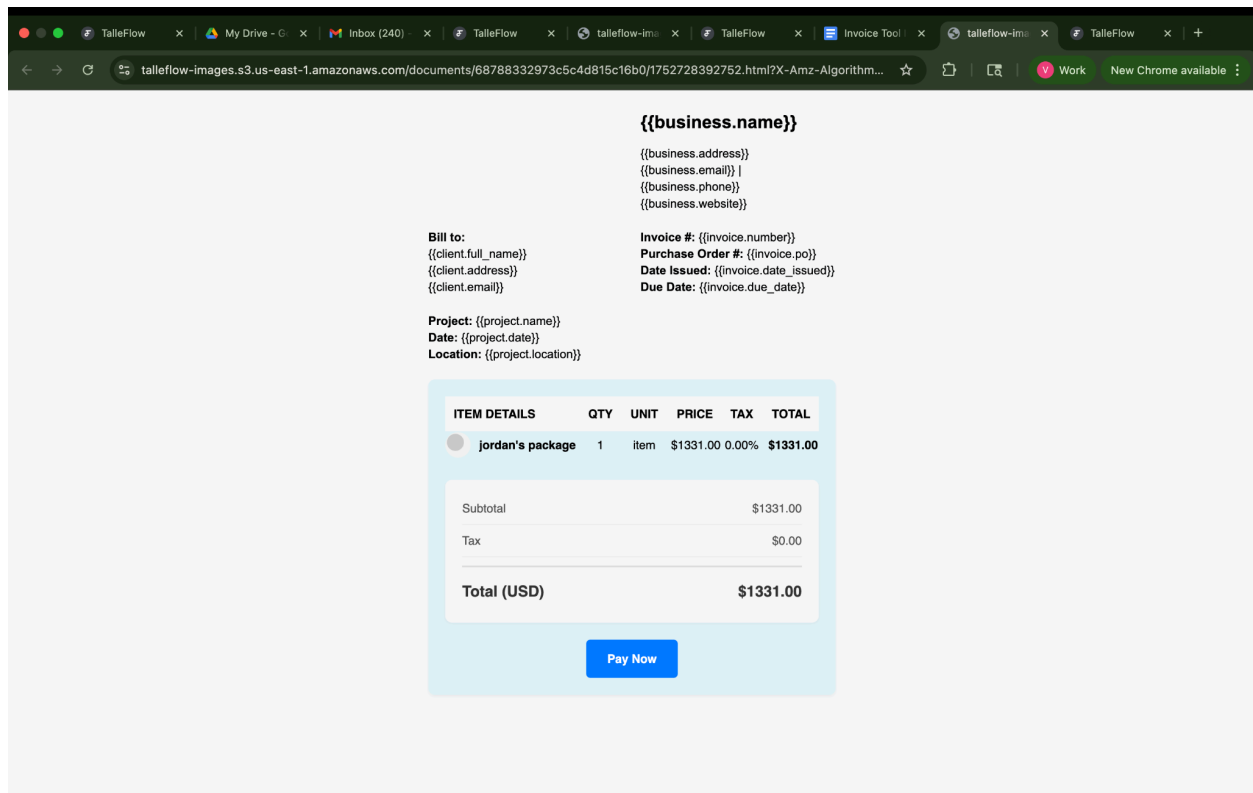**Issue:** Merge tags are not resolving in final invoice sent

Issue: Stripe payments are not going through from invoice
Description: After user pays via stripe link there is success message, but the payment is not going through to the backend and showing up in finance tab.
- Issue #1: Incorrect Stripe Account Lookup
    - Apply // WRONG: Looking for stripeAccountId on User model const user = await User.findById(invoice.user); const { stripeAccountId } = user; Problem: The code is trying to get stripeAccountId from the User model, but according to the migration notes and Business model, all Stripe fields have been moved to the Business model. Issue #2: Missing Business Lookup
    - The invoice has a businessId field, but the code completely ignores it and doesn't look up the business to get the correct Stripe Express account. Issue #3: No Customer Context
        - The checkout session is created without any customer information, which explains why you're seeing mysterious customer data in the details pane.
-

Issue: DocumentShare is not functioning properly

Description: sending documents through email is not functioning anymore. Documents can be shared via email and they will be sent through google client but that is no longer functioning. Affected area is DocumentShare file/methods

