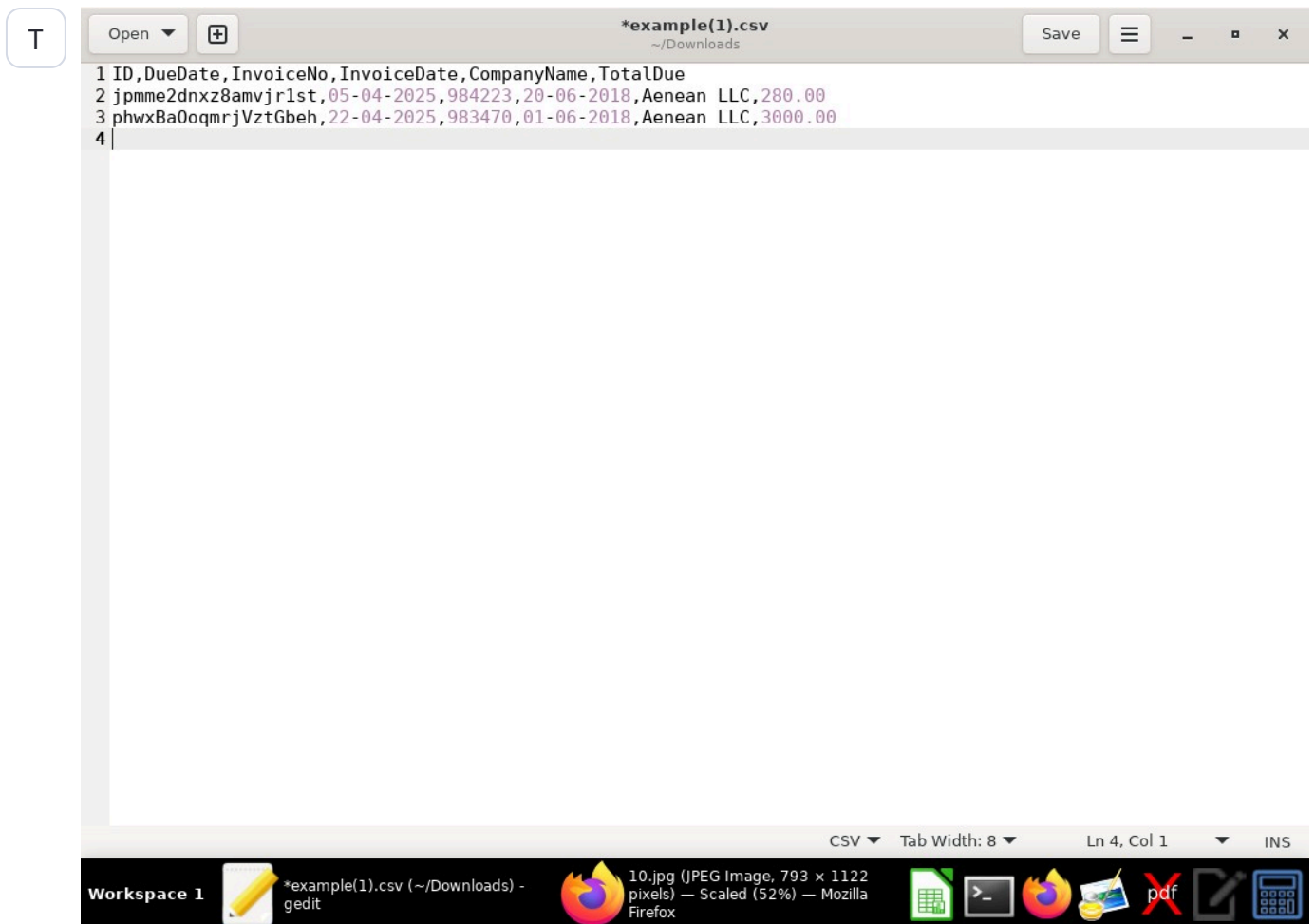
Tool Use: computer

Input: `{'action': 'left_click', 'coordinate': [629, 92]}`



 Good, now I'll press Enter to create a new line and add the third invoice data:

 Tool Use: computer
Input: {'action': 'key', 'text': 'Return'}



Perfect! Now I'll type the third invoice data, converting the date from 2018-06-03 to 03-06-2018:

Tool Use: computer
Input: {'action': 'type', 'text': '5ghe3mFpv7mtyqozkrRlx,24-04-2025,983473,'}

T

Open

*example(1).csv

~/Downloads

Save

1 ID,DueDate,InvoiceNo,InvoiceDate,CompanyName,TotalDue

2 jpmme2dnxz8amvjrlst,05-04-2025,984223,20-06-2018,Aenean LLC,280.00

3 phwxBa0oqmrjVztGbeh,22-04-2025,983470,01-06-2018,Aenean LLC,3000.00

4 5ghe3mFpv7mtyqozkrRlx,24-04-2025,983473,03-06-2018,Aenean LLC,975.

CSV

Tab Width: 8

Ln 4, Col 67

INS

Workspace 1

*example(1).csv (~/.Downloads) - gedit

10.jpg (JPEG Image, 793 x 1122 pixels) — Scaled (52%) — Mozilla Firefox

Type a message to send to Claude to control the computer...

