

# Neatly. An agentic email assistant for Investment Banking

**Report by:**

Andrei Rizea

**Student ID:**

220300153

**Word count:**

3938 (Pages 4-27)

**Organiser:**

Jonathan Jackson

**Module:**

IOT653U. Strategic Digital Leadership: Mastering Projects, Programs, and Products

**Assessment:**

002 Individual Coursework

**Live app:**

<https://neatly-silk.vercel.app>

**Video demo:**

 Demo of Neatly

# Executive summary

This report outlines Neatly, an AI-powered agentic email assistant designed specifically for investment banking (IB) analysts. Bankers face significant email overload, managing high volumes of time-sensitive messaging across their 70-100 hour working weeks. Neatly addresses this by analysing email threads to extract action items, deadlines, and stakeholder roles, while automatically researching complex topics to provide users with a concise and comprehensive overview.

Neatly began with an overly ambitious scope but was refined following assessment feedback to focus on one high-value feature with a stronger value proposition. Agentic research coupled with actionable thread analysis was chosen as the core. Built using NextJS, Supabase, and OpenAI's AgentKit, the proof of concept (POC) demonstrates the technical feasibility, delivering a responsive, secure, and scalable solution.

With the POC complete, the next steps are onboarding early users, refining the product, and preparing for growth through hiring and delegation. Neatly is positioned as a vertically aligned software-as-a-service (SaaS) business, enabling it to be an expert in the IB field rather than a horizontally aligned generic platform. It will follow an enterprise SaaS model charging £45/seat/month and is estimated to generate a net benefit of ~£500/analyst/month assuming a normal contractual 45 hour work week.<sup>§</sup>

# Contents

Executive summary.....	1
Contents.....	2
Introduction and background.....	4
Proposed solution.....	5
Overview and scope.....	5
Solution refinement.....	5
Requirements.....	6
User stories.....	6
System Architecture.....	6
AI vs Agentic AI.....	7
Data Design.....	7
Data flow.....	9
App design.....	10
Success Criteria.....	10
Business case.....	11
Value proposition.....	11
Value realisation.....	11
Cost justifications.....	12
Alternative solutions.....	12
External and Internal factors.....	13
Proof of concept.....	14
Justification of Key Design Decisions.....	14
Full user-flow.....	15
Agentic workflow implementation.....	16
Cookies and JWTs.....	18
Iteration and improvements.....	18
Version control.....	20
Code development.....	20
Task trackers.....	21
Security risk assessment.....	22
Assessment.....	22
Severity evaluation.....	22
Mitigation.....	23
Conclusion.....	26
Next steps.....	26
Critical review of presentation feedback, reflections and KSBs.....	27
References.....	28
Appendix.....	31
Appendix 1 - Neatly V1 application.....	31
Appendix 2 - MoSCoW prioritised requirements.....	35
Functional Requirements.....	35
Non-Functional Requirements.....	36

Appendix 3 - Neatly Data schema diagram.....	37
Appendix 4: Neatly V1.5 Wireframes.....	38
Appendix 5: Success criteria.....	41
Appendix 6: Cost justification calculations.....	42
Current.....	42
Proposed.....	42
Appendix 7 - Initial resources and personnel costs.....	43
Appendix 8 - Post launch resources and personnel costs.....	43
Appendix 9 - External Factors.....	44
Appendix 10 - Internal Factors.....	45
Appendix 11 - Neatly V2 POC screenshots and video.....	46
Landing Page.....	46
Auth.....	46
Home page.....	47
Email thread analysis home page.....	47
Researched topic page.....	48
Thread analysis items.....	48
Calendar.....	49
To-do.....	49
History.....	50
Dark mode.....	50
Appendix 12 - Justification for key design decisions.....	51
Appendix 13 - Content research agent definition.....	52
Appendix 14 - Content formatting agent definition.....	53
Appendix 15 - Binary Risk Analysis.....	54
Reference risk assessment.....	54
Binary Risk Analysis worksheet.....	54
Binary Risk Analysis worksheet process.....	55
2x2 outputs.....	55
3x3 outputs.....	55
Likelihood matrices.....	55
Impact matrices.....	55
Binary Risk Analysis Questions.....	56

# Introduction and background

Email overload is becoming a productivity challenge in investment banking (IB), with professionals overwhelmed with constant communication and fragmented workflows. According to McKinsey Global Institute (2012), on average, workers spend up to 28% of their week (~13 hours out of a 45-hour week) reading and responding to email (Dabbish & Kraut, 2010). Microsoft's Worklab (2025) states that employees now receive an average of 117 emails and 153 instant messages daily and 57% of the working day is consumed by communication activities rather than substantial work. Microsoft refers to this as "unprioritized chaos". Mailbird (2025) also supports this with an average of 121 a day too and Bouvier et al. (2025) even states this "cripples users" requiring huge time commitments.

As well as time, emails tend to force significant context switching. University of California, Irvine, research found that workers, on average, need 23 minutes and 15 seconds to entirely refocus after an interruption (Mark, Judith and Klocke, 2008). In addition, Harvard Business Review also found that office workers toggle between applications ~1,200 times a day causing an estimated four hours per week lost on these transitions (Murty, Dadlani and Das, 2022). In IB specifically, bankers manage multiple large deals simultaneously. This means transitions and lack of focus compounds significantly. Not only this but emails typically function as informal task management systems. This is where deadlines, responsibilities, and decisions remain hidden in unstructured email threads, making efficient tracking difficult.

The intensity of IB increases these challenges. DigitalDefynd (2025) found that analysts averaged 72 hours per week with peaks of 90-100 hours. BBC (2021) also revealed that 77% of junior Goldman Sachs (GS) analysts reported experiencing "workplace abuse" through constant overwork and what Wall Street Oasis (2025) deems as "unrealistic deadlines". JPMorgan and Bank of America have since implemented 80-hour weekly caps (AlphaSense, 2024). However, email volumes remain the same with banks processing ~10-50+ million emails annually on average (Bachan, 2024). Bank of America (2025) also states that AI adoption in the industry can save thousands of hours a year. A survey found that bankers also spend up to 40 hours weekly on manual admin tasks, with 72% considering leaving the industry due to burnout (Villemonteix, n.d.). Deals involve many stakeholders over many email chains and various companies. Missing important information or not looping in appropriate parties not only delays deals, but can cause reputational damage (MadeMarket, 2025).

Enterprise AI is increasingly being applied to various workflows. For example, Microsoft's Copilot can summarise long email threads (Microsoft, 2025), proving that summarisation is technically feasible at scale. Survey data now indicates that 75% of professionals now want AI assistants for email management and task automation (Lewis, 2024). However, summarisation alone may not fully solve the execution problem in IB contexts. Users need actionable, clear tasks, deadlines, and key decisions rather than novel summaries. This gap motivates a focused investigation into how AI-driven extraction and structuring of actionable information from email threads could reduce overload and improve productivity in IB environments.

# Proposed solution

## Overview and scope

Neatly is an AI-powered assistant designed to address email overload in IB where existing tools like Microsoft Copilot mainly provide summarisation. Neatly focuses on extraction and workflows. It transforms email content into structured and actionable outputs. For each thread, Neatly produces a concise summary, structured tasks with deadlines, assigned stakeholders, and provides contextual research on key topics. Since the initial proposal, it was found that Google requires formal application approval for production connectivity which could take a few weeks. This would delay the project so Neatly now uses a copy-paste input workflow rather than direct email integration.

## Solution refinement

The initial proposal positioned Neatly as a horizontally-aligned mass-market assistant encompassing numerous features, shown in Figure 1. Feedback from the presentation stated that this reduced the value proposition and a unique-selling-point (USP) was not clear. Established competitors already offer these general-purpose AI email tools. Hence, Neatly was repositioned around a narrower focus of supporting IB specifically. It would target the manual workflows that currently consume up to 40 hours weekly (Villemonteix, n.d.). By concentrating on structured task extraction rather than general summarisation, Neatly addresses a gap that existing tools do not fully resolve. It will also be positioned as an industry expert creating the competitive advantage and USP.

	Abnormal	Clean Email	Fyxer AI	SuperHuman	Google / Outlook / Other providers	<b>NEATLY</b>
<b>Inbox Zero</b>						
<b>Phising + Spam detection</b>	✓	✗	✓	✓	✓	✓
<b>Email grouping and filtering</b>	✗	✓	✓	✓	✓	✓
<b>Daily digests</b>	✗	✓	✓	✗	✗	✓
<b>Auto Email drafting</b>	✗	✗	✓	✓	✗	✓
<b>Automations</b>						
<b>Agentic Workflow Capabilities</b>	✗	✗	✗	Only Calendar	✗	Calender, Research, Follow-ups
<b>Industry Specific</b>						
<b>Industry</b>	Any	Any	Any	Any	Any	Any

Figure 1: Neatly V1 competitor analysis

## Requirements

Neatly's features and requirements were prioritised using MoSCoW. Core features include email thread input (FR1), stakeholder and action-item extraction (FR2), content research (FR3–FR4), to-do and calendar integrations (FR7–FR8). Supporting these are non-functional requirements like fast analysis performance (NFR1), strict data privacy rules (NFR3), and stable uptime (NFR4). Combined, these ensure the app remains fast, secure, and aligned with real pain points. Full outlines of both are available in Appendix 2.

## User stories

The following user stories have been prioritised to create guidelines for what the outcome of the project should be:

1. As a busy IB analyst, I want to have action items and deadlines extracted for me so that I can track responsibilities easily without forgetting or reading long email threads.
2. As a banker, I want the app to identify stakeholders and their roles, so that I know who is involved and who to reach out to.
3. As a banker, I want to see a research summary of businesses, topics and complex workflows in the thread, so that I can quickly get up to speed.

## System Architecture

The proposed architecture is a web-app built with NextJS and hosted on Vercel. It will store data in Supabase, which also provides authentication in one centralised platform. OpenAI will then be used for the AI layers due to its performance.

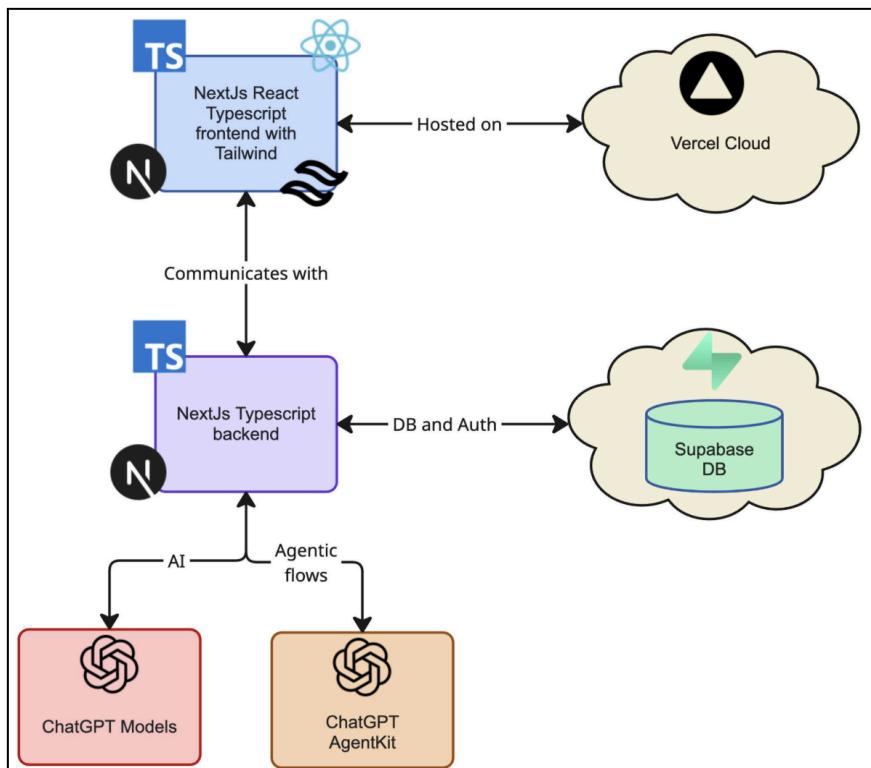


Figure 2: System architecture diagram

## AI vs Agentic AI

Differentiating these terms is important for understanding Neatly's architecture. AI is the overarching term for systems that can analyse, generate, and classify when an input is provided. Neatly uses AI to analyse email threads and extract structured details. Agentic AI builds on this foundation by adding planning capabilities and the ability for continuous execution towards a goal. In Neatly, this appears in the research flow. Agents are assigned a research topic, then autonomously search, evaluate sources, and collate information in an iterative approach rather than producing a one-off response.

## Data Design

Neatly uses PostgreSQL. For optimum performance, relational database practices like data normalisation should be incorporated to prevent duplication. This is achieved by splitting data into logically related tables, connected through primary and foreign keys. The data model centres on email threads as the parent entity. One user can have many threads analysed, and each thread generates both AI analyses and agentic research outputs. Within analyses, the system returns multiple item types including deadlines, tasks, and stakeholders, each stored in separate normalised tables to maintain referential integrity and enable flexible querying. Figure 3 shows the entity-relationship (ER) diagram for Neatly, with the complete schema available in Appendix 3.