—--^^^New Issues ^^^—--

Issue: Saving invoice to project removes invoice items section
Description: after creating an invoice file and dragging the invoice items tool onto the page, I am able to successfully add my existing packages. Once I save the invoice to a project I am navigated to the project context editor but the edits I made to the document no longer appear any more.
Screenshot or Video: VIDEO LINK

Issue: Saving an invoice template with invoice items added removes invoice items
Description: after creating an invoice template and dragging the invoice items tool onto the page, I am able to successfully add my existing packages. Once I save the template and exit the editor, if I try and click on the saved template, the invoice items disappear.
Screenshot or Video: VIDEO LINK

Issue: Image of the package needs padding
Description: The image for the package needs padding on the unlayer editor preview
Screenshot or Video: VIDEO LINK

Issue: Columns for QTY, Price, Unit, Tax, Disc need better alignment and dynamic sizing
Description: For the invoice items tool In the unlayer editor, the columns and the column headers need better alignment, right now there is misalignment with the column headers and the values for each column. Additionally in Unlayer you can change the width of the body. There needs to be dynamic sizing enabled to allow for the column header and column values to remain aligned even if the user decides to change the body size.
Screenshot or Video: VIDEO LINK

Issue: Shared Invoice Document has incorrect formatting on mobile
Description: When viewing an invoice via mobile view, the invoice tool section is displaying formatting issues. Try sending an invoice to your own email address, then opening the link on your phone. This is important because most people use their phone to view these things.
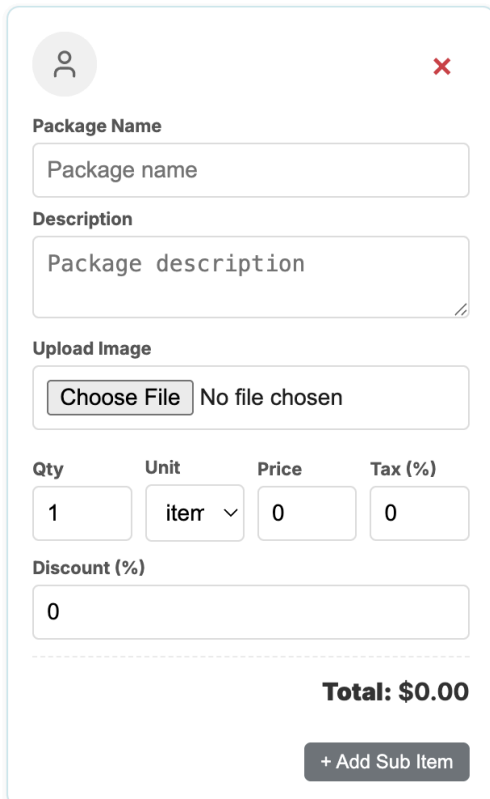**Screenshot or Video:**

Issue: Move image preview of package/service closer to "upload file" option
Description: As you can see in the screenshot the preview for the image is in a separate position from the "upload image" section. These 2 should be close together so the user understand that the preview is related to the call to action of "upload image"
Screenshot or Video:

Issue: Unresolved Merge tag modal
Description: When any merge tag is unresolved we should display a modal warning the user and which merge tags are unresolved. Also give them an explanation of what they should do to resolve the merge tag.
Screenshot or Video:

Issue: Rename package and sub item titles
Description: The naming conventions used for the invoice tool need to be redone. See video for explanation
Screenshot or Video: VIDEO LINK

Issue: Only allow invoice items to be available when creating a Proposal, Invoice, or Custom Document

Description: Invoice Items tool is available for all document types. It should only be available to use if creating an invoice or proposal or a custom document

Screenshot or Video:

—---------------------------—-----------DO NOT WORK ON YET—--

Issue: Sharing a document to contact directly shouldn't require user to save to a project, it should take them to contact sharing context first

Description: See video

Screenshot or Video: [VIDEO LINK](#)

Issue: Document sharing flow needs adjustment

Description: See the video this is a bit complex. The other important point I forgot to mention in this video which I also did not check, is that the modal to share the invoice after its been saved to a project should ask the user to add a contact if no contact is associated to it.

Screenshot or Video: [VIDEO LINK](#)

Issue: Incorrect file is being saved to project, this is due to file duplication issue.

