A REPORT

ON

INVOICE FETCHING AND BARCODE STICKER GENERATION USING ZOHO CREATOR AND ZOHO BOOKS INTEGRATION

BY

Aakifah Minhaj

2023A7PS0222U

AT

CODELATTICE

KERALA,INDIA

A Practice School – I station of

BITS Pilani, Dubai Campus

Dubai International Academic City, Dubai

UAE

(JUNE 2025 – AUGUST 2025)

BITS Pilani, Dubai Campus
Dubai International Academic City, Dubai
UAE

Station: CODELATTICE **Centre**: KERALA
Duration: 09 June 2025 – 1 August 2025 **Date of Start**: 09 June 2025
Date of Submission: 28 July 2025

Title of the Project:
Automated Invoice & Barcode Generation via Zoho Creator and Zoho Books Integration

ID No. / Name of the Student: 2023A7PS0222U, Aakifah Minhaj

Discipline of Student: Computer Science Engineering

Name(s) and Designation(s) of the Expert(s):Ms Rasmi(Tech Head),Shruthi (CEO)

Name of the PS Faculty: Dr Vivek

Key Words:
Zoho Creator, Zoho Books API, Barcode Generation, Deluge, Invoice Automation, Zoho Books

Project Area(s): Software Development, Low-Code Platforms, Cloud Integration, API Automation, UI/UX for Print Templates

Abstract: The aim of this project was to utilize Zoho Creator to create printable barcode stickers and automatically fill in the process of fetching invoice details from Zoho Books. Using Deluge scripting, the process involved integrating Zoho Creator and Zoho Books using API calls. Custom fields like MPN, Style, Size, Gender, and MRP are some of the fundamental invoice and line item information which are retrieved automatically by the Zoho Creator workflow when an invoice is being created in Zoho Books. They are stored and the barcode image links are created from MPN values using the bwip-js API. Print templates with spaces like Style, MRP, Gender, Size, MFD, Address, and the barcode were designed in Zoho Creator to precisely fit the sticker format that was provided by the client. The project provided enlightening information on Zoho's low-code capabilities and API utilization and proved the practical use of cloud-based integration and automation.

Signature of Student Signature of Faculty

Date: 1/08/2025 Date:01/08/2025

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my deep gratitude to Prof.Souri Bannerjee, the Director of BPDC, for providing me with the opportunity to apply and explore the practical aspects of engineering principles that we have learned in our classrooms.

I would also like to take this opportunity to express my deep sense of gratitude to Dr.Vivek, our PS Faculty, who has been of tremendous help throughout the internship period.

Lastly, my warm regards to Dr.Gomathi Bhavani, Dean of the Practice School Division, for providing me with the opportunity to explore into the technical domain, which has enabled me to gain some invaluable takeaways and experiences from the corporate world.

Aakifah Minhaj

2023A7PS0222U

CONTENTS

Chapter 5 RESULTS

Chapter 6 CONCLUSION

List Of Figures

Chapter 1 – Introduction

1.1 Introduction to Codelattice

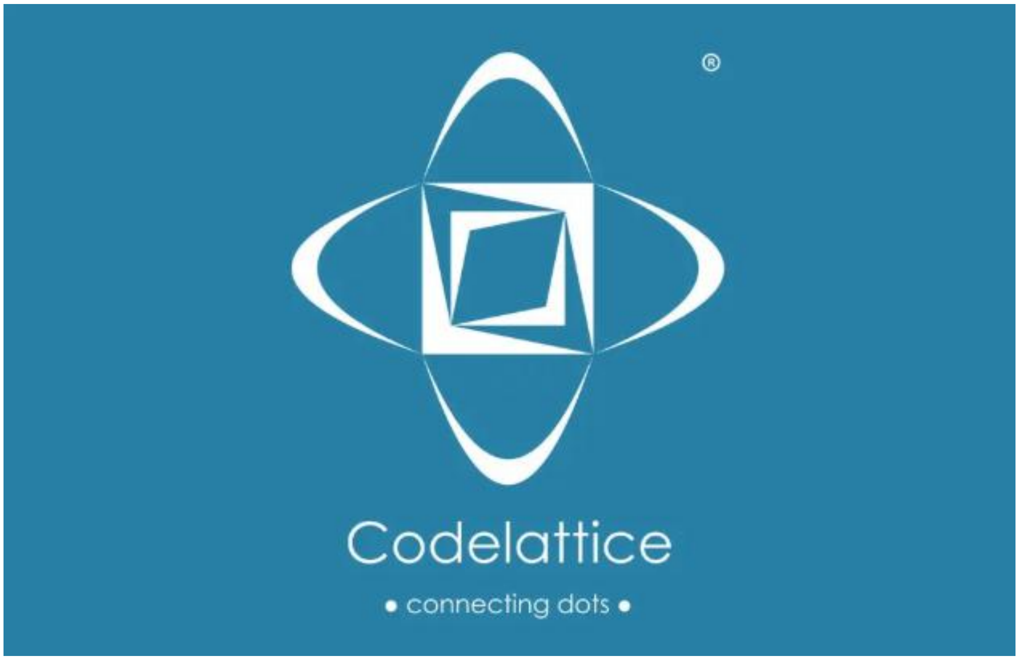


Fig 1.1 CODELATTICE

I am undertaking my Practice School-I internship at Codelattice, a technology solutions provider based in Kerala,India. The company specializes in providing digital automation solutions and customized software to a range of companies, including startups and SMEs.

My research is supervised by Ms Rasmi who is the Technical Head of Codelattice.

Codelattice is a company involved in the development of innovative web and mobile applications with strong focus on integration and automation using cloud-enabled technologies. The company has significant experience in the development of applications with Zoho Creator as well as integration of Zoho Books with other Zoho applications and customized automation for business operations. Additionally, Codelattice follows a strong customer-centric approach in its development methodologies where every solution is carefully designed to match the particular business requirement of its customers. Codelattice mainly provides the following services:

- Custom Software Development The development of apps using Zoho Creator.

- Integration with the Zoho Ecosystem (CRM, Inventory, Books Automation of workflows and process optimization

- Technical Support and Maintenance

Codelattice is built to be a solid and flexible business application that fosters innovation and expansion, thus positioning itself as a trusted technology partner for companies undergoing digital transformation.

1.2 Overview of Zoho Creator and Zoho Books

The low-code business application development platform known as Zoho Creator allows users to create, build, and deploy customized business applications with little to no programming skills. It includes a simple drag-and-drop interface for the design of forms, workflows, dashboards, and reports. Furthermore, it supports the addition of more complex logic and automation using its scripting feature based on Deluge (Data Enriched Language for the Universal Grid Environment). The platform is especially beneficial for small and medium-sized companies (SMBs) or small and medium-sized enterprises (SMEs) that seek to automate business processes at reduced infrastructure costs, as it is effective for real-time data capture, automated calculations, and cross-platform deployment across both web and mobile platforms.

Zoho Books is a cloud accounting application that is tightly integrated into the overall Zoho suite. Zoho Books allows companies to effectively manage their taxes, create financial reports, track invoices and payments, and manage their financial activities. In addition, it features a strong API along with the support for integrating external applications and other products in the Zoho suite. The combination of Zoho Books and Zoho Creator allows a complete business process automation solution. Zoho Creator allows for creating custom interfaces and automates the process of bringing data from in-house systems into Zoho Books via APIs, and Zoho Books takes care of the core accounting and invoicing capabilities.

In this project, Zoho Books and Zoho Creator were combined to automatically retrieve invoice data via APIs, extract particular custom fields like MPN, Style, Size, Gender, and MRP, and use this information to create barcode sticker patterns. This automation guarantees consistency in invoice data, drastically cuts down on manual labor, and offers a ready-to-print solution that satisfies the client's physical sticker needs.

1.3 Tools Used

To make the integration of the barcode sticker generation and the retrieval of invoice details seamless, multiple tools and technologies were used for this project. The below mentioned is the list of the most important tools:

1.3.1 Zoho Creator

Fig 1.2 ZOHO CREATOR

The central source of development was Zoho Creator. Custom forms were created, the workflows were defined using Deluge scripting, and the logic behind retrieving the invoice details from Zoho Books was created using it. It was used to create the reports and print templates that were needed for generating the barcode labels as well.

1.3.2 Zoho Books

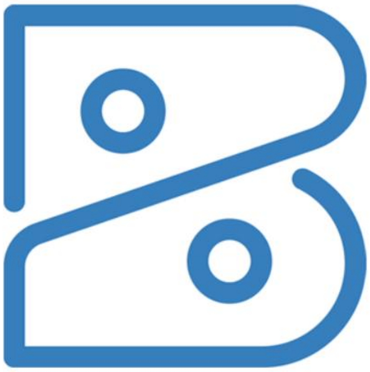


Fig 1.3 ZOHO BOOKS

Invoices were first designed based on the accounting and invoicing computer program Zoho Books. The software possessed an open API, and the project utilized it to retrieve invoice data and line-item information such as Style, Size, Gender, MPN, and MRP.

1.3.3 Deluge Scripting Language

Fig 1.4 DELUGE

Deluge (Data Enriched Language for the Universal Grid Environment) is Zoho's proprietary scripting language, and it's utilized in Zoho Creator to automate data processing, create workflows, and invoke API calls. Invoice data was extracted and manipulated heavily with it.

1.3.4 Zoho Creator Print Template Editor

The physical layout of the barcode stickers has been designed using the print template editor of Zoho Creator. According to the client's requirement, fields like Style, MRP, Gender, Size, MFD, and Barcode have been placed graphically.

Chapter 2: Objectives and Methodology

2.1 Objectives of the Project

The primary objective of this project was to make use of Zoho Creator's connectivity features with Zoho Books in automating and simplifying the generation of barcode stickers and fetching invoice information. The project was executed at Codelattice, with the purpose of fulfilling a client's immediate requirement in the fashion sector.

Personal goals were:

- Extraction of invoice details from Zoho Books automatically using Deluge scripting in Zoho Creator.
- Parsing and saving important fields of invoice like Style, MPN, Gender, Size, MRP, and MFD.
- Employing external services (such as BWIP-JS) and the built-in barcode field to create barcodes dynamically from MPN.
- Designing a custom print template in Zoho Creator that matches the client's real sticker size

Offering a semi-automated system that allowed the client to print barcode labels directly from invoice data with little or no human intervention was the final need.