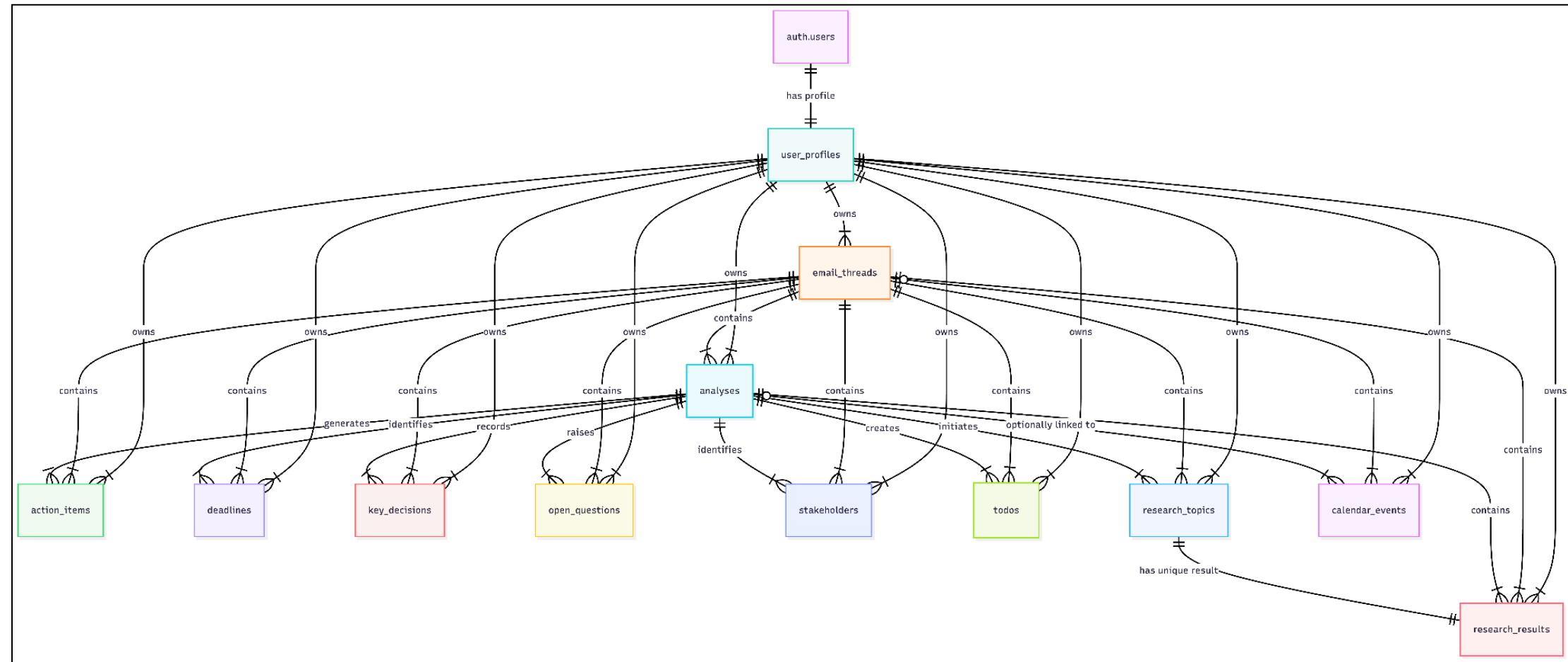


Figure 3: Entity relationship diagram

## Data flow

At a high level, Neatly follows a simple pipeline with an abstracted diagram below too:

1. User pastes a thread
2. The backend triggers
  - a. AI analysis to identify relevant data
  - b. Agentic research flow to search the web for information
  - c. Then returns data to the user

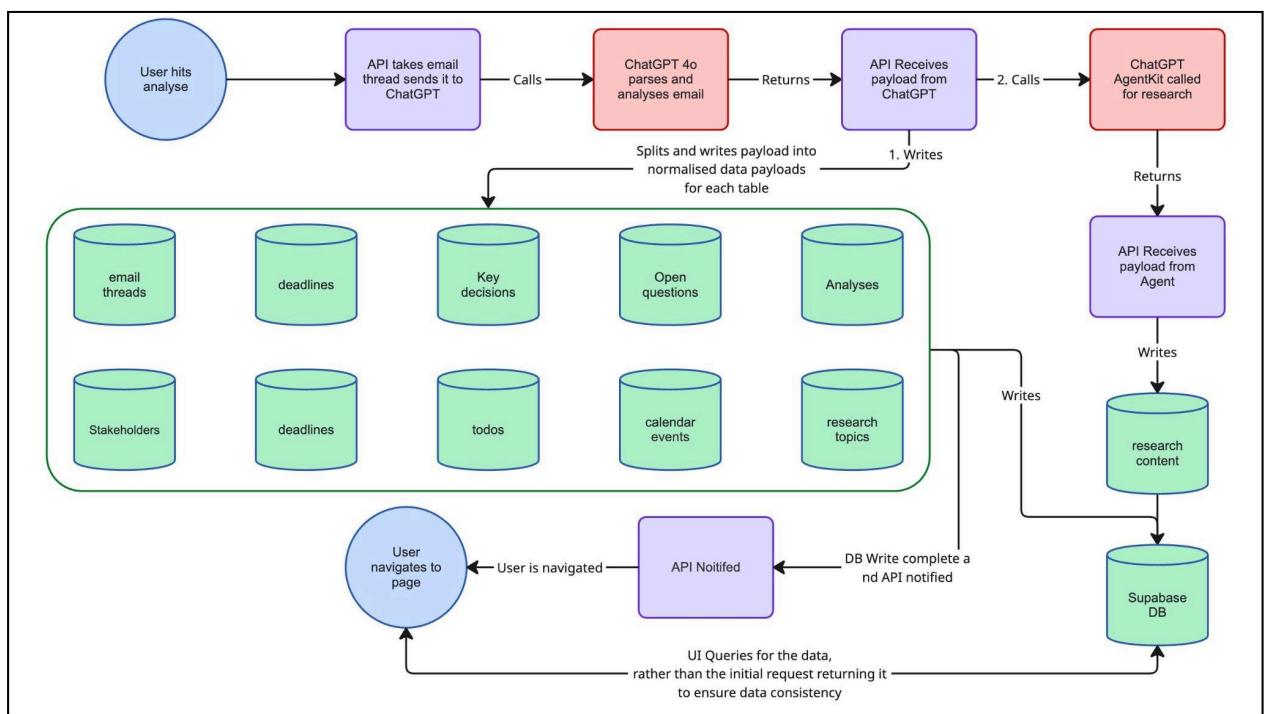


Figure 4: Data flow

## App design

To better visualise Neatly, 6 wireframes were designed through Loveable (an AI app builder). This helped provide an understanding of how the application could be presented to ensure an intuitive application. It can be viewed here: <https://neatly-app-sdl.lovable.app>. Below is just the analysis page to paint an initial picture of what the app would cover. The remaining items are in Appendix 4.

The screenshot displays the Neatly V1.5 wireframe interface. At the top, there is a navigation bar with tabs for 'Threads' (which is selected), '+ New', 'Calendar', 'To-Do', 'History', and 'Settings'. Below the navigation bar, the main content area has two main sections: 'Thread Messages' on the left and 'Research Brief' on the right.

**Thread Messages:** This section shows an email from Mike Johnson dated January 3, 2025 at 2:00 PM. The subject is 'Project kickoff meeting - Action items'. The message body contains a list of action items for the team, including tasks for Lisa, Tom, and Everyone, along with a note about the beta launch deadline.

**Research Brief:** This section shows a summary of the research brief. It includes a success message ('Brief generated successfully'), a stakeholder count (0), an action item count (1), a deadline count (0), an open question count (1), and a risk flag count (0). A button at the bottom right of this panel says 'Open Full Brief'.

Figures 5: Neatly V1.5 wireframe designs

## Success Criteria

To evaluate Neatly, success criteria were defined across five metrics. Accurate extraction is given a target value of 90% accuracy, which reflects performance achieved by modern document AI pipelines (Laurent, 2025). Loading times are also important with a target value of 5-20 seconds which aligns with Nielsen's (1993) research establishing 10 seconds as the threshold before user attention drifts. Full success criteria with justifications is in Appendix 5.

## Business case

In IB, emails are crucial for deal execution given the industry is built around communication. Junior bankers are always tasked with high workloads with 70+ hour weeks on average (Wall Street Oasis 2025). Hence, reducing low-value emails can directly increase the amount of work bankers can do. So, an AI email assistant that summarises threads, extracts action items, drafts replies, and adds contextual research is proposed to solve this.

## Value proposition

1. Efficiency: Reduces bankers' time spent on low-value emails shifting capacity and possibly increasing productivity by up to 35% (Deloitte, 2023).
2. Quality/Risk: Fewer missed tasks and deadlines through automated task tracking.
3. Preparedness: Contextual research supports quicker and more informed decisions allowing bankers to save thousands of hours a year (Bank of America, 2025).

## Value realisation

Within IB, the value of time is real but difficult to attribute and quantify for a single tool. Hence any dollar figure would be speculative. Therefore, productivity measures will be used to quantify the value. They are directly observable, attributable to the tool, measurable and will follow Sauermann's (2016) productivity-measures guidance. Below key-performance-indicator (KPI) table covers these.

KPI	Expected threshold
Avg daily time spent on email per user	Down by up to 25% as per (Deloitte, 2023)
% emails requiring manual full-read vs only summary	> 60% assuming majority of emails are low-value
Overdue action items per user per week	Down by > 50% with Neatly
Missed deadline / missed follow-up incidents	Down by > 50% due to task management

Table 1: KPIs

## Cost justifications

Neatly follows a SaaS enterprise pricing model at £45/seat/month. Knowing that 28% of a work week is spent on email (Mailbird, 2025), bankers effectively spend ~£1,664/month managing their inbox taking into consideration their average annual salary. Applying a 35% productivity improvement from AI adoption (Deloitte, 2023) results in savings of ~£564/banker/month. This justifies the £45/month subscription, delivering a net benefit exceeding £500/month per seat.

Full calculations are available in Appendix 6.

## Resources

During the POC stage, rapid prototyping is important to launch quickly. Using AI assistants allows for in-hour development removing external labour costs. Initial costs only include the tech such as Supabase, Cursor and AI tokens totalling < £57/m.

After the launch, focus shifts into scaling and growing, requiring more personnel. Specialised team-members are required for sales, support and developers. This costs roughly £15,500/m plus an additional £1,620/m for the tech subscriptions. Break-even requires over 350 active seats across enterprise plans which is an achievable target for the growth stage.

Full breakdowns are available in Appendix 7-8.

## Alternative solutions

### **Option A:**

Outlook/Gmail. These have AI solutions such as Copilot which can be coupled with additional basic email rules and manual tracking. Cheap but it lacks focused workflows. Rules do not reliably extract tasks and users are still prone to forgetting tasks.

### **Option B:**

Horizontally aligned “off-the-shelf” software which supports some filtering flows, but does not provide agentic workflows.

### **Option C:**

Neatly, a vertically aligned tool to IB. Provides accurate tasks and deadlines creation, structured digests, optional response drafting and more. Importantly, it provides contextual research improving banker meeting preparedness.

## External and Internal factors

Managing both external and internal factors is important for a successful project. PESTEL analysis was used to identify the key external factors. IB messaging requires strict compliance with GDPR and Material-Non-Public-Information (MNPI) legislation. Non-compliance will have legal implications as banks operate under some of the strictest data laws (GDPR Regulation, 2025). Technologically, hallucinations are a big risk. Chelli et al. (2024) conducted research that demonstrated hallucination rates of ~28.6% for GPT-4 on retrieval tasks which would reduce accuracy and cause trust issues. Furthermore, trust remains the main blocker to AI adoption in finance. Deloitte (2025) stated that 21.3% of workers list trust as their primary concern with AI. Data encryption, structured outputs for validation, and human-in-the-loop verifications are all mitigation steps that will be introduced.

Internally, SWOT was utilised. This showed that a lack of AI knowledge within the team increases quality risks for the agentic workflow to be built. Slow iteration also creates an opportunity cost as larger and more equipped competitors may capture the market with their brand trust. Mitigations include maintaining a tight scope to ensure rapid iteration, establishing standard-operating-procedures, and remaining vertically aligned with IB.

Full analysis is available in Appendix 9-10.

# Proof of concept

The POC is a full end-to-end web application built with scalable, community-focused technology. It has user authentication, database and AI integrations. Email-provider integration was excluded and instead a simple copy-and-paste interface was implemented avoiding approvals whilst maintaining core functionality.

Figure 6 is a screenshot of an analysis page where the user can click into a specific topic or look at the thread breakdown. All screenshots are in Appendix 11. The app is also available to test here: <https://neatly-silk.vercel.app/> or a demo is available here: [Demo of Neatly](#).

Figure 6: Individual thread analysis page

## Justification of Key Design Decisions

Full justifications are explained in Appendix 12. Below is a summarised overview containing the key drivers:

1. NextJS was chosen due to its integrated UI and API routes removing the need for a separate API project.
2. Vercel was chosen as it has one-click deployments via GitHub and great community support.
3. Supabase combines the power of Postgres such as row-level-security with authentication into a managed platform.
4. ChatGPT is the standard for top-tier AI models (Vellum AI, 2025) and has recently launched AgentKit allowing for custom agentic flows to be built.

## Full user-flow

Figure 7 is a walkthrough of the user and data flow explaining how user interactions (blue) connect to APIs (purple) which then connect to AI (orange) and agentic AI (red) and then how it writes to Supabase (green).