Description:

Summary

The document sharing functionality (`POST /api/v1/documents/:id/share`) incorrectly creates duplicate Document records in the database instead of properly handling template-to-file conversion or reusing existing file documents.

Description

When sharing documents through the `shareDocument` function, the system always creates a new Document record regardless of whether the source is a template (`category: 'template'`) or an existing file (`category: 'file'`). This results in unnecessary duplicate Document records and bypasses the proper document lifecycle management.

Current Behavior:

1. Template Sharing: Creates a new Document with `category: 'file'` but bypasses the `template.createFile()` method
1. File Sharing: Creates a duplicate Document record instead of reusing the existing file

Expected Behavior:

1. Template Sharing: Use `template.createFile()` method to properly convert template to file and update usage statistics
1. File Sharing: Reuse the existing Document record without creating duplicates

Steps to Reproduce

1. Create a Document with `category: 'template'`
1. Call `POST /api/v1/documents/:templateId/share` with contact information
1. Observe that a new Document record is created instead of using `template.createFile()`
1. Create a Document with `category: 'file'`
1. Call `POST /api/v1/documents/:fileId/share` with contact information
1. Observe that another duplicate Document record is created

Impact

- Data Integrity: Unnecessary duplicate Document records in database
- Template Statistics: Template usage statistics not properly updated
- Storage Waste: Duplicate S3 uploads and database entries
- Relationship Loss: Missing `templateId` relationships for files created from templates
- Performance: Increased database size and query overhead

Root Cause

Located in `api/src/controllers/documentController.ts`, lines 569-578:

typescript

Apply to packageServi...

```
// PROBLEMATIC CODE - Always creates new Document

const newDoc = new Document({

  businessId: req.user?.businessId,

  fileName: fileName || 'Shared Document',

  type: type || 'file',

  category: 'file',  // Always creates category: 'file'

  uploadedBy: userId,

  content: {},

  metadata: metadata || {},
});

await newDoc.save(); // DUPLICATE DOCUMENT CREATED HERE
```

Affected Files

- `api/src/controllers/documentController.ts` (shareDocument function, lines 517-750)
- `api/src/routes/v1/documentRoutes.ts` (route: `POST /:id/share`)
- `api/src/models/DocumentShare.ts` (DocumentShare records reference wrong documents)

Technical Details

Current Flow Issues:

1. Function ignores the document ID parameter from the route (`/:id`)
1. Always creates new Document instead of checking existing document category
1. Doesn't use existing `template.createFile()` method for proper conversion
1. Missing template usage tracking updates

Expected Flow:

typescript

Apply to packageServi...

```javascript
// Pseudo-code for correct implementation

const existingDoc = await Document.findById(req.params.id);

let documentToShare;

if (existingDoc.category === 'template') {

  // Use proper template-to-file conversion

  documentToShare = await existingDoc.createFile(businessId, userId, fileName);

} else if (existingDoc.category === 'file') {

  // Reuse existing file document

  documentToShare = existingDoc;

}

// Then create DocumentShare records pointing to documentToShare
```

Acceptance Criteria

- [ ] Template sharing uses `template.createFile()` method
- [ ] File sharing reuses existing Document records
- [ ] No duplicate Document records created during sharing
- [ ] Template usage statistics properly updated
- [ ] DocumentShare records reference correct documents
- [ ] All existing sharing functionality preserved
- [ ] S3 uploads only happen once per unique document

Test Cases

1. Template Sharing Test:
   - Share a template → Verify file created via `createFile()` method
   - Check template `timesUsedAsTemplate` incremented
   - Verify DocumentShare references the new file, not template
1. File Sharing Test:
   - Share an existing file → Verify no duplicate Document created
   - Verify DocumentShare references original file
1. Multiple Shares Test:
   - Share same document multiple times → Verify only DocumentShare records multiply, not Documents

Related Issues

- Template usage analytics may be inaccurate due to bypassed `createFile()` method
- Database cleanup may be needed for existing duplicate records

Priority Justification

High Priority due to:

- Data integrity issues
- Storage waste and performance impact
- Incorrect business logic for template-to-file conversion
- Potential confusion in document management workflows

1 of 6 To-dos

Completed

DocumentShare.ts

documentRoutes.ts