2.2 Methodology

The project utilized an ordered iterative development approach that was segmented into the following crucial steps to achieve the above aims:

2.2.1. Requirement Gathering and Understanding

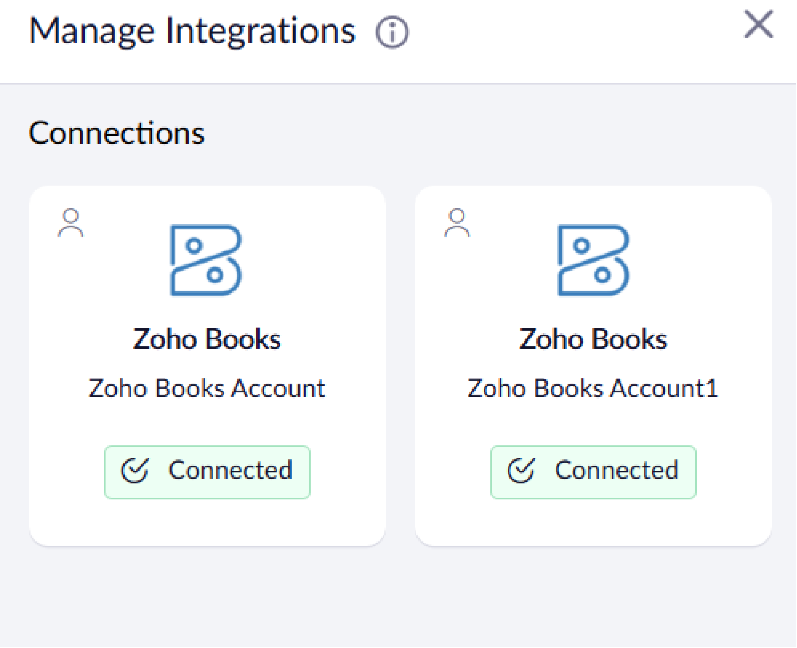


Fig 2.1 INTEGRATING CREATOR AND BOOKS

- Client requirements were collected through interaction with the Codelattice team and the final client.

- Comprehension of fields needed on the end printed sticker and how the invoice is generated in Zoho Books.

2.2.2 Form and Subform Creation in Zoho Creator

- Two forms were created: Zoho Invoice to get and hold invoice details.

- EG_Invoice for real sticker print reporting.

- Subforms have been used to handle item level details such as quantity and product properties.

2.2.3. Deluge Workflow Implementation

• Workflow Implementation Using Deluge: Two principal processes were implemented in the Zoho Invoice form: fetching_zoho_invoice_details and fetchInvoiceOnID.

•The Zoho Books API was utilized within these workflows to:

•Retrieve the invoice ID using the invoice number.

•Fetch the line item details.

•Fetch custom fields at the item level, e.g., MPN, Style, and MRP.

2.2.4. Barcode Generation

- The MPN value was used in a BWIP-JS URL in order to generate a Code128 barcode image dynamically.

- The image URL for display and printing was saved in an image field called Barcode_Image.

2.2.5. Print Template Design

- Both the EG_Invoice and Zoho Invoice forms were assigned a print template.• The layout was changed to accommodate the client's desired physical sticker.•Style, MRP, Barcode, Gender, Size, MFD, and Address were the fields placed to accommodate the structure required.

2.6.6. Testing and Debugging

- Zoho Books' real invoice entries were employed to test workflows

- API requests were debugged through alerts and information logs.

- Creator reports were used to check rendering of pictures and population of subforms.

Chapter 3: Implementation Of Modules

Some platform-level and technical issues were faced during the implementation of the Zoho Creator barcode sticker automation system. Such issues have been discussed in this chapter along with the appropriate strategies undertaken to overcome them.

3.1 API Integration and Data Parsing

Challenge Encountered:

The huge nested JSON responses the Zoho Books APIs return needed to be parsed so that some item-level information like MPN, Style, MRP, and Size could be extracted. It was also tough to map custom fields at the item level to description fields.
When we first tried to fill in fields of the subform with the parsed data, input errors were found, namely where data structure mismatch existed due to wrong custom fields missing or badly structured descriptions.

Debugging:

- At certain points (after invoice retrieve, after parsing description, after iterating over custom fields), there was a certain debugging strategy used that made use of info statements to print intermediate outcomes. This helped us to:

- Ensure the data was coming back from the API correctly.

- Locate the failure point of the logic or return null.

- Ensure the MPN and barcode URLs being created for each item.

Module Implemented:

- The solution was calling invokeurl and for each statements to generate formatted Deluge script that iterated through the line items on the invoice.

- To break the description into style, size, and gender for mapping, .toList("") was used.

- To prevent input errors and blank fields in the subform, null checks and fallback handling were used.

- To prevent Creator forms crashing while filling, errors were corrected by actually checking list sizes and preventing incorrect entries.

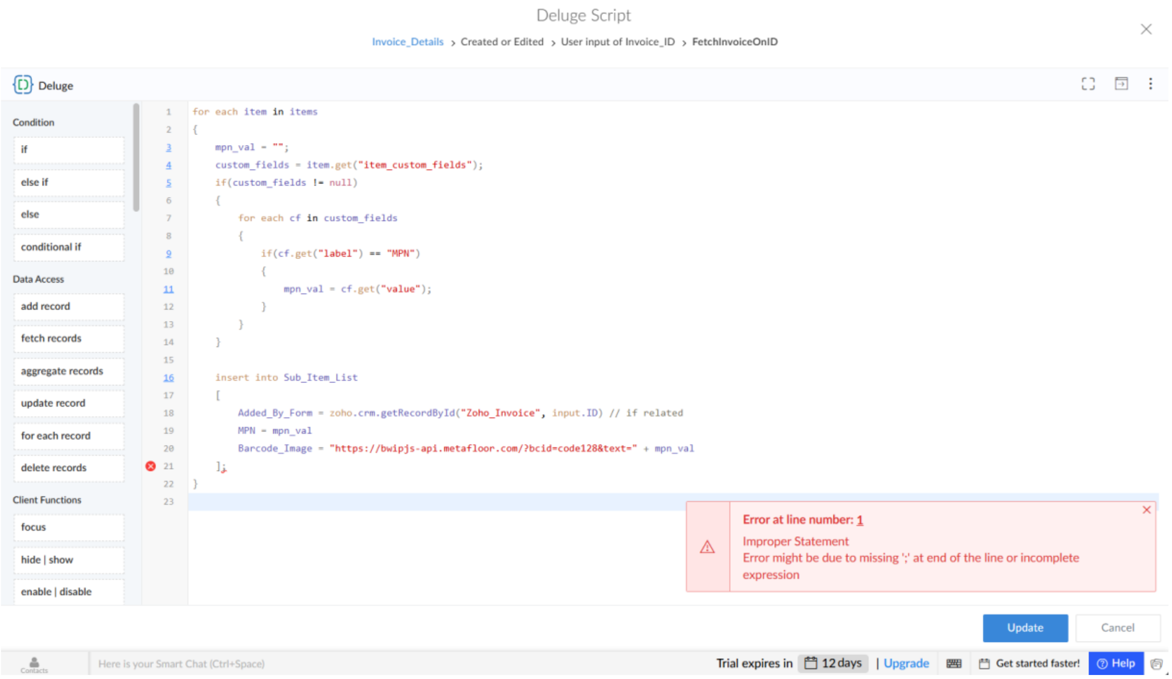
3.2 Multiple Workflows for Reliability of Data Fetching

Challenge Encountered:

Zoho Books' get invoice by number and get invoice by ID APIs had to be invoked together, as depending on either one alone was insufficient to retrieve all of the necessary item-level and custom field data.

Module Implemented:

- Two modular procedures were established:
- FetchInvoiceOnID: this initially retrieves the invoice ID followed by the complete item and custom field details;
- fetching_zoho_invoice_details: this retrieves the description and maps the invoice data.
- This split greatly simplified debugging and enhanced error identification.
- Despite complex invoices, we made sure that data was flowing consistently and Creator forms were updating uniformly.



```
for each item in items
{
    mpn_val = "";
    custom_fields = item.get("item_custom_fields");
    if(custom_fields != null)
    {
        for each cf in custom_fields
        {
            if(cf.get("label") == "MPN")
            {
                mpn_val = cf.get("value");
            }
        }
    }

    insert into Sub_Item_List
    [
        Added_By_Form = zoho.crm.getRecordById("Zoho_Invoice", input.ID) // if related
        MPN = mpn_val
        Barcode_Image = "https://bwipjs-api.metafloor.com/?bcid=code128&text=" + mpn_val
    ];
}
```

Error at line number: 1
Improper Statement
Error might be due to missing ';' at end of the line or incomplete expression

Fig 3.1 NOT WORKING FOR EACH LOOP

3.3 Barcode Image Generation

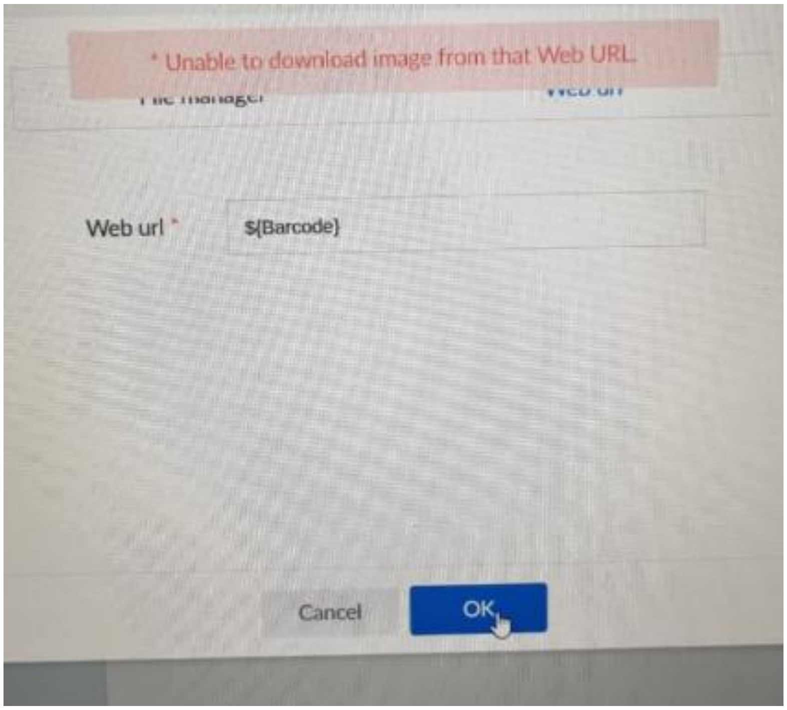
Challenge Encountered:

The original intention was to use the item's MPN as a barcode data and create a Code128 barcode image using the BWIP-JS API. A URL would be used to embed the image into the print template. But the images were not showing up in the Zoho Creator report or print template at all times even when the correct barcode URL (https://bwipjs-api.metafloor.com) was being created. This was most probably because of cross-origin or rendering problems in the Creator environment, particularly on the trial plan.