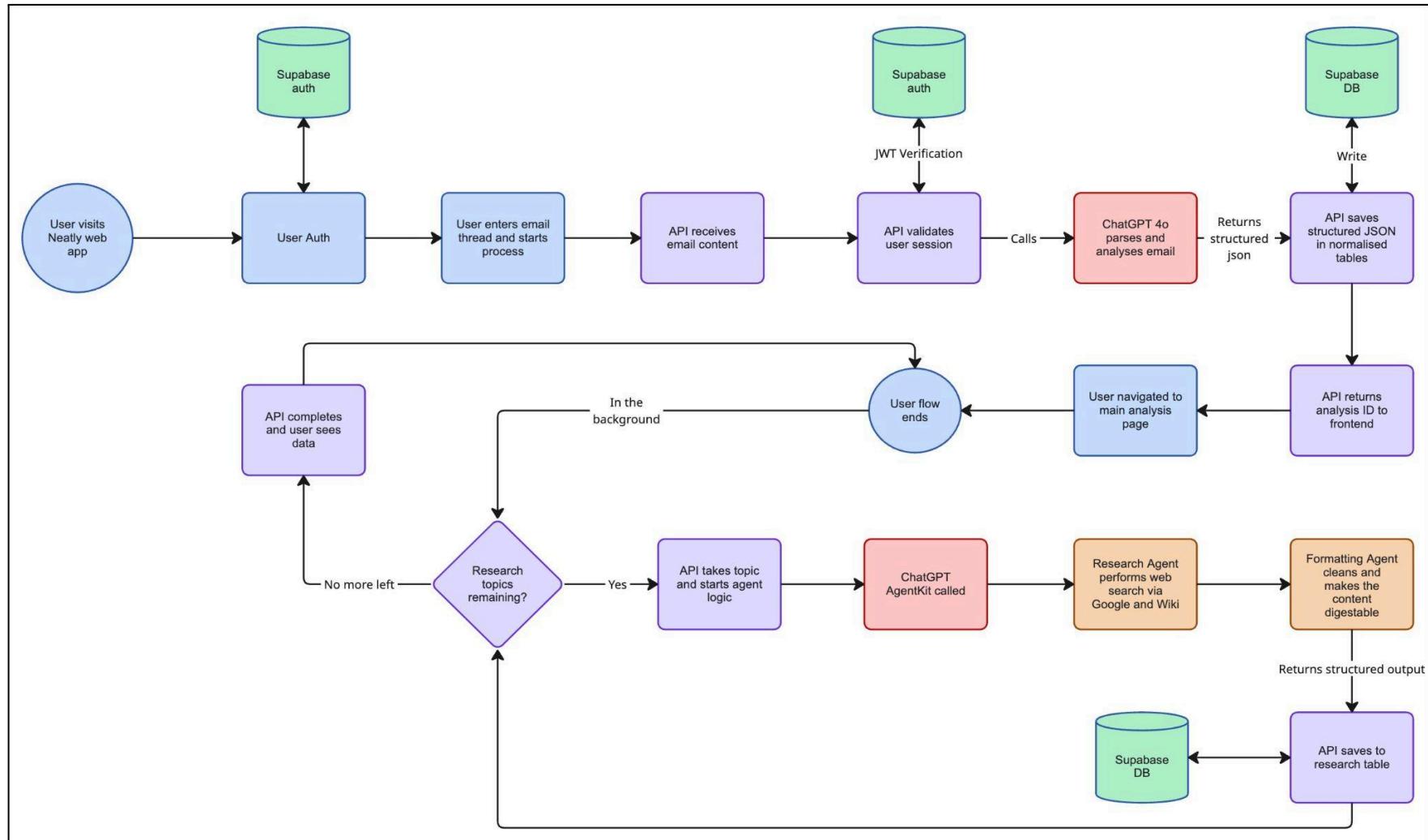


Figure 7: Full user flow diagram

## Agentic workflow implementation

AgentKit from OpenAI was used to build out the research flow. A content researcher and formatter agent were created. This split ensures each agent only focuses on one responsibility ensuring it can produce high-quality outputs.

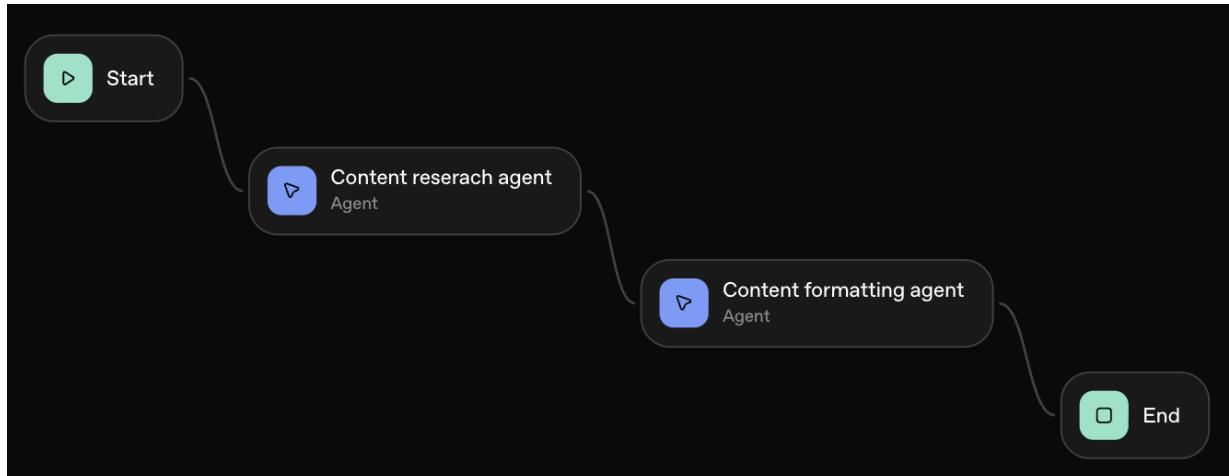


Figure 8: AgentKit screenshot of Neatly agent

The research agent has been set to use OpenAI's web-search tool. This allows it to query the internet for relevant data and provide sources to ensure validity following human-in-the-loop best-practices. Figure 9 is the configuration for the web search set to Google and Wikipedia for now.

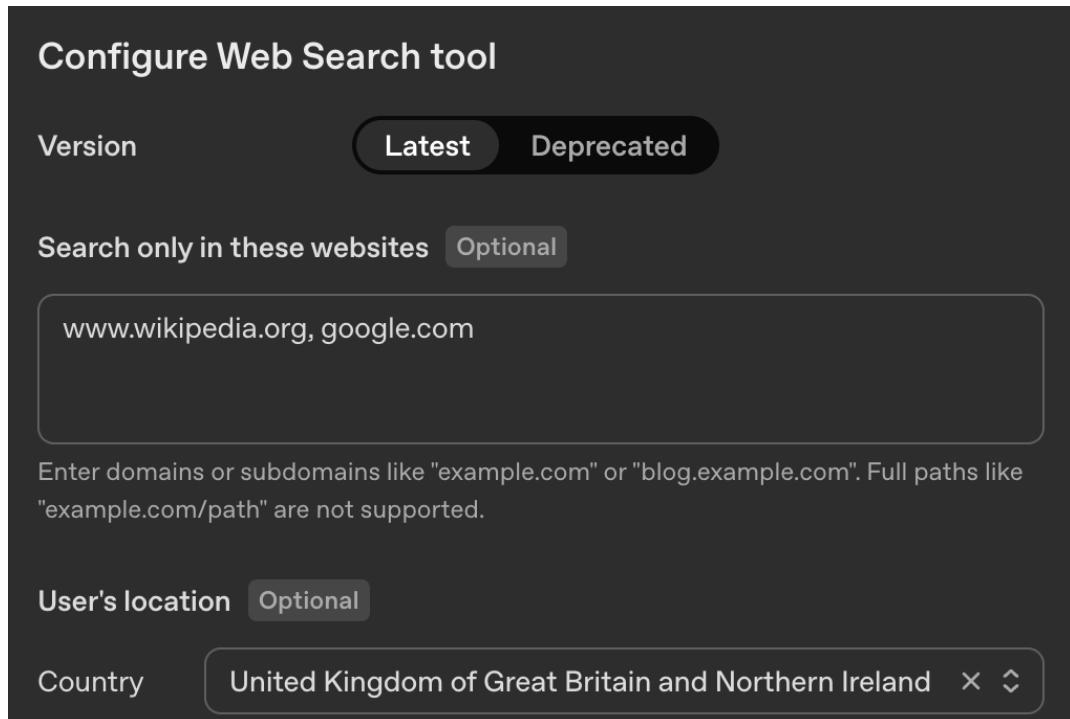


Figure 9: AgentKit research agent web search tool configuration

There are 2 main methods of calling this workflow. ChatKit and AgentsSDK. ChatKit is used if you have a chat bot style app where the user converses with a bot. Neatly does not have this and it needs programmatic calls. Hence, AgentsSDK will be used. Figure 10 is the entry point for the agent in the NextJS backend controller.

```
// main code entrypoint
export const runWorkflow = async (workflow: WorkflowInput) => {

  const conversationHistory: AgentInputItem[] = [
    { role: "user", content: [{ type: "input_text", text: workflow.input_as_text }] }
  ];
  const runner = new Runner();

  const contentReserachAgentResultTemp = await runner.run(
    contentReserachAgent,
    [...conversationHistory]
  );

  conversationHistory.push(...contentReserachAgentResultTemp.newItems.map((item) => item.rawItem));

  if (!contentReserachAgentResultTemp.finalOutput) {
    throw new Error("Research Agent result is undefined");
  }

  // the formatting agent takes the research history + its instructions to formatted output
  const contentFormattingAgentResultTemp = await runner.run(
    contentFormattingAgent,
    [...conversationHistory]
  );

  conversationHistory.push(...contentFormattingAgentResultTemp.newItems.map((item) => item.rawItem));

  if (!contentFormattingAgentResultTemp.finalOutput) {
    throw new Error("Formatting Agent result is undefined");
  }

  // contentFormattingAgentResultTemp.finalOutput is parsed by Zod/SDK
  return contentFormattingAgentResultTemp.finalOutput;
}
```

Figure 10: Code entry point for the agent

This essentially utilises the “Runner” class from "@openai/agents" to trigger the agent definitions. These definitions take in system prompts and other configs such as parameters and tools. The full agent definitions can be found in Appendix 13-14.

## Cookies and JWTs

Every request header is populated with JSON-web-tokens (JWTs) to ensure API authorisation, preventing unwanted endpoint calls. When users log in, Supabase stores the JWT in cookies. On each request, the browser sends these cookies. The middleware files (the bridge between the UI and API) read cookies via getAll(), pass them to createServerClient, which validates the JWT. API routes then call supabase.auth.getUser() to verify the cookie.

```
const supabase = createServerClient(
  process.env.NEXT_PUBLIC_SUPABASE_URL!,
  process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!,
  {
    cookies: [
      getAll() {
        return request.cookies.getAll()
      },
      setAll(cookiesToSet) {
        cookiesToSet.forEach(({ name, value }) => request.cookies.set(name, value))
        supabaseResponse = NextResponse.next({
          request,
        })
        cookiesToSet.forEach(({ name, value, options }) => supabaseResponse.cookies.set(name, value, options))
      },
    ],
  },
)
```

Figure 11: Code for JWT cookies

## Iteration and improvements

During testing, a bottleneck appeared. A long delay was visible when an analysis was started because it was waiting for all calls to return. This wasn't ideal as the UI would be stuck. So, loading is now shown only until the initial analysis returns which populates all but the research. As the agents in the background complete, they write to the database and the UI will update asynchronously. This reduces initial cumbersome 1-3 minute loading times to quick 10-20 second initial loads. Below is an overview of the change in flow improvement.

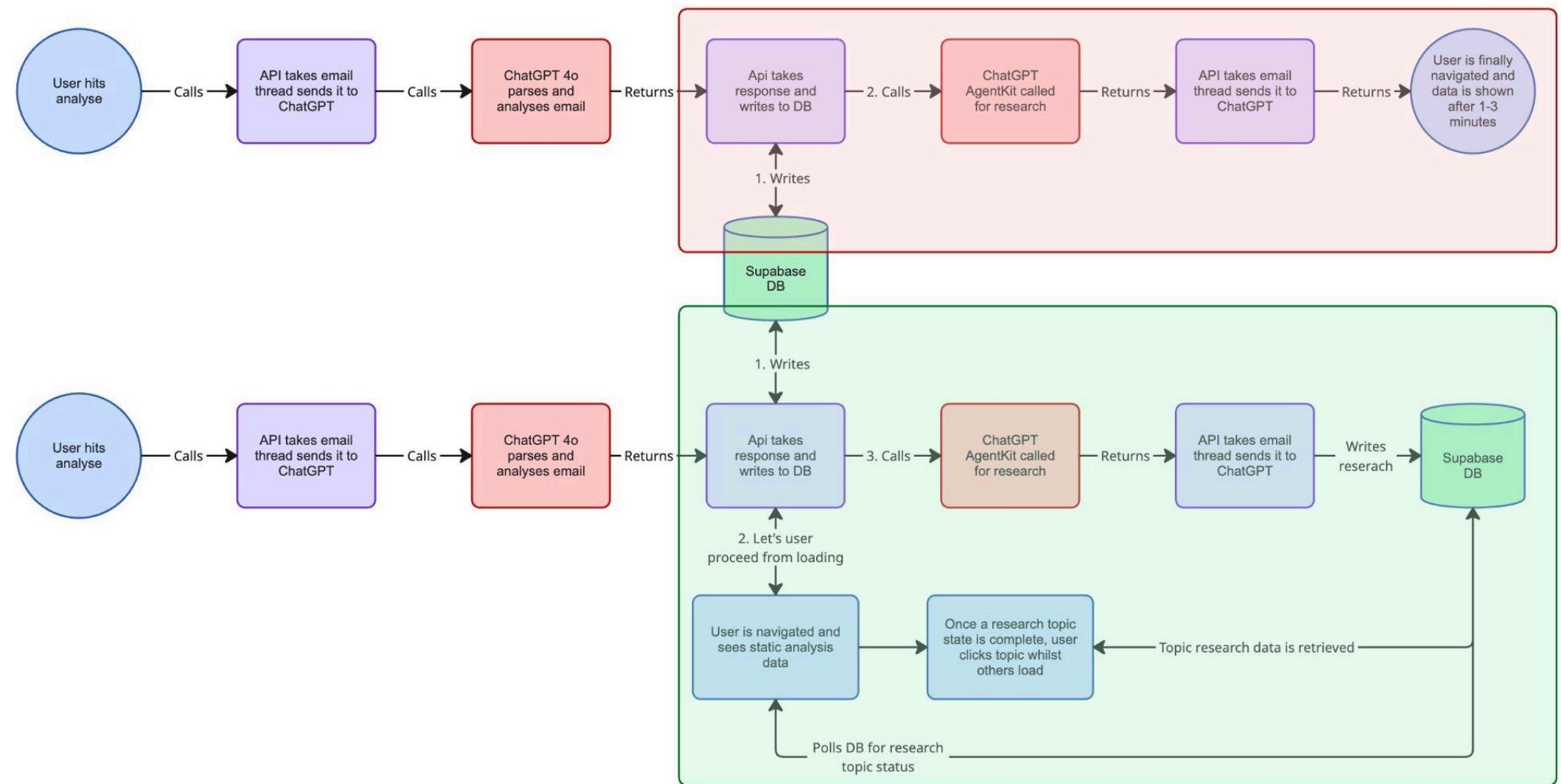


Figure 12: Backend redesign explained. The old section is in red, new in green

## Version control

GitHub was used throughout the implementation for version control, enabling progress tracking and rollbacks. Using branches, it was easy to decompose and manage separate tasks.

