TASKIT APP PLAN

By: Vaibhavi Bhatt

PHASE 1: IDEATION

STEP 1: PRODUCT AND USER BASICS

1.1 Product one-liner

- **Product name**: **TaskIt**
- **One-liner**: *For busy team leads, TaskIt is a simple AI tool that turns long meeting notes into clear, assigned and trackable next steps in minutes, so they can close every follow-up without retyping notes or chasing people for updates.*

1.2 Who is this for

- **Primary user:** Team lead or project manager who runs 5+ meetings a week.
- **Context:** Right after a meeting, with rough notes or a pasted transcript, needing to send out who-does-what.
- **Top 3 pains today:**
    a. Action items stay buried in notes; nobody re-reads them.
    b. Owners and deadlines are vague ("someone should…", "soon"), so tasks slip.
    c. Turning notes into tasks in a tracker is slow, manual retyping.
    d. Team leads waste time chasing people for updates because there's no single place to see who's doing what.

1.3 MVP success for this hackathon

- **A user should be able to:** Paste meeting notes, get AI-drafted action items with owners and deadlines, review and edit them, and track them on a To do → In progress → Done board.
- **Demo success metric:** One set of sample notes becomes a reviewed board of 5+ tasks, each with an owner and deadline, in under 2 minutes.

STEP 2: 5 WHYS

2.1 Problem v0: Team leads lose track of action items after meetings.

2.2 5 Whys

Why #	Question	Answer
1	Why is this a problem for your user?	Follow-ups get missed, projects slip, and the lead spends time chasing people for updates.
2	Why does that happen right now?	Action items sit in free-text notes with no clear owner or date.

Why #	Question	Answer
3	Why haven't existing tools solved this well?	Meeting AI tools summarize but don't make owners and dates reviewable; task tools need manual entry.
4	Why is your user still stuck?	Turning notes into assigned tasks takes 15–30 minutes, so it gets skipped, and with no shared view, the lead has to chase each person for status.
5	Why does solving this now matter for them?	Teams run more meetings (often remote); every lost follow-up costs time and trust.

2.3 Final problem statement: Busy team leads lose follow-ups after meetings because action items stay buried in notes without clear owners or deadlines, and converting them into tracked tasks is too slow to do consistently.

STEP 3: COMPETITOR SCAN

3.1 Reference products (Mark: add screenshots of pricing and feature pages)

Product	Pages checked	What they do well	What frustrates users
Otter.ai	Pricing, review	Live transcription; action items with named owners when the transcript makes it clear	Free plan capped at 300 min/month; owner attribution needs manual checks in big calls
Fireflies.ai	Review, pricing	Auto-joins calls; structured summaries with action items	Action items stay inside the summary; paid plans from $10/user/month
Fathom	Pricing	Free unlimited recording; clean summaries	AI action items capped at 5 calls/month on free; no task tracking after the meeting

3.2 AI feature table (summary)

Feature	What it does	Why users care	Tools
Transcription	Records and transcribes calls	No manual note-taking	All
AI summary	Condenses the meeting	Quick recap	All
Action item extraction	Lists to-dos from the meeting	Captures follow-ups	All
Owner detection	Names who owns each item	Accountability	Otter

Feature	What it does	Why users care	Tools
CRM sync	Pushes notes to Salesforce/HubSpot	Sales workflows	Otter, Fathom (paid)
⭐ Review + edit before saving	User confirms owners and dates	Trust in AI output	None found
⭐ Built-in task board	Tracks items to Done	Follow-ups actually close	None found

Pattern: Every tool stops at the summary.

Our differentiators: works from any pasted notes (no bot), a human review step, and a board to track completion.

Sources checked Oct 29, 2026; pricing from third-party reviews, so verify on each official site.

STEP 4: MVP SCOPE (MOSCOW)

4.2 MoSCoW

Feature	M	S	C	W	Differentiator?	Why here
Paste notes + meeting details (title, date, time, attendees)	☑	☐	☐	☐	-	Entry point; attendees let AI match owners.
AI extracts action items	☑	☐	☐	☐	⭐	The core promise.
AI fills owner + deadline when mentioned	☑	☐	☐	☐	⭐	Named in the brief.
Defaults: no owner → team lead; no deadline → 5 business days; both flagged "AI-assigned, please review"	☑	☐	☐	☐	⭐	No task left without an owner or date; flag keeps the human in control.
Review screen: edit, delete, add items before confirming	☑	☐	☐	☐	⭐	Human in the loop.
Task board: To do → In progress → Done	☑	☐	☐	☐	-	Tracking is in the brief.
Edit / delete / move cards	☑	☐	☐	☐	-	Basic CRUD.