Module Implemented:

- We made use of Zoho Creator's own barcode field, which displays Code128 barcodes, rather than a remote image URL.

- The barcode field was then added to the print template to enable the sticker to display a Creator-native, scannable barcode, which resolved the issue. The MPN value was entered directly into this barcode field (Barcode_Image), ensuring it displayed correctly across records without any external API dependences.



* Unable to download image from that Web URL.

Web url * ${Barcode}

Fig 3.2 UNABLE TO DOWNLOAD BARCODE IMAGE FROM THE URL

3.4 Sticker Template Design Limitations

Challenge Encountered:

The customer required a precise sticker design (50 mm × 25 mm) with barcode fields and text positioned exactly where they are required. Pixel-accurate positioning and mm-based design control is not supported by Zoho Creator's print template editor.

Module implemented:

- The template was created manually in the Creator print template editor by using drag-and-drop.

- The following fields were replicated in the same order: gender, size, MFD, style (center left), MRP, address, and barcode (right).

- To maintain similarity in format, the identical template was replicated inside the EG_Invoice and Zoho Invoice forms.

- Exact proportionality was not possible, but the resulting output resembled the intended design.

3.5 Quantity-Based Barcode Duplication

Challenge Encountered:

Depending on quantity, the customer needed more than one barcode label per item (e.g., quantity = 5; 5 labels).
Dynamic replication is not supported by Zoho Creator, particularly when using free/trial

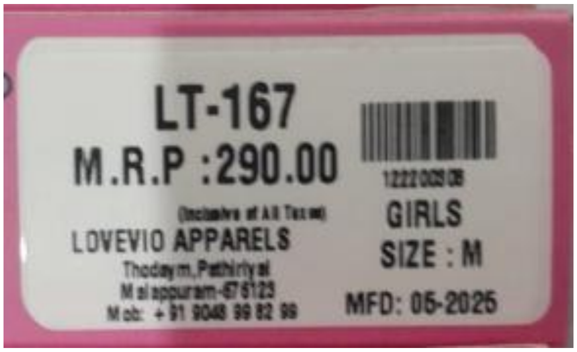
Module Implemented:

- It was recognized as a platform limitation.

- A theoretical solution that consisted of using subforms with repeated input was suggested, but trial limitations rendered it impossible.

- As a stopgap measure, the template now generates a barcode sticker for each line item, and also displays the quantity visually.

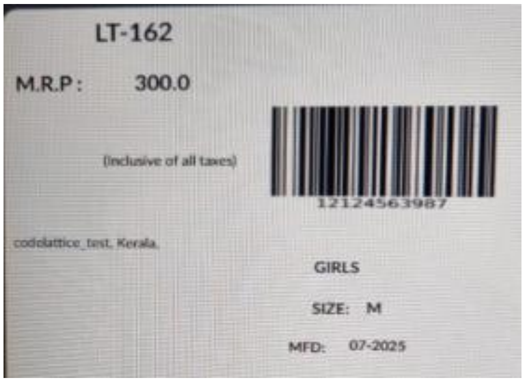
- This was piloted for usage and well documented for future modifications.



LT-167
M.R.P :290.00
(Inclusive of All Taxes)
LOVEVIO APPARELS
Thodaym,Pathiriyal
Malappuram-676123
Mob: +91 9048 99 82 99

122200908
GIRLS
SIZE : M
MFD: 05-2025

Fig 3.3 CLIENTS STICKER



LT-162
M.R.P : 300.0
(inclusive of all taxes)
12124563987
codelattice_test, Kerala.
GIRLS
SIZE: M
MFD: 07-2025

Fig 3.4 CODELATTICE STICKER

3.6 Creator Platform Constraints (Trial Plan)

Challenge Encountered:

This Zoho Creator application was designed and implemented all on the back of the limitations of a trial Zoho Creator account. Although the platform itself features strong workflow automation, third-party integration, and form/report customization functionality, various key features were available only with an annual subscription. This severely restricted the project's automation and scalability.
Some of the key limitations encountered were:

- API call quota and limited number of integration connections
- No custom domain access or public app embedding;
- No scripting scheduling or performing time-based actions;
- No Zoho Creator Sandbox available for safe testing of new processes;
- No advanced print customization or external print control

Despite these limitations, by fully concentrating on elements that could be accessed using the trial account, the core functionality of the application was developed successfully.

Module Implemented:

- Using the given invokeurl and basic scripting tools, the necessary procedures to create barcodes using external APIs (BWIP-JS) and retrieve invoices from Zoho Books were created.

- While dynamic on-success events and form-wide changes were out of reach, subforms and field logic were utilized to mimic data relationships.

- The value of pre-empting platform constraints and modifying customer expectations in an appropriate way was highlighted by this exercise. Though the trial limitations were stringent, they formed a useful constraint to test the fundamental reasoning of the program before scaling up.

Chapter 4: Sticker Printing and Zoho Creator Customization

4.1 Exploring Print Template Layouts in Zoho Creator

Exploring Print Template Layouts in Zoho Creator

A big part of our Practice School project was creating sticker print templates in Zoho Creator that exactly matched the physical size the client provided (50mm x 25mm).

We had to use HTML and CSS inside the Print Template editor to place each piece of information, like style, MRP, gender, size, MFD, and barcode, in the right spot with the correct font sizes and spacing. One big problem was that Zoho Creator didn’t have a live print preview, so we had to test everything by trial and error, either by printing or using other tools.

- Used HTML and CSS in Zoho Creator to line up all the elements properly.

- Placed the style in the middle left, the barcode on the right, and the other details below them.

- Removed extra tags and padding to fix problems with content overflowing.

- Adjusted line height, font size, and font weight to make the text easier to read on the small label.

- Encountered layout issues that needed repeated changes at the field level to fix.

4.2 Tools

Because the built-in editor in Zoho Creator had some limits, we used tools like JSFiddle and JSitor to show how the HTML layout would look on a label. These tools let us see in real time how changes in the code affected the alignment, which helped us adjust the layout better before sending it to Zoho Creator. We used placeholder values for the barcode and text fields so we could check how spacing, font weight, and alignment worked on different screen sizes.

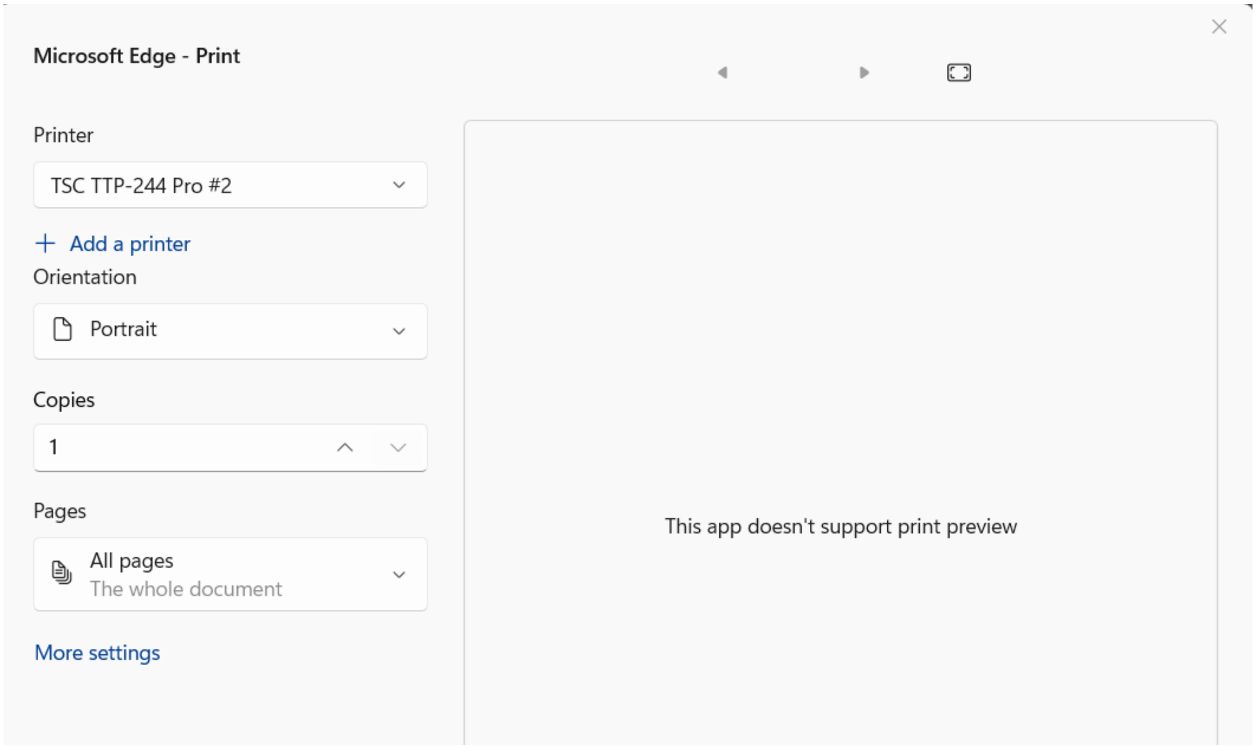
4.2.1 BarTender

Fig 4.1 BARTENDER

At the beginning of the sticker testing, we tried to mimic the client’s label printing setup by adding the TSC TTP-244 Pro printer model to our laptop. We didn’t actually connect to the client’s printer, just to access its settings and get a print preview of our 50 mm by 25 mm sticker design.We installed the right TSC drivers and set the correct stock size (50 mm wide, 25 mm tall) in the printer settings.