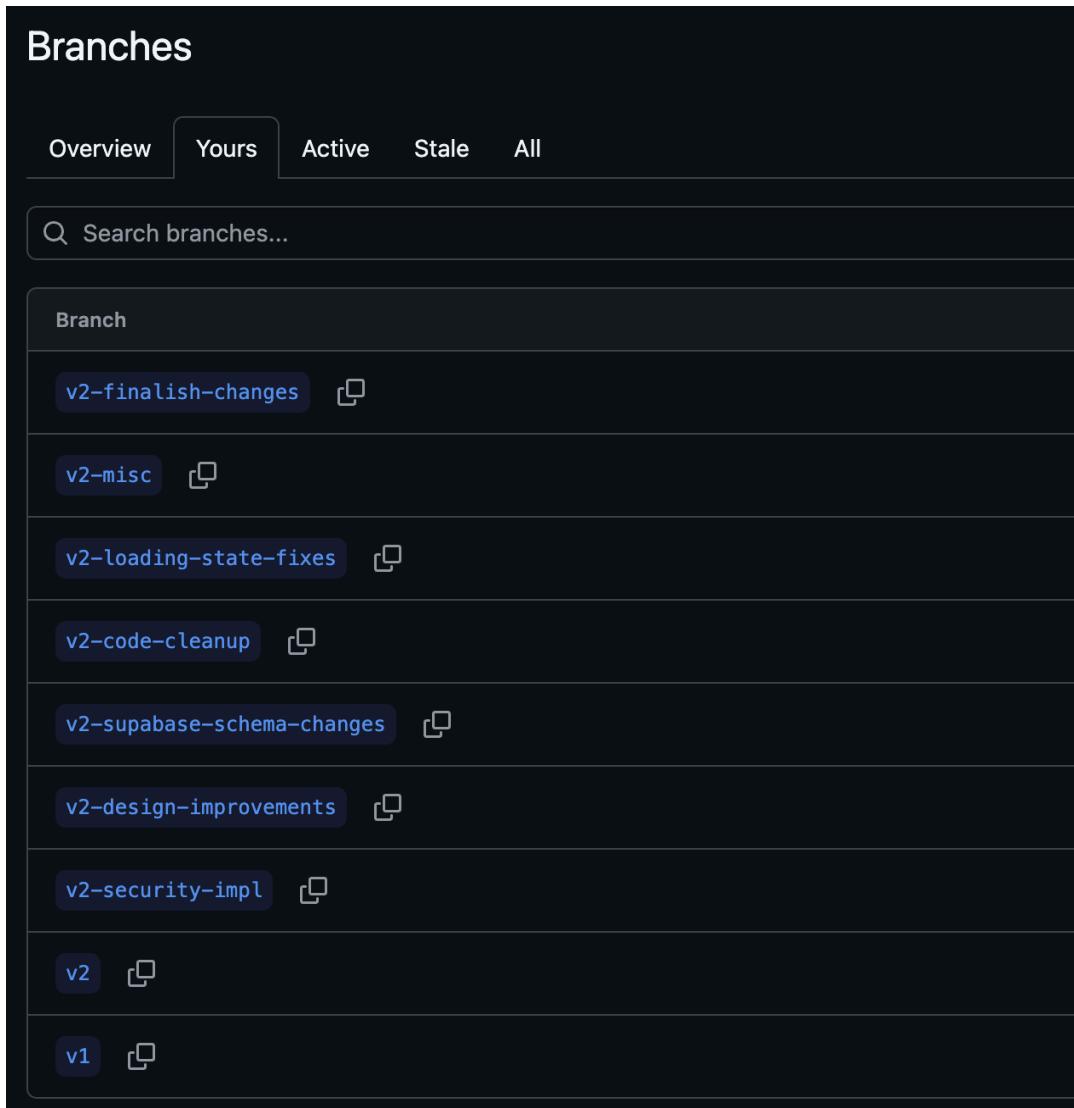


Figure 13: Neatly GitHub branches

## Code development

Generative AI tools allowed a working POC to be finished within the short allocated timeframe. Initial wireframes (appendix 4) were generated using Loveable to get a baseline view. Cursor was used to generate boilerplate NextJS code and then set up Supabase, saving time on repetitive setup. Claude also helped with API integration design, allowing for rapid prototyping. Specifically, it helped integrate JWT handling. These tools saved an estimated 25-35% of development time and were primarily used for high-level architecture design and complex integration rather than full end-to-end coding.

## Task trackers

To help plan the workload and track progress, Jira was used, which is industry standard. Specifically, a kanban board was chosen over a Gantt chart to better prioritise development tasks.

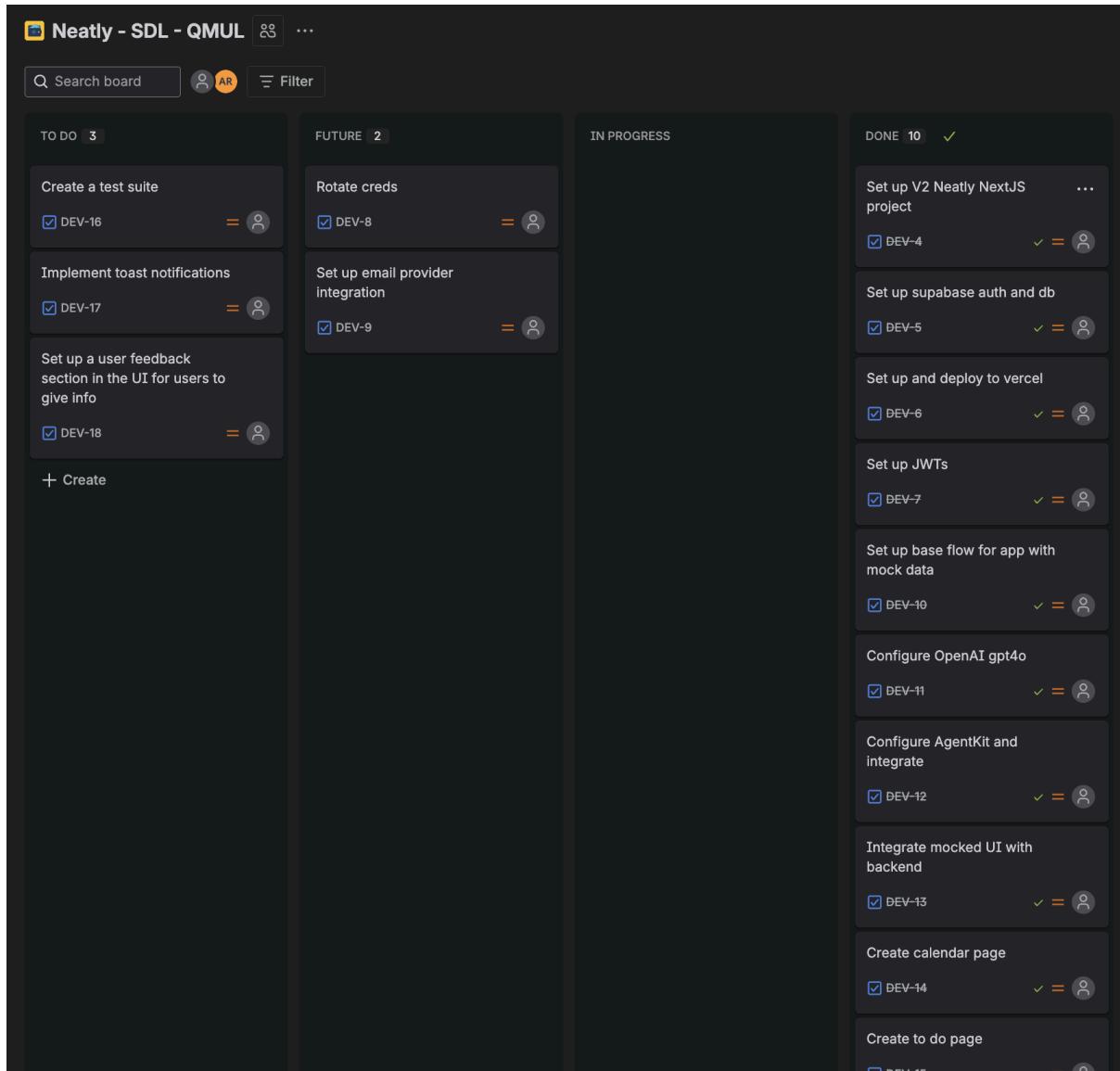


Figure 14: Neatly Jira Kanban board

# Security risk assessment

Neatly processes sensitive workplace data in order to generate summaries, create deadlines, and conduct research. Hence, potential risks must be assessed and mitigated.

## Assessment

ID	Risk	Description
R1	Unauthorised Data Access	Authentication flaws could let other users view other users' sensitive data.
R2	Unauthorised API Access	If API endpoints are left unprotected, malicious users could access or manipulate other users' data.
R3	AI key theft	The ChatGPT credentials can be stolen leading to increased costs and data leaks.
R4	AI Hallucination	Inaccurate AI outputs could mislead users, possibly affecting quality of output affecting company reputation.
R5	Third party outages	Vercel, Supabase and ChatGPT could have downtimes making Neatly inoperable.
R6	Data leaks	Sensitive information could be accidentally displayed in server logs or browser cache.

Table 2: Risk assessment

## Severity evaluation

The severities of risks were evaluated using Binary Risk Analysis (BRA) (Sapiro, 2011). BRA evaluates risk through answering ten structured yes/no questions. The first six look at threat likelihood and the rest measure threat impact. The 6 likelihoods are then passed through 5 matrices with the other 4 going through another 3 matrices. This produces an overall classification of Low, Medium, or High, allowing for objective prioritisation without subjective scoring.

The result was one High, four Medium, and one Low risk. The highest risk is third-party outages. Neatly depends on external platforms for hosting, data storage, and AI services, meaning outages will directly impact availability. AI key theft was deemed Low risk due to strong server-side credential management. The remaining risks were calculated as Medium risks. Although they had mitigations in place, that only reduces the likelihood, not the impact of the risk.

The full risk analysis, questions, process and reference matrices are available in Appendix 15.

## Mitigation

Most mitigations have already been implemented. For unauthorised data access, Supabase Auth is used for secure user authentication. It also provides RLS isolating users to only their rows. For unauthorised API access, endpoints are protected using JWT-based header authorisation verified on the server against Supabase (figure 11). This ensures only authenticated access.

To prevent AI key theft, credentials are stored on the server-side in environment variables and are never exposed in the client-side browser. Moving forward, a strict plan will be in place to rotate the secrets regularly.

The screenshot shows the Vercel interface for managing environment variables. The left sidebar has a dark theme with various sections like General, Build and Deployment, Domains, Environments, Environment Variables (which is selected and highlighted in grey), Git, Integrations, Deployment Protection, Functions, Caches, Cron Jobs, Microfrontends, Project Members, Drains, Security, Connectivity, and Advanced. The main content area has two search bars at the top: one for 'Search...' and another for 'All Environments' with a 'Last Updated' dropdown. Below is a table of environment variables:

Name	Scope	Last Updated	Actions
SUPABASE_URL	All Environments	Added Jan 3	...
SUPABASE_ANON_KEY	All Environments	Added Jan 3	...
SUPABASE_SERVICE_ROLE_KEY	All Environments	Added Jan 3	...
NEXT_PUBLIC_SUPABASE_URL	All Environments	Added Jan 3	...
NEXT_PUBLIC_SUPABASE_ANON_KEY	All Environments	Added Jan 3	...
NEXT_PUBLIC_DEV_SUPABASE_REDIRECT_URL	All Environments	Added Jan 3	...
OPENAI_API_KEY	All Environments	Added Jan 3	...
NEXT_PUBLIC_AGENTKIT_WORKFLOW_ID	All Environments	Added Jan 3	...

At the bottom of the table, it says 'Shared Environment Variables'.

Figure 15: Vercel environment variables

For hallucinations, outputs are validated using Zod and sources are provided, allowing for human-in-the-loop verification. Zod allows descriptions to be added to JSON-data-schemas which can then be input to OpenAI to define the output. Figure 16 is an example of how Zod is used.

```

export const sectionSchema = z.object({
  kind: z.enum([
    "whats_new",
    "numbers",
    "pros_cons",
    "risks_mitigations",
    "recommendations",
    "faq",
    "custom",
  ]),
  title: z.string(),
  summary: z.string().nullable(),
  blocks: z.array(blockSchema),
})

export const formattedBriefSchema = z.object({
  schema_version: z.string(),
  title: z.string(),
  subtitle: z.string().nullable(),
  topic: z.string().nullable(),
  audience: z.string().nullable(),
  tone: z.string().nullable(),
  tldr: z.array(z.string()).max(5),
  sections: z.array(sectionSchema),
  sources: z.array(sourceSchema),
})

// Email Analysis Schema for structured extraction
export const stakeholderSchema = z.object({
  name: z.string().describe("Full name of the stakeholder"),
  email: z.string().describe("Email address if available, or empty string"),
  role: z.string().describe("Role or position inferred from context"),
  evidence: z.string().describe("Quote or context from the email that identifies this stakeholder"),
})

export const actionItemSchema = z.object({
  description: z.string().describe("Clear, actionable description of the task"),
  assignee: z.string().describe("Person or team responsible, or 'Unassigned' if unclear"),
  priority: z.enum(["high", "medium", "low"]).describe("Priority level based on urgency and importance"),
  evidence: z.string().describe("Quote or context from the email that identifies this action item"),
})

export const deadlineSchema = z.object({
  date: z.string().describe("ISO 8601 date string, or descriptive text if exact date not specified"),
  description: z.string().describe("What the deadline is for"),
  evidence: z.string().describe("Quote or context from the email mentioning this deadline"),
})

export const keyDecisionSchema = z.object({
  decision: z.string().describe("The decision that was made"),
  rationale: z.string().describe("Why this decision was made"),
  evidence: z.string().describe("Quote or context from the email about this decision"),
})

export const openQuestionSchema = z.object({
  question: z.string().describe("The unanswered question"),
  context: z.string().describe("Why this question matters"),
  evidence: z.string().describe("Context from the email thread"),
})

export const emailAnalysisSchema = z.object({
  stakeholders: z.array(stakeholderSchema).describe("People involved in the email thread"),
  action_items: z.array(actionItemSchema).describe("Tasks or actions that need to be completed"),
  deadlines: z.array(deadlineSchema).describe("Any dates or deadlines mentioned"),
  key_decisions: z.array(keyDecisionSchema).describe("Decisions that were made in the thread"),
  open_questions: z.array(openQuestionSchema).describe("Questions that remain unanswered"),
  suggested_replies: z.array(z.object({
    title: z.string().describe("Short title for this reply option (e.g., 'Polite Decline', 'Enthusiastic Acceptance')"),
    content: z.string().describe("The full email body text"),
  })).describe("3 distinct reply options with different tones or approaches"),
})

```

Figure 16: Zod Schema definitions

For data leakage, care is taken to avoid writing sensitive data to logs or browser storage. In the future, a custom GitHub bot is planned to help identify sensitive logs before production.