This was meant to give us a local print preview so we could check how everything would look—like positioning, size, and scaling—before testing it on-site. However, the setup didn’t support print preview. So we couldn’t check if the design elements, especially the barcode images, text, and spacing, would fit properly inside the sticker. This made it hard to know how the final print would look on the real thermal printer. Because of this problem, we looked into using BarTender as another tool.

BarTender is a professional software that lets you design and test barcode labels with a clear visual preview. It supports high-quality barcodes and accurate label sizes. After installing BarTender, we recreated the sticker layout on our own with the same dimensions (50 mm × 25 mm). This allowed us to see a full visual preview and test how each part—like the barcode, text fields such as MRP, Style, Gender, and Size, and spacing—would align properly.



Microsoft Edge - Print

Printer

TSC TTP-244 Pro #2

Orientation

Portrait

Copies

1

Pages

All pages
The whole document

This app doesn't support print preview

Fig 4.2 UNSUPPORTED PRINT PREVIEW

4.2.2 JSitor

Fig 4.3 JSITOR

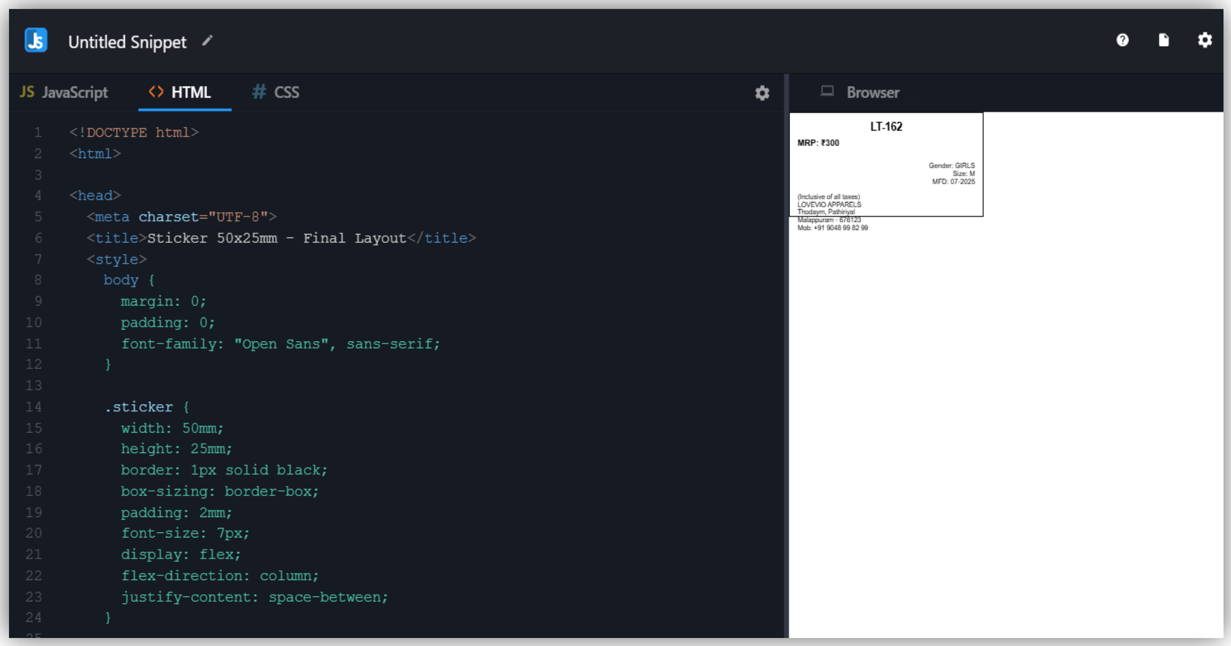
- We used JSitor to test and preview our HTML code for the sticker layout because Zoho Creator didn’t have a live print preview feature.

- The HTML code had layout for fields such as Style, MRP, Barcode Image, Gender, Size, and MFD.

- Even though JSitor helped us check the basic alignment and spacing, the barcode image (${ImageElement}) didn’t show up.

- This became a problem — we couldn’t be sure if the barcode and its related fields would fit inside the 50x25mm box or if they would go beyond the space.

- Because the image wasn’t visible, we had to switch to other software that supported barcode rendering.



```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Sticker 50x25mm - Final Layout</title>
  <style>
    body {
      margin: 0;
      padding: 0;
      font-family: "Open Sans", sans-serif;
    }

    .sticker {
      width: 50mm;
      height: 25mm;
      border: 1px solid black;
      box-sizing: border-box;
      padding: 2mm;
      font-size: 7px;
      display: flex;
      flex-direction: column;
      justify-content: space-between;
    }
```

LT-162

MRP: ₹300

Gender: GIRLS
Size: M
MFD: 07-2025

(Inclusive of all taxes)
LOVEVIO APPARELS
Thodaym, Pathiriyal
Malappuram - 676123
Mob: +91 9048 99 82 99

Fig 4.4 BARCODE NOT RENDERING

4.2.3 JsFiddle

Fig 4.5 JSFIDDLE

Unlike JSitor, JSFiddle was able to render the barcode image, so we moved to it.

- To test how each element displayed inside the 50mm × 25mm sticker box, we copied the sticker's whole HTML layout code into JSFiddle.

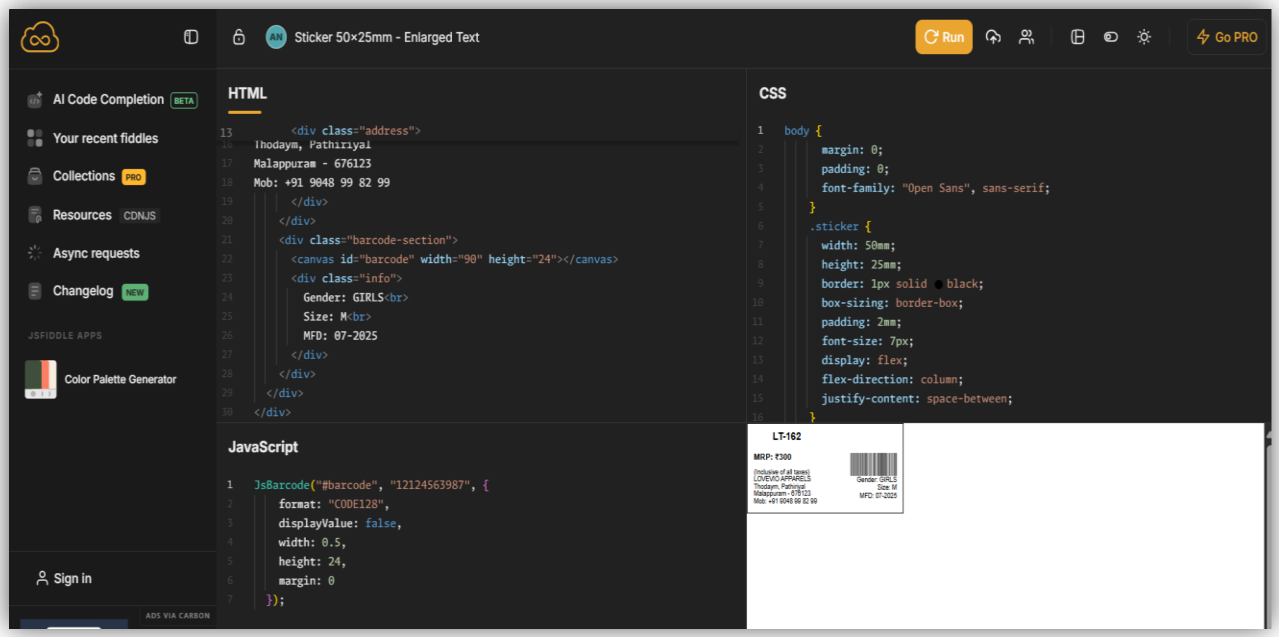
- We were able to visually verify that the layout was balanced and positioned correctly because the barcode image was visible.

- To prevent text overflow and guarantee correct sticker alignment, we checked font widths, padding, and spacing.

- The barcode was placed on the right side with no spaces below after being resized and positioned using JSFiddle.

- We ensured that Gender, Size, and MFD were correctly aligned beneath the barcode, and that Style, MRP, and Brand Address remained on the left.

- We were able to iterate rapidly without printing each time by using JSFiddle's live preview.



```html
        <div class="address">
Thodaym, Pathiriyal
Malappuram - 676123
Mob: +91 9048 99 82 99
        </div>
      </div>
      <div class="barcode-section">
        <canvas id="barcode" width="90" height="24"></canvas>
        <div class="info">
          Gender: GIRLS<br>
          Size: M<br>
          MFD: 07-2025
        </div>
      </div>
    </div>
</div>
```

```css
body {
    margin: 0;
    padding: 0;
    font-family: "Open Sans", sans-serif;
  }
  .sticker {
    width: 50mm;
    height: 25mm;
    border: 1px solid ● black;
    box-sizing: border-box;
    padding: 2mm;
    font-size: 7px;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
  }
```

```javascript
JsBarcode("#barcode", "12124563987", {
    format: "CODE128",
    displayValue: false,
    width: 0.5,
    height: 24,
    margin: 0
  });
```

```
    LT-162

MRP: ₹300
                                Gender: GIRLS
(Inclusive of all taxes)              Size: M
LOVEVIO APPARELS               MFD: 07-2025
Thodaym, Pathiriyal
Malappuram - 676123
Mob: +91 9048 99 82 99
```

Fig 4.6 PRINT PREVIEW WITH BARCODE

4.3 TSC TTP-244 Pro Printer Configuration and Calibration

The printer configuration was just as important as the Zoho template for producing precise label output. Using Windows Print Preferences, we manually configure the TSC TTP-244 Pro's label dimensions. The printer occasionally skipped labels or misplaced the print, even with the right template design. We fixed issue by making sure the gap settings were set appropriately and recalibrating the printer using the FEED button approach.

- In the printer's options, set the label size to 50 x 25 mm.

- To detect gaps, the printer was calibrated by holding FEED while it was turned on.

- By modifying gap height (2–3 mm), feed skipping and spacing were corrected.

- In addition to Creator customization, printer-side tweaking was crucial.

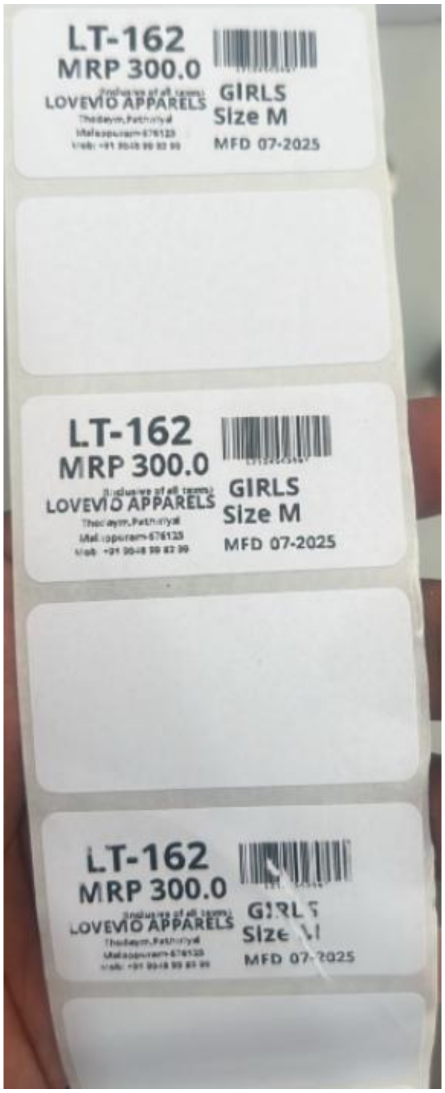
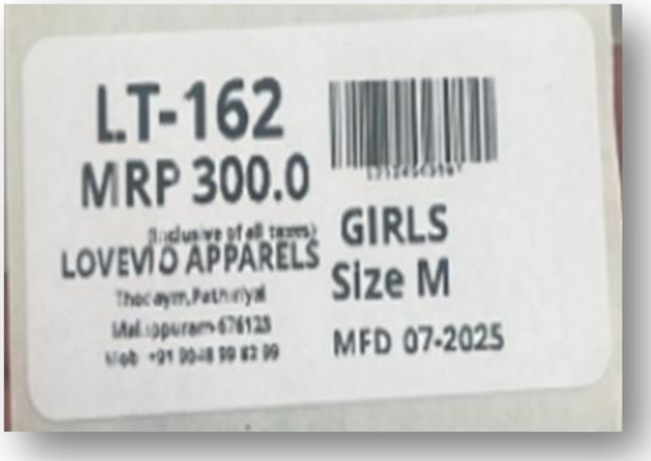


Fig 4.7 ALTERNATE STICKER PRINTING

4.4 Problems with the Zoho Account and Inability to Continue the Task

While the Zoho Creator trial account expired, we ran across a significant obstacle while evaluating the finished sticker outputs. The account wasn't renewed in time even though the support staff was emailed. Our momentum was halted since we were unable to test or make additional adjustments. After this phase, we also anticipated receiving a new task assignment from the supervisor, but the team was left in a condition of confusion because no explicit instructions were provided.

- A support request was made, but it was not promptly resolved; the Zoho Creator trial ended during final testing.
- After sticker testing, new duties were anticipated



LT-162

MRP 300.0

(Inclusive of all taxes)

LOVEVIO APPARELS

Thodayam,Pathiriyal

Malappuram-676123

Mob: +91 9048 99 82 99

GIRLS

Size M

MFD 07-2025

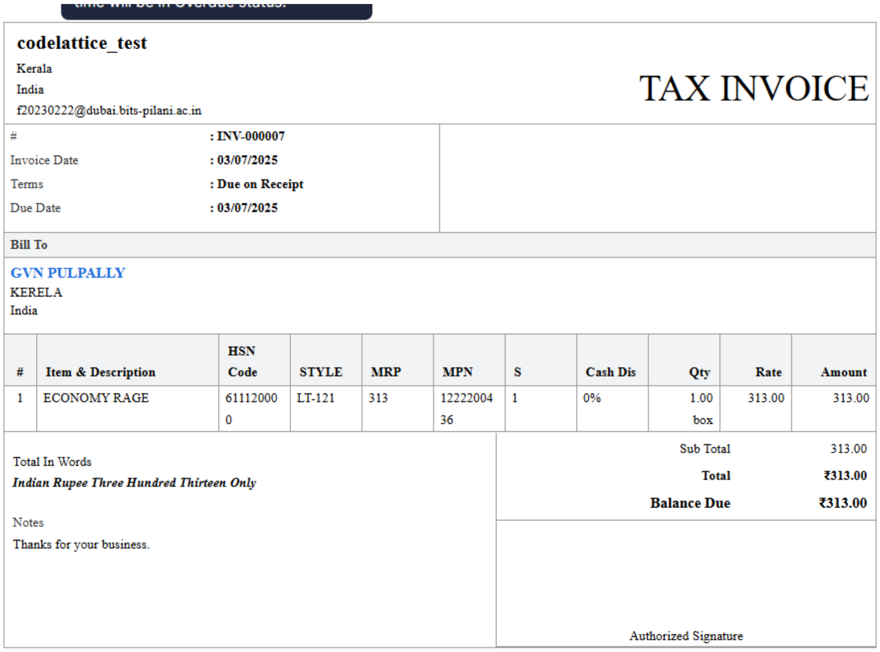
Fig 4.8 FINALISED STICKER

Chapter 5: Results

Automating the process of invoice data retrieval and barcode-based sticker printing were the major goals of the Zoho Creator application, which was designed and developed during the Practice School-I program and successfully integrated with Zoho Books. The following results were obtained:
By enabling API-based fetching using the invoice number or internal invoice ID, the application was able to easily automate the extraction of invoice data from Zoho Books. Besides invoice-level information like Customer Name and Invoice Date, the Deluge workflows were able to fetch precise item-level information like Style, MPN, Gender, Size, MRP, Quantity, and MFD. This kept Zoho Books and Zoho Creator in perfect sync.

Dynamic barcode generation from the Manufacturer Part Number (MPN) for every line item was one of the major results. While barcode generation was initially tried with a public API (BWIP-JS), it faced URL rendering issues. This was addressed by using Zoho Creator's inbuilt barcode field that correctly rendered Code128 barcodes which were scannable and merged with the print layout.
Within the limitations of Zoho Creator's print template, the client-provided sticker dimension—which was 25 mm by 50 mm—was duplicated. Style, MRP, Barcode, Gender, Size, Address, and MFD were all included in the print design to match the appearance of the client's actual sticker format. To make it easier to use and to eliminate duplicate data entry, the EG_Invoice form was chosen as the final print source. It provided a report perspective in which users are able to preview and print stickers through a single click per row.

In order to enable formatting consistency and flexibility, templates were established for both EG_Invoice and Zoho Invoice forms. Dynamic duplication of barcode stickers by quantity was limited by platform constraints, but the approach established a solid foundation. The app's well-documented and modular structure enables future automation improvement after the platform is enhanced. Generally, the project provided a functional integration of barcode sticker generation and invoice processing that was tailored to the business requirements of the client.



time will be in Overdue status.

codelattice_test
Kerala
India
f20230222@dubai.bits-pilani.ac.in

TAX INVOICE

#	: **INV-000007**
Invoice Date	: **03/07/2025**
Terms	: **Due on Receipt**
Due Date	: **03/07/2025**

Bill To

GVN PULPALLY
KERELA
India

#	Item & Description	HSN Code	STYLE	MRP	MPN	S	Cash Dis	Qty	Rate	Amount
1	ECONOMY RAGE	611120000	LT-121	313	1222200436	1	0%	1.00 box	313.00	313.00

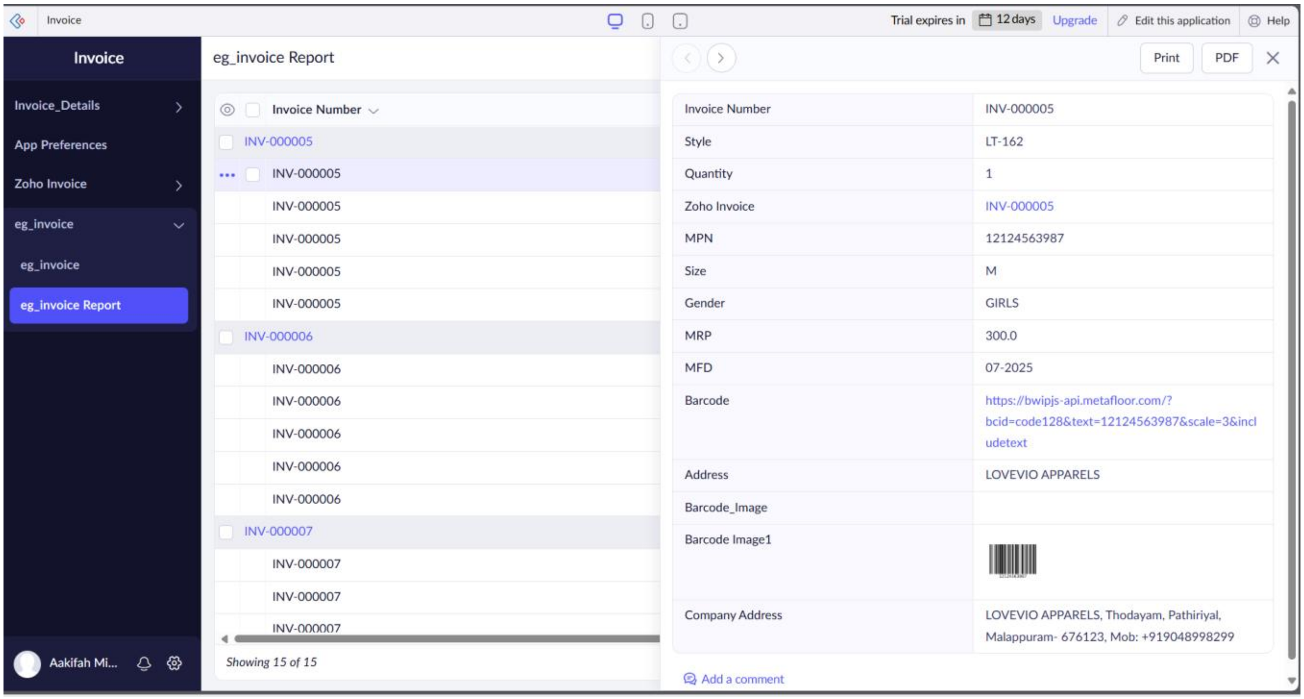
Total In Words
Indian Rupee Three Hundred Thirteen Only

Sub Total	313.00
Total	**₹313.00**
Balance Due	**₹313.00**

Notes
Thanks for your business.