The above mitigations have reduced the likelihoods but the impacts remain prominent. For third-party outages, a loosely coupled architecture is implemented where some services can be unresponsive with the UI providing informative error messages. However, this likelihood remains high due to the permanent dependency.

# Conclusion

The POC successfully validated the core aim of reducing email overload through AI-driven extraction of tasks, deadlines, and stakeholders, combined with agentic research capabilities. Although fewer features were built compared to the initial V1 scope, the narrowed focus better addresses user pain-points. The implementation of OpenAI's AgentsSDK also demonstrated the technical feasibility of agentic workflows.

## Next steps

The next steps following submission are email provider integration to automate analysis as emails are received. This improvement solves the usability barrier when it comes to inputting threads manually. Research indicates that reducing interaction steps significantly improves application adoption (Nielsen, 1993). Outlook will be prioritised due to its monopoly within financial companies, followed by Gmail for independent analysts. The updated architecture in Figure 17 outlines these planned improvements in contrast with Figure 2.

At the same time user onboarding and marketing will begin. This has a primary goal of gathering qualitative feedback to validate a product-market fit. The target is ten IB analysts within three months, where email overload is most common. User feedback will help with iterative improvements following an Agile methodology.

After product-market validation, Neatly will scale through team expansion, starting with a technical co-founder to accelerate development. The revenue model targets a break-even number-of-users at 350 seats (with base system-usage). I believe this to be achievable within the first year given the ~£500/banker/month value add demonstrated in the business case.

Furthermore, long-term ambitions include expanding vertically into other industries such as management consulting and legal, where similar communication overload challenges exist.

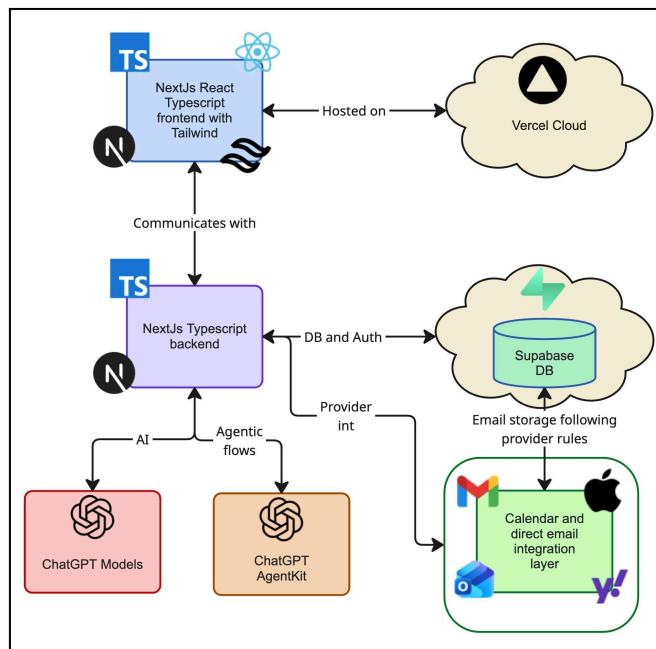


Figure 17: Future proposed architecture outlined in green

## Critical review of presentation feedback, reflections and KSBs

One of the most important lessons I learned was to prioritise user value over technical ambition. In the first version of Neatly, I attempted to build too much (a vast number of features, cron jobs, agentic flows, and automated data pipelines) to maximise the technical learning. However, feedback from my presentation made it clear that the app was trying to solve too many problems and had too broad a value proposition. This led to an important realisation. A focused application that solves one real pain point is more valuable than a complex project with vaguely connected features.

Reducing the scope significantly changed my approach and aligned me with KSB-C9, where I learned to refine business cases around a specific niche (vertically aligned). I shifted Neatly to focus on only one feature: agentic research. This aligned better with user needs and made the completion of the POC more achievable.

The project also helped me align with KSB-C13, as I explored best practices in API security using JWTs. I also researched the risks of exposing secrets on the client-side. Aligned with KSB-C17, I learned that Gmail API integration requires Google approval, due to contractual data-storage rules, which takes weeks making it unachievable in the timeframe. So, the approach was changed and a manual input flow was integrated to better meet the timeline, while still ensuring the core concept is functional.

The refinement from V1 to V2 directly improved the final concept's effectiveness, enabling a fully functional POC with a clearer value proposition. The project also developed my understanding of KSB-C6, as I applied efficient project management through Jira and GitHub to track progress and manage tasks. Furthermore, KSB-C10 was engaged through using new AI development tools, which sped up prototyping whilst maintaining code quality. These experiences have shaped how I will approach future complex technology solutions, balancing ambition with pragmatic delivery.

# References

- AlphaSense (2024). *The Toll of Analyst Burnout*. [online] Alpha-sense.com. Available at: <https://www.alpha-sense.com/blog/trends/toll-of-analyst-burnout> [Accessed 11 Jan. 2026].
- Bachan, P. (2024). *Tackling the email overload in financial services: Challenges and solutions*. [online] Uipath.com. Available at: <https://www.uipath.com/blog/industry-solutions/banks-overcoming-email-overload-with-automation> [Accessed 11 Jan. 2026].
- Bank of America (2025). *AI Adoption by BofA's Global Workforce Improves Productivity, Client Service*. [online] Bank of America. Available at: <https://newsroom.bankofamerica.com/content/newsroom/press-releases/2025/04/ai-adoption-by-bofa-s-global-workforce-improves-productivity--cl.html> [Accessed 10 Jan. 2026].
- BBC (2021). Young Goldman Sachs bankers ask for 80-hour week cap. *BBC News*. [online] 19 Mar. Available at: <https://www.bbc.co.uk/news/business-56452494> [Accessed 11 Jan. 2026].
- Bouvier (2025). *Human Technology*. [online] Humantechology.jyu.fi. Available at: <https://humantechology.jyu.fi/archive/vol-20/issue-1/bouvier-et-al> [Accessed 7 Jan. 2026].
- Chelli, M., Descamps, J., Lavoué, V., Trojani, C., Azar, M., Deckert, M., Raynier, J.-L., Clowez, G., Boileau, P. and Ruetsch-Chelli, C. (2024). Hallucination Rates and Reference Accuracy of ChatGPT and Bard for Systematic Reviews: Comparative Analysis. *Journal of Medical Internet Research*, [online] 26, p.e53164. doi:<https://doi.org/10.2196/53164> [Accessed 11 Jan. 2026].
- Dabbish, L.A. and Kraut (2010). Email overload at work: an analysis of factors associated with email strain. *IEEE Engineering Management Review*, 38(1), pp.76–90.  
doi:<https://doi.org/10.1109/emr.2010.5494697> [Accessed 11 Jan. 2026].
- Deloitte (2023). *Unleashing a new era of productivity in investment banking through the power of generative AI*. [online] Deloitte Insights. Available at: <https://www.deloitte.com/us/en/insights/industry/financial-services/generative-ai-in-investment-banking.html> [Accessed 11 Jan. 2026].
- Deloitte (2025). *Trust Emerges as Main Barrier to Agentic AI Adoption in Finance and Accounting, Despite Optimism Around the Tech – Press release | Deloitte US*. [online] Deloitte. Available at: <https://www.deloitte.com/us/en/about/press-room/trust-main-barrier-to-agnostic-ai-adoption-in-finance-and-accounting.html> [Accessed 11 Jan. 2026].
- GDPR Regulation (2025). *GDPR for Financial Services - GDPR Regulation*. [online] GDPR Regulation -. Available at: <https://www.gdprregulation.eu/gdpr-for-financial-services> [Accessed 10 Jan. 2026].
- Laurent, A. (2025). *Pharma Document AI & OCR Accuracy: A Benchmark Analysis*. [online] IntuitionLabs. Available at: <https://intuitionlabs.ai/articles/pharma-document-ai-ocr-benchmarks> [Accessed 13 Jan. 2026].

Lewis, T. (2024). *We Tested 15+ AI Email Assistants. Here Are The Top 5 in 2025 | Motion.* [online] Usemotion.com. Available at: <http://usemotion.com/blog/best-ai-email-assistant> [Accessed 8 Jan. 2026].

MadeMarket (2025). *4 Investment Banking Fire Drills That Shouldn't Exist (And How to Eliminate Them).* [online] Mademarket.com. Available at: <https://www.mademarket.com/blog/4-investment-banking-fire-drills-that-shouldnt-exist-and-how-to-e-liminate-them> [Accessed 7 Jan. 2026].

Mailbird (2025). Mailbird. [online] Getmailbird.com. Available at: <https://www.getmailbird.com/email-overload-survey> [Accessed 13 Jan. 2026].

Mark, G., Judith, D. and Klocke, U. (2008). The cost of interrupted work. *Proceedings of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, [online] pp.107–110. doi:<https://doi.org/10.1145/1357054.1357072> [Accessed 11 Jan. 2026].

McKinsey Global Institute (2012). *The social economy: Unlocking value and productivity through social technologies.* [online] McKinsey & Company. Available at: <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-social-economy> [Accessed 11 Jan. 2026].

Microsoft (2025). *Breaking down the infinite workday.* [online] Microsoft.com. Available at: <https://www.microsoft.com/en-us/worklab/work-trend-index/breaking-down-infinite-workday> [Accessed 13 Jan. 2026].

Microsoft WorkLab, M. (2025). *WorkLab: Hard Data, Compelling Stories, Vital Insights.* [online] Microsoft.com. Available at: <https://www.microsoft.com/en-us/worklab/work-trend-index/infinite-workday-report> [Accessed 7 Jan. 2026].

Murty, R.N., Dadlani, S. and Das, R.B. (2022). *How Much Time and Energy Do We Waste Toggling Between Applications?* [online] Harvard Business Review. Available at: <https://hbr.org/2022/08/how-much-time-and-energy-do-we-waste-toggling-between-applications> [Accessed 11 Jan. 2026].

Nielsen, J. (1993). *Response Times: The 3 Important Limits.* [online] Nielsen Norman Group. Available at: <https://www.nngroup.com/articles/response-times-3-important-limits> [Accessed 11 Jan. 2026].

PostHog (2023). *NPS vs CSAT vs CES: Which is best for SaaS? - PostHog.* [online] Posthog.com. Available at: <https://posthog.com/product-engineers/nps-vs-csat-vs-ces> [Accessed 10 Jan. 2026].

Sapiro, B. (2011). *Binary Risk Analysis.* [online] Protect.io. Available at: <https://binary.protect.io> [Accessed 7 Jan. 2026].

Sauermann, J. (2016). Performance measures and worker productivity. *IZA World of Labor.* [online] doi:<https://doi.org/10.15185/izawol.260> [Accessed 11 Jan. 2026].

Selby-Jennings (2024). *Investment Banking Compensation benchmarking for key investment banking roles across the USA, Europe, and APAC*. [online] Available at: <https://hub.selbyjennings.com/hubfs/Selby%20Jennings%202024/EUROPE/Selby-Jennings-Investment-Banking-Compensation-Guide-Global.pdf> [Accessed 8 Jan. 2026].

DigitalDefynd (2025). *Are Banking Sector Jobs Stressful? [10 Key Factors]/[2026]*. [online] DigitalDefynd Education. Available at: <https://digitaldefynd.com/IQ/is-banking-a-stressful-job/> [Accessed 10 Jan. 2026].

Vellum AI (2025). *LLM Leaderboard 2024*. [online] www.vellum.ai. Available at: <https://www.vellum.ai/llm-leaderboard> [Accessed 2 Jan. 2026].

Villemondeix, J. (n.d.). *Investment Banking Burnout: A Temperature Check*. [online] Available at: <https://upslide.com/wp-content/uploads/2024/12/Investment-Banking-Burnout.pdf> [Accessed 11 Jan. 2026].

Wall Street Oasis (2025). *2024 Investment Banking Working Conditions Survey*. [online] Available at: <https://www.wallstreetoasis.com/files/2024%20WSO%20IB%20Working%20Conditions%20Survey.pdf> [Accessed 11 Jan. 2026].

# Appendix

## Appendix 1 - Neatly V1 application

This version of the app was also built using the same architecture and tech stack as defined in Figure 2.

The screenshot shows the 'Priority' inbox view in Neatly V1. The sidebar includes links for INBOX ZERO, Priority, Spam & Phishing, Bulk Unsubscribe, Analytics, AUTOMATIONS, Daily Digest, Calendar, Research, Follow Ups, Dark Mode, Settings, and Sign Out. The main area displays a list of messages from 'Redacted - Personal Emails' with dates Oct 19, Oct 19, Oct 19, Oct 15, and Oct 9. Each message has 'URGENT' and 'IMPORTANT' labels. A placeholder text 'Select an email to view' is visible on the right.

The screenshot shows the 'Follow Ups' interface in Neatly V1. The sidebar includes links for INBOX ZERO, Priority, Spam & Phishing, Bulk Unsubscribe, Analytics, AUTOMATIONS, Daily Digest, Calendar, Research, Follow Ups, Dark Mode, Settings, and Sign Out. The main area lists various follow-up tasks with their details and auto-follow-up status (e.g., Product Demo Request - Acme Corp, Partnership Proposal Follow-up, Pricing Question - Enterprise Plan, Conference Speaking Opportunity, Job Application - Senior Developer Position, Proposal Review - Website Redesign, Trial Extension Request).

## Research

**Sarah Chen**  
Jan 15, 2025  
Preparing for Q4 Strategy Meeting with Acme Industries  
Client: Acme Industries  
January 22, 2025 at 2:00 PM

**David Park**  
Jan 16, 2025  
Client Prep: Meeting with Global Dynamics Next Week  
Client: Global Dynamics  
January 24, 2025 at 10:00 AM

**Emily Rodriguez**  
Jan 17, 2025  
Prep Materials for Innovation Labs Partnership Discussion  
Client: Innovation Labs  
January 25, 2025 at 3:30 PM

**Michael Chen**  
Jan 18, 2025  
Quarterly Review with Summit Enterprises - Research Needed  
Client: Summit Enterprises  
January 28, 2025 at 1:00 PM

### Background

Summit Enterprises is a family-owned business that has grown into a \$1.2B revenue company over 45 years. They specialize in supply chain management and logistics, serving the retail and e-commerce sectors.

### Recent News

They recently acquired two regional competitors and are expanding their warehouse network by 40%.

### Leadership

The company is currently run by the second generation, with CEO Patricia Summit taking over from her father in 2018.

### Current Challenges

Their main challenge is digital transformation, which is why they're partnering with technology companies.

### Key Contacts

Key contacts include COO, John Summit (Patricia's brother), and Chief Digital Officer, Karen Lee, who joined from Amazon last year.

## Calendar

**Automatically scheduled events**

<b>Meeting with Alex</b> November 4, 2025 at 2:00 PM <a href="#">Thread</a> <a href="#">Calendar</a>	<b>Meeting with Jill</b> November 5, 2025 at 10:00 AM <a href="#">Thread</a> <a href="#">Calendar</a>
--	---

**November 2-8, 2025** [Today](#)

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
11 AM						
12 PM						
1 PM						
2 PM						
3 PM						
4 PM						
5 PM						

**Meeting with Alex**  
11:30 AM

**Meeting with Jill**  
3:45 PM

**Meeting with Jill**  
3:45 PM

## Daily Digest

Sunday, November 2, 2025

**Total Emails** 47 today **Unread** 23 emails **Important** 8 emails **Urgent** 3 emails

**What You Need to Do Today**

- Urgent: Q4 Budget Review Meeting - Response needed by EOD [View >](#)
- Important email: Client Presentation Tomorrow - Final Review [View >](#)
- Urgent: Server Migration Tonight at 11 PM [View >](#)
- Important email: New Partnership Opportunity with Fortune 500 Company [View >](#)
- Important email: Security Audit Results - Action Items Required [View >](#)

Select an email to view

## Analytics

Last week Oct 26, 2025 - Nov 2, 2025

**Received** 10 emails **Read** 4 emails **Archived** 0 emails **Sent** 0 emails

**Who emails you most**

SENDER	EMAILS
m	5
Redacted - Personal Emails	1
1	1
1	1
1	1

**Domains that email you most**

DOMAIN	EMAILS
Redacted - Personal Emails	5
1	1
1	1
1	1

**Who you email the most**

RECIPIENT	EMAILS
Redacted - Personal Emails	10

**Detailed Analytics**

DATE	COUNT
Oct 27, 2025	1
Oct 28, 2025	5
Oct 30, 2025	1
Oct 31, 2025	1
Nov 1, 2025	1

## Bulk Unsubscribe

One-click bulk unsubscribe from emailers • 49 senders

FROM	EMAILS	READ	ARCHIVED	ACTIONS
Redacted - Personal Emails	9	33%	0%	Unsubscribe Skip Inbox ...
	9	11%	0%	Unsubscribe Skip Inbox ...
	9	78%	0%	Unsubscribe Skip Inbox ...
	8	50%	0%	Unsubscribe Skip Inbox ...
	7	100%	0%	Unsubscribe Skip Inbox ...
	7	43%	0%	Unsubscribe Skip Inbox ...
	7	29%	0%	Unsubscribe Skip Inbox ...
	6	17%	0%	Unsubscribe Skip Inbox ...
	6	100%	0%	Unsubscribe Skip Inbox ...
	6	17%	0%	Unsubscribe Skip Inbox ...
	5	0%	0%	Unsubscribe Skip Inbox ...
	5	60%	0%	Unsubscribe Skip Inbox ...

## Spam & Phishing Detection

AI-powered detection of suspicious emails • 4 suspicious emails

Analyzing...

SPAM (1)	Phishing (3)
stallone.alliancestreet.uk@viazohocrm.com	Sep 9
Your free copy of "Fast-Track To Zero Tax" is inside...	
Hi Andrei, Thanks for requesting a copy of "Fast-Track To Zero Tax". Simply click the link below to download your copy. Inside...	
SPAM (85%)	
Redacted - Personal Emails	Sep 9
PHISHING (85%)	
Redacted - Personal Emails	Sep 9
PHISHING (85%)	
Redacted - Personal Emails	Sep 9
PHISHING (85%)	

Your free copy of "Fast-Track To Zero Tax" is inside...

stallone.alliancestreet.uk@viazohocrm.com  
Sep 9, 2025, 1:07 PM

SPAM (85%)

Detection Reasons:

- Suspicious sender patterns
- Urgency tactics
- Too-good-to-be-true offers

Hi Andrei,

Thanks for requesting a copy of "Fast-Track To Zero Tax". Simply click the link below to download your copy. Inside you'll learn:

- The #1 reason why business owners around the globe can't seem to achieve a zero tax status, no matter how hard they try!
- Why you should NEVER trust people who tell you escaping your home-country tax is impossible and what you should do instead.
- The honest TRUTH about Dubai's Freezones and why it can be so much easier to relocate your business in 2024 than you ever before!
- And much, much more!

[Click or tap here to download.](#)

## Appendix 2 - MoSCoW prioritised requirements

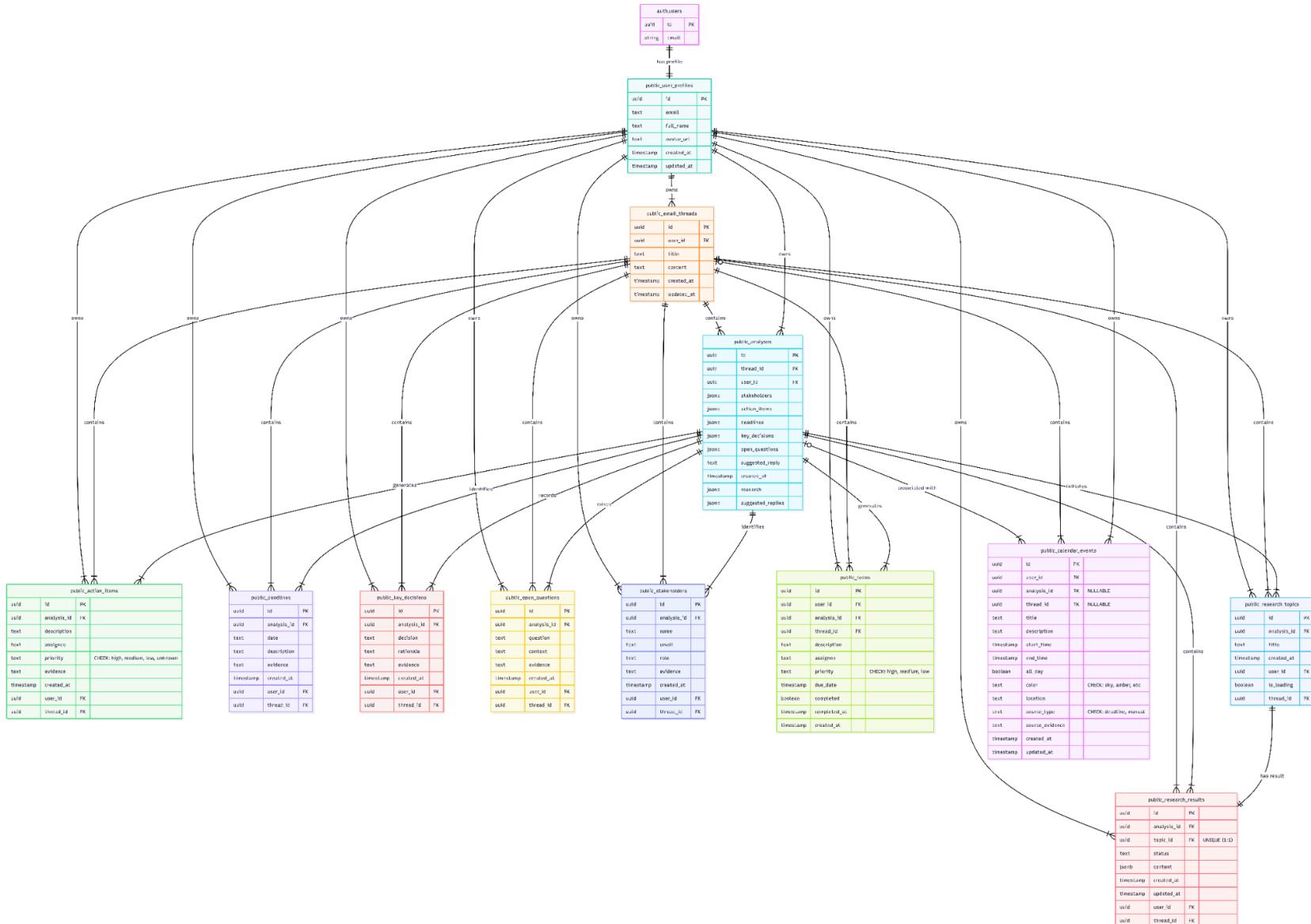
### Functional Requirements

ID	Requirement	Priority
FR1	Accept pasted email threads via input form.	Must have
FR2	Extract stakeholders, action items, deadlines, and key questions	Must have
FR3	Trigger research agents to identify and summarise topics	Must have
FR4	Display research outputs (FAQs, TLDR, Why it matters, Sources)	Must have
FR5	Generate suggested replies from email content	Should have
FR6	View structured summaries for past analyses	Must have
FR7	Create to-do items from action points	Must have
FR8	Task event scheduling to ensure task completion	Must have
FR9	User login and session-based access via Supabase Auth	Must have
FR10	Direct email provider integration	Should have
FR11	Agentic meeting scheduling to book meetings off the back of threads	Could have

## Non-Functional Requirements

ID	Requirement	Priority
NFR1	Response time for analysis/UI render < 60 seconds in 60% of cases	Must have
NFR2	Secure API access via JWT and Supabase Auth	Must have
NFR3	Row-Level Security for data security	Must have
NFR4	99% availability during testing	Must have
NFR5	Intuitive user-experience focused design.	Should have
NRF6	Code is written with future maintainability in mind ensuring seamless work in the future	Should have
NFR7	The app should be portable to work with different browsers, hosting, databases and AI providers.	Could have

## Appendix 3 - Neatly Data schema diagram



## Appendix 4: Neatly V1.5 Wireframes

**New Thread**

Paste or upload an email thread to analyze

**Email Content**  
Include the full email thread for the most accurate analysis

**Title (optional)**  
Auto-detected from subject line if empty

**Email Thread**  
Paste your email thread here...  
Include headers like From:, To:, Date:, Subject: for better parsing.

0 characters

This wireframe shows the 'New Thread' page. At the top, there's a header with the Neatly logo and navigation links for Threads, New, Calendar, To-Do, History, and Settings. Below the header is a section titled 'New Thread' with a sub-instruction 'Paste or upload an email thread to analyze'. This is followed by a 'Email Content' section with a note to include the full email thread for accurate analysis. There's a 'Title (optional)' field containing 'Auto-detected from subject line if empty'. Below that is a large text area for pasting an email thread, with instructions to include headers like From:, To:, Date:, Subject:. A note at the bottom of this area says '0 characters'.

**To-Do**

**Pending (2)**

- Schedule call for Thursday or Friday afternoon  
Re: Contract update for Q1 deliverables **High**
- Get legal sign-off on liability clause changes  
Re: Contract update for Q1 deliverables **High**

This wireframe shows the 'To-Do' page. At the top, there's a header with the Neatly logo and navigation links for Threads, New, Calendar, To-Do, History, and Settings. Below the header is a section titled 'To-Do' with a 'Pending (2)' count. It lists two tasks: 'Schedule call for Thursday or Friday afternoon' and 'Get legal sign-off on liability clause changes', both associated with 'Re: Contract update for Q1 deliverables' and a 'High' priority level.

Neatly

Threads + New Calendar To-Do History Settings

## History

Thread created (New Thread)  
Project kickoff meeting - Action items · Jan 3, 2025 2:00 PM

Thread created (New Thread)  
Re: Contract update for Q1 deliverables · Jan 2, 2025 10:30 AM

Research brief generated (Brief Ready)  
Re: Contract update for Q1 deliverables · Jan 2, 2025 10:30 AM

Neatly

Threads + New Calendar To-Do History Settings

## Threads

Your analyzed email threads

Search threads...

Project kickoff meeting - Action items  
Jan 3, 2025 No brief >

Re: Contract update for Q1 deliverables  
Jan 2, 2025 Brief ready >

[Back to thread](#)

## Research Brief

Project kickoff meeting - Action items

[Regenerate](#) [Export JSON](#) [Copy Summary](#)

### Summary

- Analysis complete for this email thread

### Action Items

Review and respond to the thread

Medium

1 item

### Open Questions

What is the desired outcome of this thread?