Authorized Signature

Fig 5.1 INVOICE CREATED IN ZOHO BOOKS



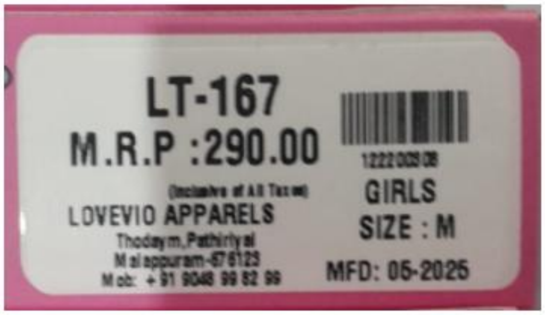
eg_invoice Report

Invoice Number
INV-000005
INV-000005
INV-000005
INV-000005
INV-000005
INV-000005
INV-000006
INV-000006
INV-000006
INV-000006
INV-000006
INV-000006
INV-000007
INV-000007
INV-000007
INV-000007

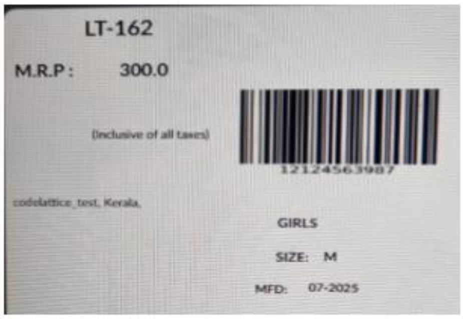
Showing 15 of 15

Invoice Number	INV-000005
Style	LT-162
Quantity	1
Zoho Invoice	INV-000005
MPN	12124563987
Size	M
Gender	GIRLS
MRP	300.0
MFD	07-2025
Barcode	https://bwipjs-api.metafloor.com/?bcid=code128&text=12124563987&scale=3&includetext
Address	LOVEVIO APPARELS
Barcode_Image	
Barcode Image1	
Company Address	LOVEVIO APPARELS, Thodayam, Pathiriyal, Malappuram- 676123, Mob: +919048998299

Fig 5.2 AUTOMATIC FETCHING OF RECORDS FROM THE INVOICE THROUGH ZOHO CREATOR AND ZOHO BOOKS INTEGRATION



CLIENTS STICKER



CODELATTICE PRODUCED STICKER

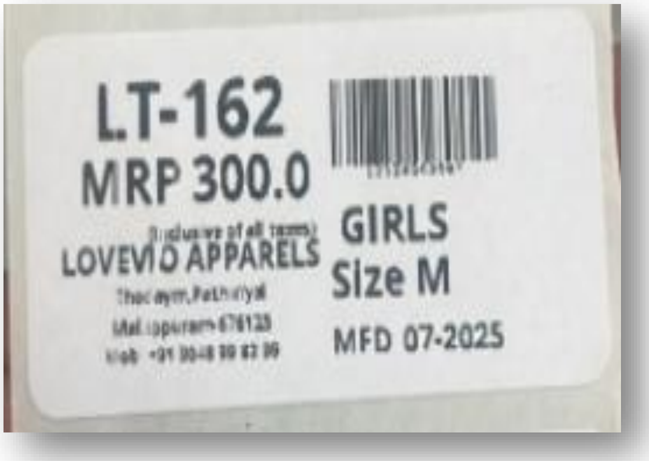


Fig 5.3 STICKERS

Chapter 6: Conclusion

Codelattice's Practice School-I project gave us the chance to work through a full process that involved cloud-based low-code tools, connecting APIs, and customizing designs. The main goal was to link Zoho Creator with Zoho Books to automatically get invoice data and create barcode stickers that matched the client's exact formatting needs. Even though we used Zoho Creator's trial version and faced some platform limits, we were able to build a strong system that pulls invoice data, extracts important details like Style, MRP, Gender, MFD, and more, and then creates barcode stickers with the right layout.

The project taught us several key skills, including using APIs through the invokeurl function, writing logic with Deluge scripting, and designing visuals with HTML and CSS.
We made print templates using inline CSS to fit all the data onto a small sticker size of 50mm × 25mm. One of the bigger challenges was printing on the right side vertically, which we solved with a creative CSS trick using transform: rotate(-270deg) to make sure everything fit neatly within the space.

Another important part of the project was using outside tools to check and preview our sticker layout before actually printing.
JSFiddle worked really well because it let us see the full HTML layout, including the dynamic barcode images. JSitor could show the layout basics, but it didn't support barcode images, which made it harder to check the alignment. These tools helped us try out different spacing, padding, and font sizes in a safe way before we started printing.

We also found out that code alone wasn't enough to fix all the issues.
We had to make sure the hardware was set up correctly too, like adjusting the label size to 50mm × 25mm and properly setting up the TSC TTP-244 Pro printer.Even after carefully designing the layout, there were still problems like text getting cut off vertically and white spaces showing up because of how the Creators print template worked. Printer calibration and adjusting the feed were important steps in getting the final layout right.

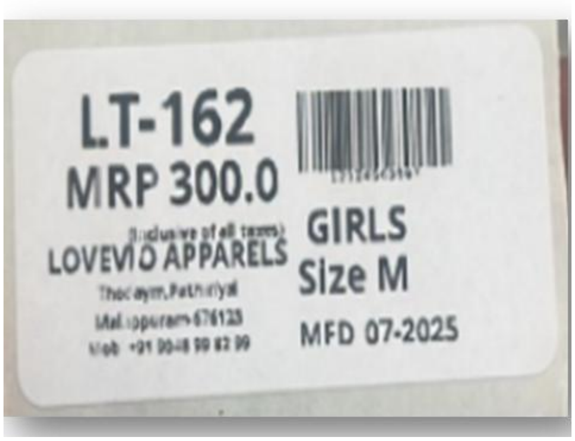
In the end, this project required a mix of front-end design, back-end logic, and hardware setup.
It also gave us a real-world understanding of how different parts — like the creator logic, APIs, HTML, and printer settings — all work together in a system. The main lesson was that checking both the code and the hardware is essential for the best results, something we learned through patience and constant testing.

The final output of this integrated system was a print-ready barcode sticker that matched the client's layout expectations within the 50mm × 25mm constraint.

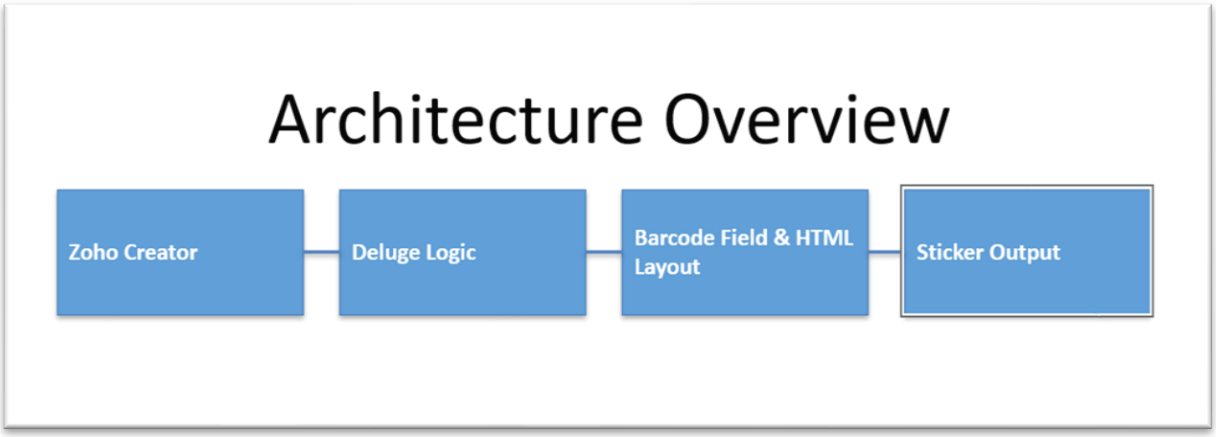
This journey not only sharpened our technical skills but also improved our problem-solving abilities as we faced real-time challenges that required quick thinking and adaptability. From refining CSS layout styles to repeatedly testing print outputs on a physical printer, every step demanded attention to detail. It also deepened our understanding of how visual formatting, data accuracy, and hardware behavior all intersect—especially in a constrained format like a 50mm × 25mm sticker. These tasks simulated what it feels like to develop client-facing solutions that must work under practical limitations.

Moreover, this hands-on experience emphasized the importance of coordination between software design and physical implementation. While Zoho Creator and APIs gave us the tools to automate data flow and visual generation, it was the fine-tuning of printer settings and calibration that ensured the final output met real-world expectations. This holistic approach—

combining cloud platforms, scripting, layout design, preview testing, and printer configuration—reflected the complexity of deploying a full-stack automated solution.