Low

1 item

Generated 04/01/2026, 22:30:44

[Threads](#) [New](#) [Calendar](#) [To-Do](#) [History](#) [Settings](#)

## Project kickoff meeting - Action items

[View Brief](#) [Draft Reply](#) [Regenerate](#)

### Thread Messages

Mike Johnson January 3, 2025 at 2:00 PM

From: Mike Johnson <mike@agency.com>  
Date: January 3, 2025 at 2:00 PM  
To: Team <team@agency.com>  
Subject: Project kickoff meeting - Action items

Team,

Great kickoff meeting today! Here are the action items we discussed:

@Lisa - Complete brand guidelines by Jan 10  
@Tom - Set up development environment by Jan 8  
@Everyone - Review competitor analysis document and provide feedback by EOD Friday

The client has emphasized that the beta launch must happen by Feb 28th. No exceptions.

Let's make this happen!

Mike

### Research Brief

Brief generated successfully

Stakeholders	Action Items	Open Questions	
0	95	1	
Deadlines	0	Open Questions	1
Risk Flags	0		

[Open Full Brief](#)

## Appendix 5: Success criteria

Metric	Target value	Justification	Test methods
Accurate extraction of actions, deadlines, stakeholders, and decisions	$\geq 90\%$ accuracy	Document AI pipelines achieve $\sim 90\%$ accuracy on extraction tasks (Laurent, 2025)	User error flagging and comparison against annotated test set
User-perceived usefulness	$\geq 75\%$ rate as “useful” or “very useful”	75-85% is a good CSAT benchmark for SaaS (PostHog, 2023)	In app feedback surveys
Time to generate analysis	Between 5-20 seconds in 90% of tests with varying token input lengths	10 seconds is maximum before user attention drifts (Nielsen, 1993)	Log inspection
Uptime and backend service availability	$\geq 99.5\%$ uptime	Standard SLA threshold for production systems	Log inspection
Secure user authentication and data isolation	100% JWT-protected routes and row-level-security (RLS) in Supabase	Non-negotiable for financial data handling	Supabase analytics inspection
Reliable agentic tool calls	< 1% failure rate	Standard SLA threshold for production systems	Log inspection

## Appendix 6: Cost justification calculations

### Current

Base IB UK analyst salary 2025: **£75,000 (Selby-Jennings, 2024)**

Base 45-hour weeks: **2,340 workable hours-per-year**

Hourly wage: **£32**

Base office worker time spent on emails: **28% (Mailbird, 2025)**

Time spent on email: **28% of 45-hours = ~13-hours**

Base wage spent on email usage per month = **(13\*£32) \* 4 = £1664/month**

### Proposed

Base IB UK analyst salary 2025: **£75,000 (Selby-Jennings, 2024)**

Base 45-hour weeks: **2,340 workable hours-per-year**

Hourly wage: **£32**

Base office worker time spent on emails after reasonable productivity increases: **35% (Deloitte, 2023)**  
improvement on **28% (Mailbird, 2025)**

Time spent on email: **35% improvement on 28% of 45-hours = ~8.5-hours**

Base wage spent on email usage per month = **(8.5\*£32) \* 4 = ~£1100/month**

Total estimated base savings: **~£564/banker**

## Appendix 7 - Initial resources and personnel costs

Category	Item	Cost	Justification
Personnel	Full stack developer	£0	In house developer to build the app
Tech	Cursor	£20/m	Increase developer productivity
Tech	Supabase	£25/m	Required for data storage and authentication.
Tech	Vercel	£0/m	Generous free hosting for UI & API
Tech	OpenAI (ChatGPT 4o)	< £12/m (per 1M tokens I/O, sufficient for testing)	Powers agentic flow and AI thread analysis
Personnel	Marketing	£0/m	No ads, self outreach
	Total	< £57/m	

## Appendix 8 - Post launch resources and personnel costs

Category	Item	Cost	Justification
Personnel	Frontend developer	£4000	Maintains, improves, manages the UX of the application
Personnel	AI developer	£6500	Builds, maintains improved agentic and AI capabilities based on user feedback.
Tech	Cursor	£200/m	Increase developer productivity
Tech	Supabase	£400/m	Required for data storage and authentication.
Tech	Vercel	£20/m	Hosting UI & API
Tech	OpenAI (ChatGPT 4o)	> £1000/m	Powers agentic flow and AI thread analysis
Personnel	Sales	£3000/m	Outreaches and sells the app to IBs
Personnel	Support	£2000/m	Manages user support queries and coordinates with the developers
	Total	> £17,120/m	Could be higher due to AI costs

## Appendix 9 - External Factors

Factors	Obstacle	Mitigation
Political	Cross-border data rules and government sensitivity around financial communications in multi-national companies.	Data residency will be configured. Region restrictions may also be used.
Economical	The rising costs of vendor software and LLM costs as usage scales.	POC usage and costs will be monitored with a future plan to run a local model instead of third-party models.
Social	Bankers may be hesitant to trust AI in their core business email.	Provide rigorous human-in-the-loop methods and provide clear training sessions
Technological	LLM hallucinations, errors and outages.	Structured JSON outputs and thorough service-level-indicators for outage detection.
Environmental	Increased LLM usage utilises more water and increases carbon emissions conflicting with company environmental-social-governance (ESG) goals	Cached research topics minimising token usage. Batch processing during the night during lower cost hours
Legal	Upholding MNPI, GDPR and other data requirement laws.	Encrypted data, Supabase row-level-security and data redaction if necessary.

## Appendix 10 - Internal Factors

Factors	Obstacle	Mitigation
Strengths	Team may not standardise workflows creating separation rather than unifying strengths	Create robust workflows ahead of time and standard-operating-procedures (SOPs)
Weaknesses	Limited AI experience within the team can slow down development and quality of the agentic flow.	Minimise scope creep and focus on the AI feature mainly. If needed a contractor can be hired to oversee its development.
Opportunities	Opportunity cost for slow rollout and launch process causing talent and team churn or faster competitor adoption.	Focus on fast iteration and prototyping focusing on a few key-performance-indicators at a time.
Threats	Low adoption of the tool does not cover expenses. Larger more trusted companies can release the same features and have the brand reputation to undercut Neatly.	Stay niched on the IB vertical tailoring the model specifically to investment banking.

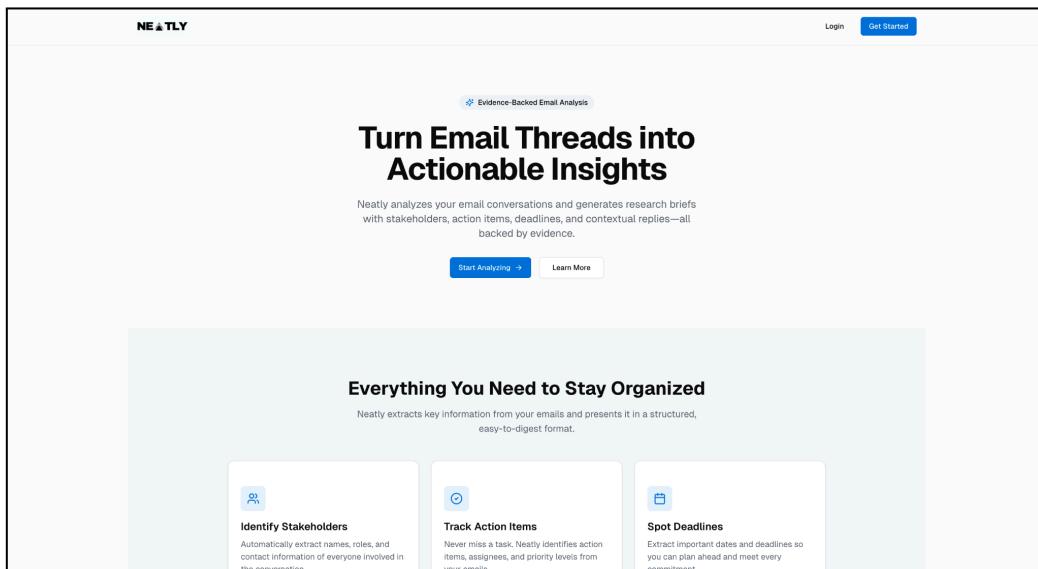
## Appendix 11 - Neatly V2 POC screenshots and video

A video demo can be watched here: [Demo of Neatly](#)

The app can also be securely logged in (Supabase Auth is utilised) and tested here:  
<https://neatly-silk.vercel.app/>

Below are some screenshots of the entire app too

### Landing Page



### Auth

Sign up'. The right screenshot shows the 'Create Your Account' sign-up screen with fields for 'Email' (containing 'name@example.com'), 'Password', and 'Confirm Password', and a 'Sign Up' button. Below the form is a link 'Already have an account? [Login](#)'."/>

## Home page

**Welcome back**

Here's what's happening with your analyses

**Total Analyses**: 1 Email threads analyzed

**Pending To-Dos**: 4 Tasks to complete

**Upcoming Events**: 0 In the next 7 days

**Recent Analyses**: Q1 Onboarding Workshop — agenda + next steps (1/6/2026)

**Pending To-Dos**:

- Confirm recording of the workshop for internal use only and not for ... (Medium)
- Confirm if the 90-minute format is still feasible (Medium)
- Confirm which department the invoice should be sent to: People O... (Medium)
- Send estimated cost to Priya Shah (Medium)

## Email thread analysis home page

Analysis > Q1 Onboarding Workshop — agenda + next steps

Analyzed on 6 January 2026 | All Analyses

**Q1 Onboarding Workshop — agenda + next steps**

**Research Topics**

- Feasibility and efficiency of different workshop durations (60-minute vs. 90-minute formats) for onboarding processes.
- Ethical considerations and policies for recording internal workshops and usage restrictions.
- Impact of including a security section in onboarding workshops on organizational safety protocols.
- Cost implications and budget allocation strategies for workshops involving multiple departments (e.g., People Ops vs. Finance).
- Challenges of coordinating remote workshops across international time zones (London vs. New York).

**STAKEHOLDERS**: 4

**ACTION ITEMS**: 6 (1 HIGH PRIORITY)

**DEADLINES**: 2 (Next: 5 Jan)

**QUESTIONS**: 0 (1 DECISIONS)

**SUGGESTED REPLIES**: 3

**Original Email Thread**

## Researched topic page

**WHAT IT MEANS**

- Different resource allocation can significantly improve workshop outcomes and interdepartmental collaboration.
- Ensures financial transparency and accountability.
- Aligns resources with organizational objectives.
- Allocates available staff across departments.

**WHAT IT IS**

Managing budgets for workshops that involve multiple departments require a sophisticated approach to balance costs and benefits across units such as People Ops and Finance.

Workshops that involve multiple departments necessitate careful planning of both direct costs like facilitator fees and indirect costs such as administrative overheads. Employing various budget allocation strategies helps distribute these costs fairly and ensures that each department understands its financial responsibilities.

**RECOMMENDATIONS**

- Direct costs include facilitator fees, materials, and recording logistics.
- Indirect costs involve overheads like staff time and administrative support.

**Budgeting Strategies**

- Zero-based budgeting prioritizes resource efficiency by requiring justification of all expenses.
- Bottom-up budgeting allows for more realistic, department-driven planning.

**Proportional Cost Sharing**

- Costs may be shared equally proportionally based on usage or through other means.
- Departmental account aligns costs with departmental benefits.

**PROS/CONS**

- Transparent budgeting fosters teamwork and collaboration.
- Proportional cost-sharing is equitable and aligns with usage.
- Contingency funds offer flexibility in unexpected situations.
- Zero-based budgeting is time-consuming.
- Proportional budgeting may lead to administrative inefficiencies.
- Managing congruence can reduce liquidity for other expenditures.

**NOTES**

- Address both direct and indirect costs early in planning.
- Employ zero-based or bottom-up budgeting based on departmental preferences.
- Use proportional cost sharing for more efficient resource utilization.
- Coordinate closely with procurement to speed up financial processing.
- Document cost allocation metrics to ensure transparency and fairness.

**TOOLS**

Effective Financial Budgeting  
Budgeting Techniques: Zero Based & Bottom-up

**Share Cost Allocation Metrics**

## Thread analysis items

**Q1 Onboarding Workshop — agenda + next steps**

**Stakeholders**

People involved in this conversation

- Priya Shah  
priya.shah@acmeCorp.co.uk  
Head of People Ops, Acme Corp
- Andrei Riza  
AndreRiza@acmeCorp.co.uk  
Representative of Hearty Ltd
- Tom Evans  
tom.evans@acmeCorp.co.uk  
Programme Manager, Acme Corp
- Finance Team  
Finance@acmeCorp.co.uk  
Finance team at Acme Corp

**Q1 Onboarding Workshop — agenda + next steps**

**Action Items**

Tasks identified from the thread

- Send revised agenda with security section and final quote to Priya Shah  
Assignee: Andrei Riza  
**High Priority**  
"Can you send me the revised agenda (with the security section) and a final quote by end of day tomorrow?"
- Include company registration number on the invoice  
Assignee: Andrei Riza  
**Medium Priority**  
"Also please include your company registration number on the invoice."
- Confirm recording of the workshop for internal use only and not for marketing  
Assignee: Andrei Riza  
**Medium Priority**  
"Please confirm you won't use it for marketing."
- Confirm if the 90-minute format is still feasible  
Assignee: Andrei Riza  
**Medium Priority**  
"Also, is the 90-minute format still feasible? We may need to shorten it."

**Q1 Onboarding Workshop — agenda + next steps**

**Research Topics**

a security section in onboarding workshops on organizational

**Deadlines**

Important dates and milestones

- 05/01/2026  
Deadline for sending revised agenda and final quote  
"by end of day tomorrow"
- 05/01/2026  
Finalize suppliers  
"we're aiming to finalize suppliers tomorrow."

**Q1 Onboarding Workshop — agenda + next steps**

**Suggested Replies**

Choose from distinct draft options

**Brief Reply**

H1 Priya,  
Acknowledged. I'll have the revised agenda and quote ready by the end of day tomorrow.  
Best,  
Andrei

**Detailed Reply**

**Question-Focused Reply**

**Original Email Thread**

View the full conversation context

## Calendar

**NEATLY**

MENU

- Home
- Calendar
- To Do
- Analysis

Light

Sign Out

**Calendar**

View and manage your schedule

Today < > **January 2026** Month + New event

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

## To-do

**NEATLY**

MENU

- Home
- Calendar
- To Do
- Analysis

Light

Sign Out

**To Do**

Track and complete actionable items from your analyses

Search tasks...