A simple system architecture diagram and description for Zoho Creator sticker printing project



Architecture Overview

Zoho Creator

Deluge Logic

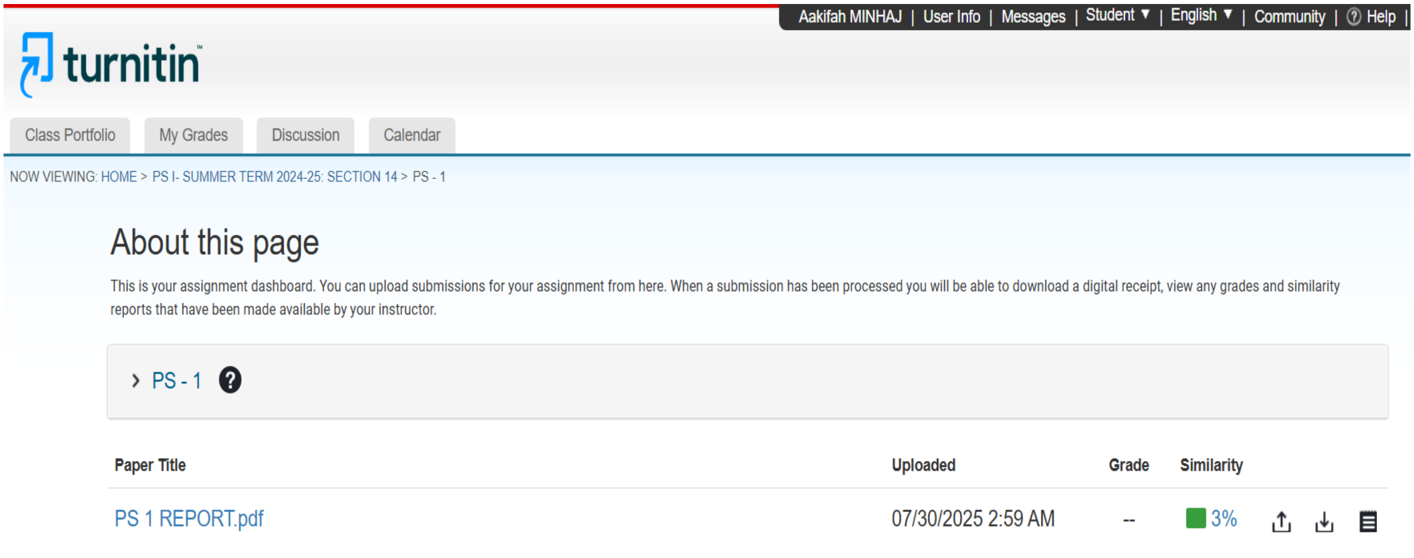
Barcode Field & HTML Layout

Sticker Output

References

1. Zoho Creator Documentation
   https://www.zoho.com/creator/help/
2. Zoho Books API Documentation
   https://www.zoho.com/books/api/v3/
3. Zoho Deluge Scripting Guide
   https://www.zoho.com/deluge/help/
4. Zoho Creator Community Forum
   https://help.zoho.com/portal/en/community/zoho-creator
5. Barcode Generation using BWIP-JS
   https://bwip-js.metafloor.com/
6. Zoho Creator Print Templates Help
   https://www.zoho.com/creator/help/app-settings/print-template.html
7. Zoho OAuth and Connections for APIs
   https://www.zoho.com/accounts/protocol/oauth/
8. Codelattice Official Website
   https://codelattice.com/

TURNITIN SUMMARY REPORT



Aakifah MINHAJ | User Info | Messages | Student ▼ | English ▼ | Community | ⓘ Help |
turnitin
Class Portfolio
My Grades
Discussion
Calendar

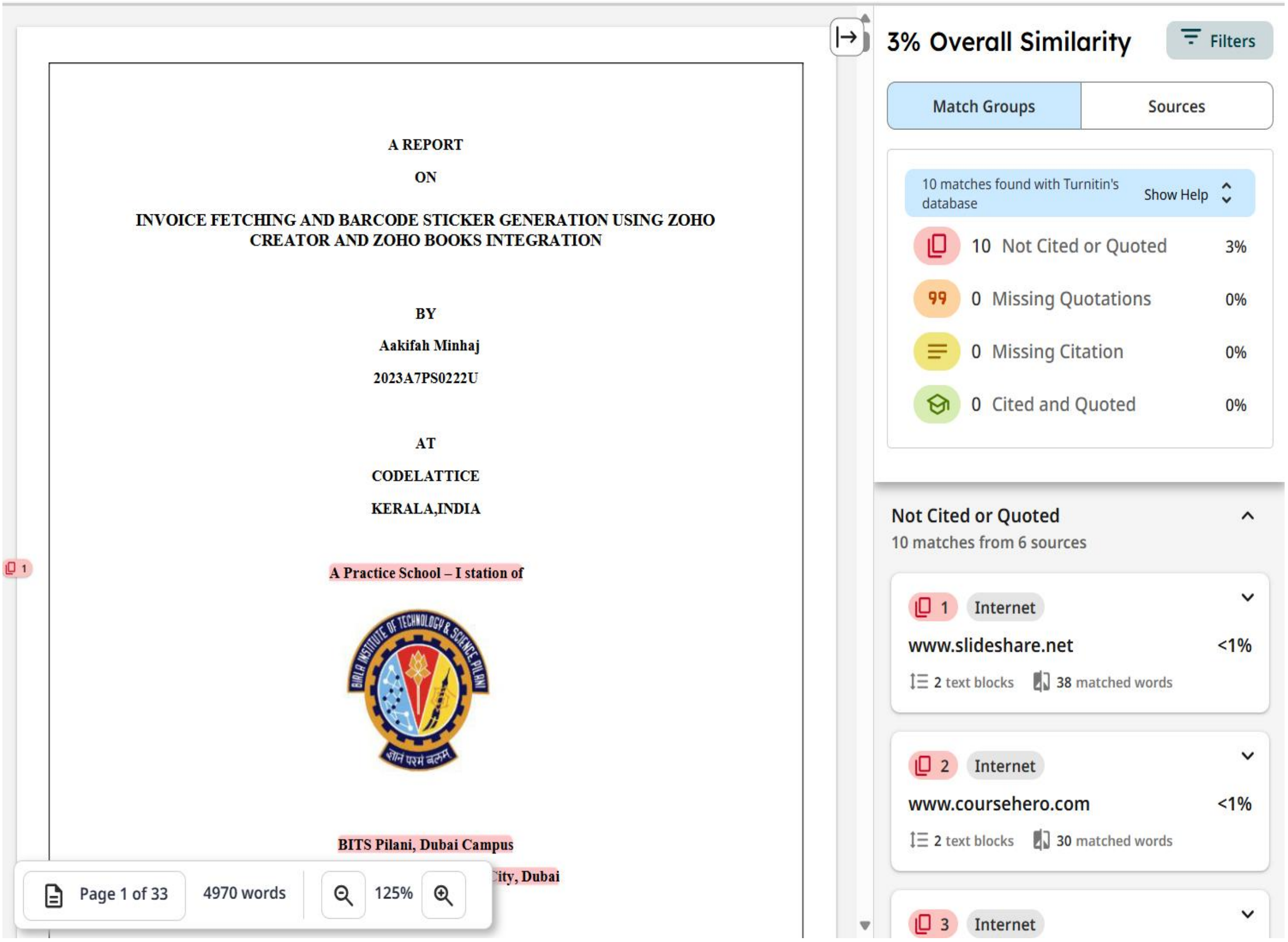
NOW VIEWING: HOME > PS I- SUMMER TERM 2024-25: SECTION 14 > PS - 1

About this page

This is your assignment dashboard. You can upload submissions for your assignment from here. When a submission has been processed you will be able to download a digital receipt, view any grades and similarity reports that have been made available by your instructor.

› PS - 1 ?