Status Outstanding Priority Sort

Task	Priority	Thread	Due Date	Assignee
<input type="checkbox"/> Confirm recording of the workshop for internal use only and not for marketing	Medium	Q1 Onboarding Workshop ...	-	Andrei Rizea
<input type="checkbox"/> Confirm if the 90-minute format is still feasible	Medium	Q1 Onboarding Workshop ...	-	Andrei Rizea
<input type="checkbox"/> Confirm which department the invoice should be sent to: People Ops or Finance	Medium	Q1 Onboarding Workshop ...	-	Andrei Rizea
<input type="checkbox"/> Send estimated cost to Priya Shah	Medium	Q1 Onboarding Workshop ...	-	Andrei Rizea

https://neatly-silk.vercel.app/todos

## History

The screenshot shows the 'Analysis History' section of the Neatly interface. On the left, a sidebar menu includes 'Home', 'Calendar', 'To Do', and 'Analysis' (which is selected and highlighted in blue). Below the menu are 'Light' and 'Dark' mode switches, and a 'Sign Out' link. The main content area is titled 'Analysis History' and displays a single analysis card for 'Q1 Onboarding Workshop — agenda + next steps'. The card shows it was created on 06/01/2026 at 20:33:11 and is marked as 'Analyzed'. A 'View Analysis' button and a trash bin icon are also present.

## Dark mode

The screenshot shows the Neatly interface in Dark mode. The sidebar menu is identical to the Light mode version. The main content area starts with a 'Welcome back' message: 'Here's what's happening with your analyses'. It features three cards: 'Total Analyses' (1 email threads analyzed), 'Pending To-Dos' (4 tasks to complete), and 'Upcoming Events' (0 events in the next 7 days). Below this are two larger sections: 'Recent Analyses' (listing 'Q1 Onboarding Workshop — agenda + next steps' from 1/6/2026) and 'Pending To-Dos' (listing four tasks with descriptions like 'Confirm recording of the workshop for internal use only and not for ...' and 'Send estimated cost to Priya Shah').

## Appendix 12 - Justification for key design decisions

Component	What is it?	Why was it chosen?	Strengths	Trade offs / Risks	Mitigations
NextJS (UI + API)	Single framework for the web UI with integrated backend API routes (server actions / route handlers)	Keeps frontend and backend in one codebase reducing the need for UI & API deployment. It is fast and has great community support.	Server-side-rendering for performance and SEO. TypeScript-friendly, easy auth/session patterns and strong community.	Can blur responsibilities (UI & API), with risk of bloated server routes increasing security risks	Clear route layering with controller & services. Caching and pagination to avoid large requests and API validation and verification.
Vercel	Cloud hosting platform	Integrated with GitHub providing one click deployments. Developer friendly and good community support.	Simple one click deploys. Integrated analytics.	Vendor lock, pricing scales with usage substantially more than self-hosting as well as NextJS serverless function limits and rate limits.	Neatly will be kept portable with local env files and usage will be monitored to prevent unwanted bills.
Supabase	Managed Postgres database with authentication	It acts as a backend-as-a-service (BAAS) combining auth and Postgres database (DB) strengths. Postgres is developer standard for structured data and Supabase has amazing community support.	Provides SQL integration, Row-level-security providing data security, built in authentication reducing coding burden and cheap.	Vendor lock, pricing increases with usage and Postgres still has to be managed effectively as it is not a done-for-you service provider.	Efficient storage of data ensures data normalisation. Indexes and views will be created to speed up DB queries reducing load on the Postgres instance.
ChatGPT & AgentKit	LLM and an agent builder	LLMs are constantly ranked as number 1. AgentKit is new however it provides a simple interface to build agentic flows that can be connected directly via their agent kit SDK in code.	Strong reasoning and language processing. Scores highly on benchmarks (Vellum AI, 2025)	Has a tendency to hallucinate affecting response quality. Using a third party introduces risks of upstream downtime which will break Neatly.	Structured JSON outputs will be used as well as human-in-the-loop practices to prevent hallucinations.

## Appendix 13 - Content research agent definition

```

const contentReserachAgent = new Agent({
  name: "Content reserach agent",
  instructions: `
    You are the Content Research Agent.
    Your job is to do light web research on a topic and
    return a Research Notes document in plain text/Markdown (NOT JSON).

    Requirements
    - Use web search.
    - Summarize the most important facts, definitions, and key takeaways.
    - Keep it concise and skimmable.
    - Always include a "Sources" section at the end with link name + URL only.

    Output format (exact sections)
    Return in this order:
    1) Topic
    2) Scope (what you covered)
    3) Overview (4–6 sentences)
    4) Key facts (8–15 bullets)
    5) Definitions (5–12 bullets)
    6) Recent developments (0–8 bullets, include dates if relevant)
    7) Numbers (0–10 bullets, include context + date if relevant)
    8) Pros / Cons / Tradeoffs
    9) Risks (reliability + privacy/security) + mitigations
    10) Open questions
    11) Sources (required)

    Sources section rules (important)
    - Provide 5–12 sources when possible.
    - Format each source as one line:
    - {Link Name} – {https://...}
    - No extra fields (no publisher, no accessed date, no commentary here).
    - Don't invent links. If you truly can't find sources, write:
    - None found

    Failure handling (mandatory)
    If a web search tool call fails or returns weak results, do NOT output "Research could not be completed...".
    Instead, return a partial Research Notes response using the required headings, with:
    - fewer bullets,
    - and a Sources section that includes whatever links were found (or – None found).
    Also add:

    Formatting contract
    You MUST include the headings exactly:
    Topic, Scope, Overview, Key facts, Definitions, Sources
    Even if some sections are short/empty.
  `,
  model: "gpt-4o",
  tools: [
    webSearchPreview
  ],
  modelSettings: {
    temperature: 1,
    topP: 1,
    maxTokens: 2048,
  }
});
  
```

```

// Tool definitions
const webSearchPreview = webSearchTool({
  filters: {
    allowedDomains: [
      "www.wikipedia.org",
      "google.com"
    ]
  },
  searchContextSize: "low",
  userLocation: {
    country: "GB",
    type: "approximate"
  }
})
  
```

## Appendix 14 - Content formatting agent definition

```
// The Formatting agent takes the research history + its instructions to formatted output
const contentFormattingAgent = new Agent({
  name: "Content formatting agent",
  instructions:
    You are the Content Formatting Agent.

    Input is research text produced by the research agent.
    Your job is to transform it into UI-ready JSON that is clean, structured, digestible, and designed to be read quickly.

    Output requirement
    - Output ONLY valid JSON that matches the FormattedBrief schema (below).
    - No Markdown. No code fences. No commentary. No extra keys.
    - NEVER use "/" as a placeholder value for empty fields. Use an empty string "" or omit the field if optional.

    Hard rules
    - No new facts.
    - Only rephrase, reorganize, deduplicate, and clarify what's present in the research text.
    - If something is missing, add it only as a question in FAQ / Open questions style sections, not as a claim.

    Return sources
    - Extract the Sources section from the research text.
    - Output sources: [{ id, name, url }] with ids s1, s2, ... in order.
    - Each source must have name + url only.
    - If a URL is missing scheme but clearly a domain, prepend https://.
    - Drop malformed lines if you can't confidently fix them.
    - sources must exist even if empty.

    Make it quick to read (your main value)
    - Default to short bullets over paragraphs.
    - Prefer strong headings and chunking.
    - Avoid repetition and filler.

    Cleaning + structuring steps (do these in order)

    Step A – Parse & normalize
    - Identify the research sections (Topic, Overview, Key facts, Definitions, Recent developments, Numbers, Pros/Cons, Risks, Open questions, Sources).
    - Normalize casing, fix obvious typos, remove duplicated bullets, merge near-duplicates.
    - Convert long sentences into tight bullets.

    Step B – Extract “signal”
    - Choose the 10–15 highest-signal points as Key points.
    - Convert the overview into:
    - a short TL;DR (max 5 bullets),
    - and a short “What it is” paragraph if needed.

    Step C – Write for scanning (UI-first)
    - Keep most blocks to:
    - 3–7 bullets per block
    - 1 short paragraph max when necessary
    - Use simple language and concrete phrasing.
    - Avoid nested bullets deeper than 1 level (UI readability).

    Step D – Build sections
    - Create sections in this default order (skip if empty):
    1) why_it_matters
    2) what_it_is
    3) key_points (group into 3–6 subheadings using heading + bullets)
    4) whats_new (dated bullets)
    5) numbers (table if possible; else bullets)
    6) pros_cons (two bullets lists + tradeoffs)
    7) risks_mitigations (reliability + privacy/security + mitigations)
    8) recommendations (prioritized, action-oriented)
    9) faq (Q/A pairs)
      - IMPORTANT: for each question, you MUST provide a meaningful answer based on the research.
      - If the research doesn't answer it directly, synthesize an answer from related information or note it's unclear.
      - Empty answers are NOT allowed.

    Step E – Recommendations (must be actionable)
    - Produce 5–10 next steps.
    - Prefer verbs: “Define…”, “Measure…”, “Add…”, “Restrict…”
    - If the topic is technical, include “how to validate” and “rollout safety” bullets.

    Step F – Citations (optional)
    - If the research text clearly maps a point to a source, attach citations.
    - If not confident, omit citations; do not guess.

    Output fields to populate
    - schema_version: "1.0.0"
    - title: short, informative
    - subtitle: optional
    - topic: from input
    - tldr: 1–5 bullets
    - sections: array of sections with blocks
    - sources: extracted sources list

    Parsing + fallback rules (add verbatim)
    - If headings exist, parse by headings:
    Topic / Scope / Overview / Key facts / Definitions / Recent developments / Numbers / Pros / Cons / Risks / Open questions / Sources.
    - If headings are missing, fall back to heuristic extraction:
    - topic: first non-empty line (or “Untitled topic”)
    - tldr: summarize the first ~15 lines into 3–5 bullets
    - key_points: extract 8–12 bullets from anywhere in the text
    - sources: extract URLs from anywhere in the text (not only “Sources” section)
    - Never output a single “Status/error” section unless the input is empty or only contains an error message.
    - If the input contains an error message, still output a normal brief with:
    - a callout block warning that research was incomplete,
    - sources: [],
    - and reasonable placeholder sections (why_it_matters / recommendations / faq) based on the user’s topic.
    - Do not use section.id = “error” in normal operation.
    - Only use that if you truly cannot extract any content.

  },
  model: "gpt-4o",
  outputType: formattedBriefSchema, // Using our app's defined schema
  modelSettings: {
    temperature: 1,
    top: 1,
    maxTokens: 8192,
  }
});
```

## Appendix 15 - Binary Risk Analysis

### Reference risk assessment

ID	Risk	Description
R1	Unauthorised Data Access	Authentication flaws could let other users view other users' sensitive data.
R2	Unauthorised API Access	If API endpoints are left unprotected, malicious users could access or manipulate other users' data.
R3	AI key theft	The ChatGPT credentials can be stolen leading to increased costs and data leaks.
R4	AI Hallucination	Inaccurate AI outputs could mislead users, possibly affecting quality of output affecting company reputation.
R5	Third party outages	Vercel, Supabase and ChatGPT could have downtimes making Neatly inoperable.
R6	Data leaks	Sensitive information could be accidentally displayed in server logs or browser cache.

### Binary Risk Analysis worksheet

Risk ID	Risk likelihood						Risk impact				Result		
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Likelihood	Impact	Overall
R1	Y	N	N	N	N	N	Y	Y	Y	Y	Low	High	Medium
R2	Y	N	N	N	N	N	Y	Y	Y	Y	Low	High	Medium
R3	Y	N	N	N	N	N	Y	N	N	N	Low	Low	Low
R4	Y	Y	N	N	N	Y	Y	N	Y	N	Medium	Medium	Medium
R5	Y	Y	N	N	Y	Y	Y	Y	Y	N	High	High	High
R6	Y	N	N	N	N	N	Y	Y	Y	N	Low	High	Medium

## Binary Risk Analysis worksheet process

The process is split into 2. Likelihood of risks and impact of risks. The results for each risk is not a simple count of the Y/N answers. The process details can be found here as well:

<https://binary.protect.io/workcard.pdf>

2x2 outputs

NN = Low

YN/NY = Medium

YY = High

3x3 outputs

&	Low	Medium	High
Low	Low	Medium	Medium
Medium	Medium	Medium	High
High	Medium	High	High

Likelihood matrices

### **Matrix A Threat Scope:**

Uses Q1 + Q2 and outputs S/M/L (small/medium/large attacker pool).

### **Matrix B Protection Capability:**

Uses Q3 + Q4 and outputs C/P/I (complete/partial/incomplete defence).

### **Matrix C Attack Effectiveness:**

(3x3) Uses matrix A + B output and outputs L/O/C (limited/occasional/consistent).

### **Matrix D Occurrence:**

Uses Q5 + Q6 and outputs R/P/A (rarely/periodically/always).

### **Matrix E Threat Likelihood:**

(3x3) Uses matrix C + matrix D output and outputs LL/ML/HL (low/moderate/high likelihood).

Impact matrices

### **Matrix F Harm:**

Uses Q7 + Q8 and outputs L/M/D (limited/material/damaging)

### **Matrix G Valuation:**

Uses Q9 + Q10 and outputs P/S/E (peripheral/supporting/essential).

### **Matrix H Impact:**

(3x3) Uses matrix F + matrix G outputs and outputs Li/Mi/Hi (low/moderate/high impact).

## Binary Risk Analysis Questions

- Q1:** Can the threat event be completed with common skills?
- Q2:** Can the threat event be completed without significant resources?
- Q3:** Is the asset undefended?
- Q4:** Are there known weaknesses in the current defences?
- Q5:** Is the vulnerability in the asset always present?
- Q6:** Can the threat event occur without meeting pre-conditions?
- Q7:** Will there be consequences from internal sources?
- Q8:** Will there be consequences from external sources?
- Q9:** Does the asset have or create significant business value?
- Q10:** Will the repair or replacement costs be significant