Paper Title	Uploaded	Grade	Similarity
PS 1 REPORT.pdf	07/30/2025 2:59 AM	--	🟩 3%



A REPORT

ON

INVOICE FETCHING AND BARCODE STICKER GENERATION USING ZOHO CREATOR AND ZOHO BOOKS INTEGRATION

BY

Aakifah Minhaj

2023A7PS0222U

AT

CODELATTICE

KERALA,INDIA

A Practice School – I station of

BITS Pilani, Dubai Campus

ity, Dubai

3% Overall Similarity

10 matches found with Turnitin's database

10	Not Cited or Quoted	3%
0	Missing Quotations	0%
0	Missing Citation	0%
0	Cited and Quoted	0%

Not Cited or Quoted

10 matches from 6 sources

1 Internet

www.slideshare.net <1%

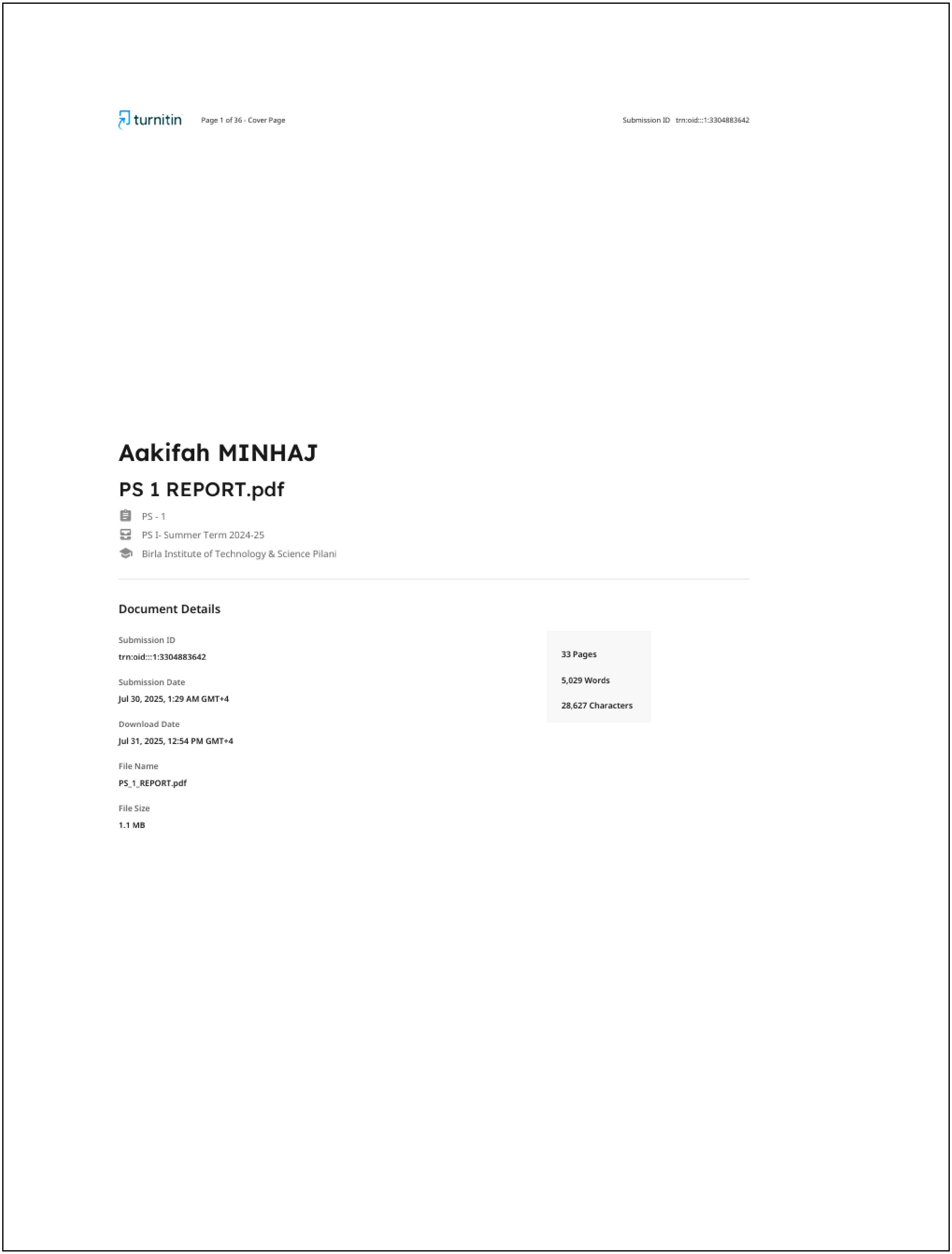
2 text blocks 38 matched words

2 Internet

www.coursehero.com <1%

2 text blocks 30 matched words

3 Internet



Aakifah MINHAJ

PS 1 REPORT.pdf

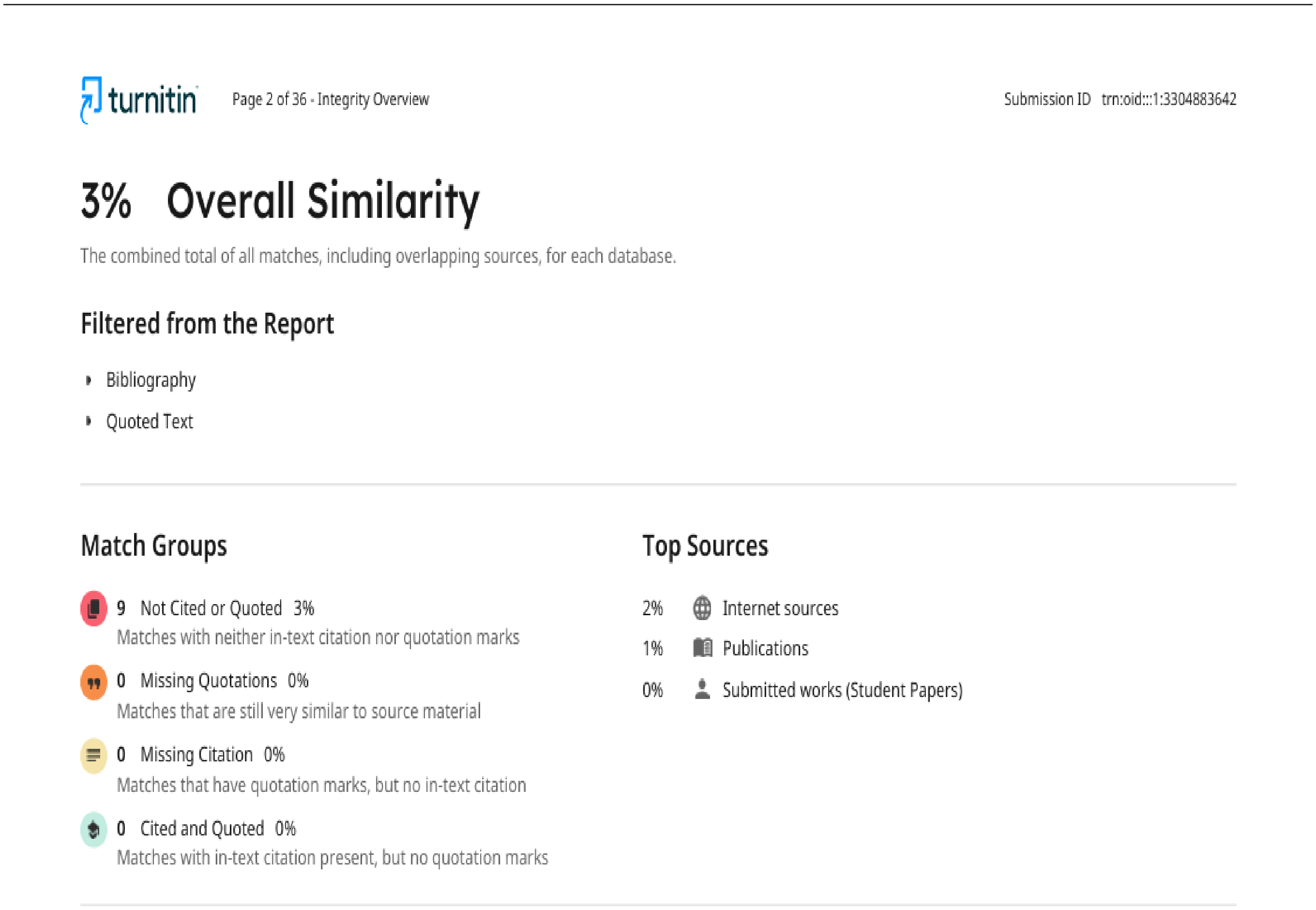
PS - 1

PS I- Summer Term 2024-25

Birla Institute of Technology & Science Pilani

Document Details

Submission ID trn:oid:::1:3304883642	33 Pages
Submission Date Jul 30, 2025, 1:29 AM GMT+4	5,029 Words
Download Date Jul 31, 2025, 12:54 PM GMT+4	28,627 Characters
File Name PS_1_REPORT.pdf	
File Size 1.1 MB	



turnitin Page 2 of 36 - Integrity Overview Submission ID trn:oid:::1:3304883642

3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

9 Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks

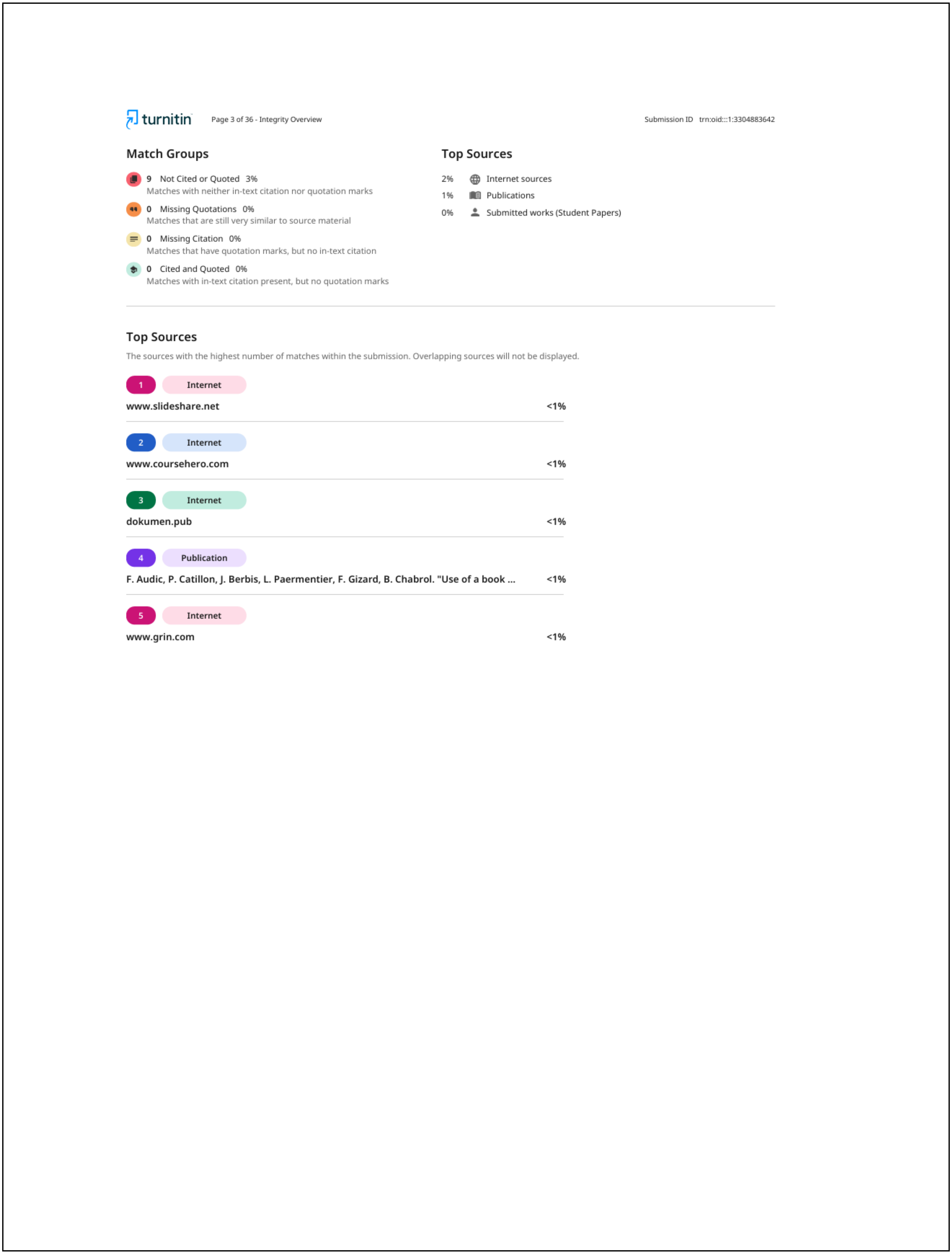
0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2%	Internet sources
1%	Publications
0%	Submitted works (Student Papers)



Match Groups

- 9 Not Cited or Quoted 3%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2%	Internet sources
1%	Publications
0%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.slideshare.net	<1%
2	Internet	www.coursehero.com	<1%
3	Internet	dokumen.pub	<1%
4	Publication	F. Audic, P. Catillon, J. Berbis, L. Paermentier, F. Gizard, B. Chabrol. "Use of a book ...	<1%
5	Internet	www.grin.com	<1%