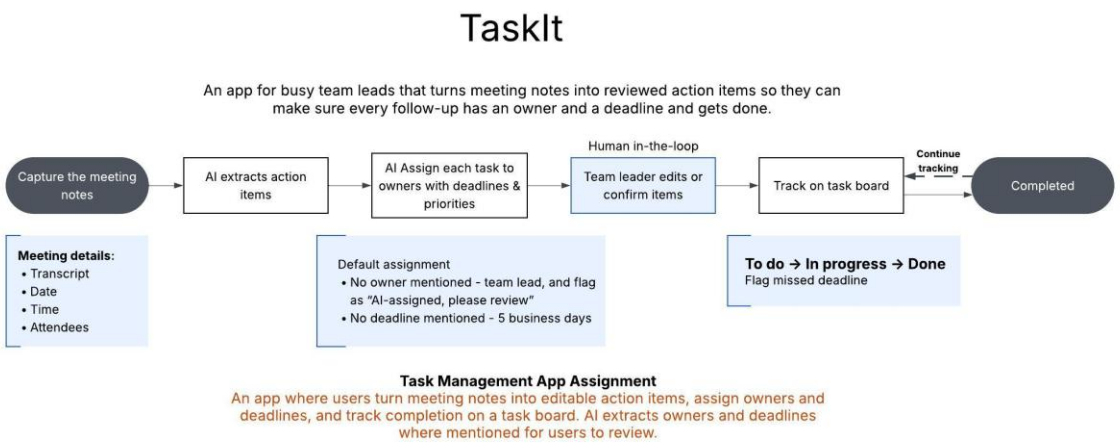
Feature	M	S	C	W	Differentiator?	Why here
Flag missed deadlines on the board	☐	☑	☐	☐	-	Makes slipping tasks visible.
Priority (High / Med / Low) suggested by AI	☐	☑	☐	☐	-	Useful, but a guess, so not core.
Filter board by owner	☐	☐	☑	☐	-	Nice once there are many tasks.
Reopen a completed task	☐	☐	☑	☐	-	Drag back from Done.
Card links back to source meeting	☐	☐	☑	☐	-	Context, not needed for demo.
Audio upload / transcription	☐	☐	☐	☑	-	Heavy; users can paste transcripts.
Email / Slack / calendar integrations	☐	☐	☐	☑	-	Out of 48-hour scope.

4.3 Final MVP scope

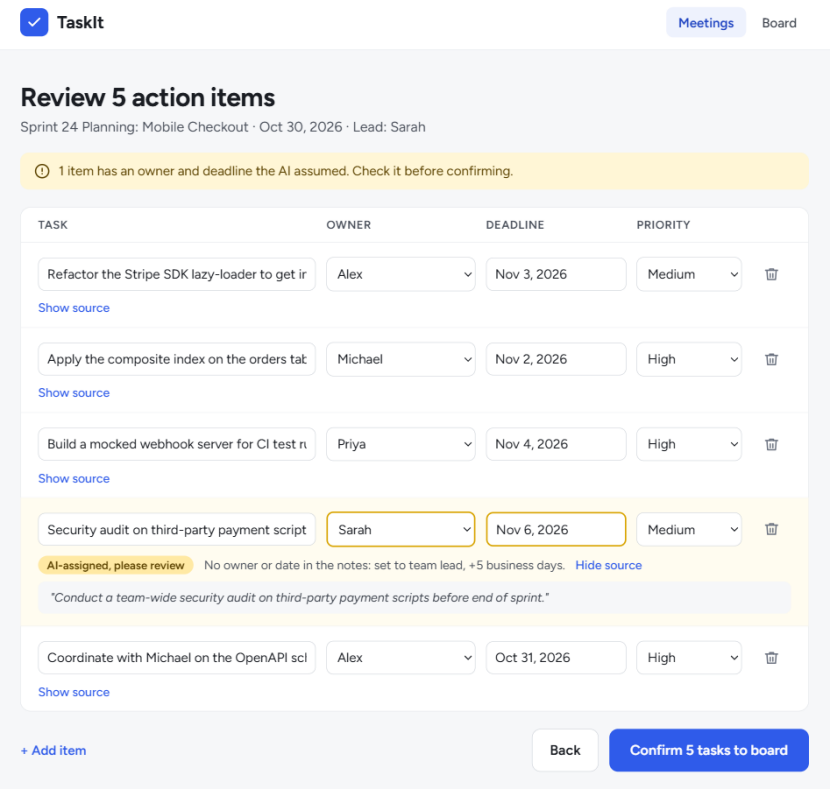
- **Must-have flow:** Enter meeting details + notes → AI drafts items with owners and deadlines (defaults flagged) → team lead reviews and confirms → items tracked on the board to Done.
- **Should-haves:** Missed-deadline flag; AI-suggested priority.
- **Parked:** Filter by owner, reopen, meeting links, audio, integrations.

PHASE 2: BUILDING

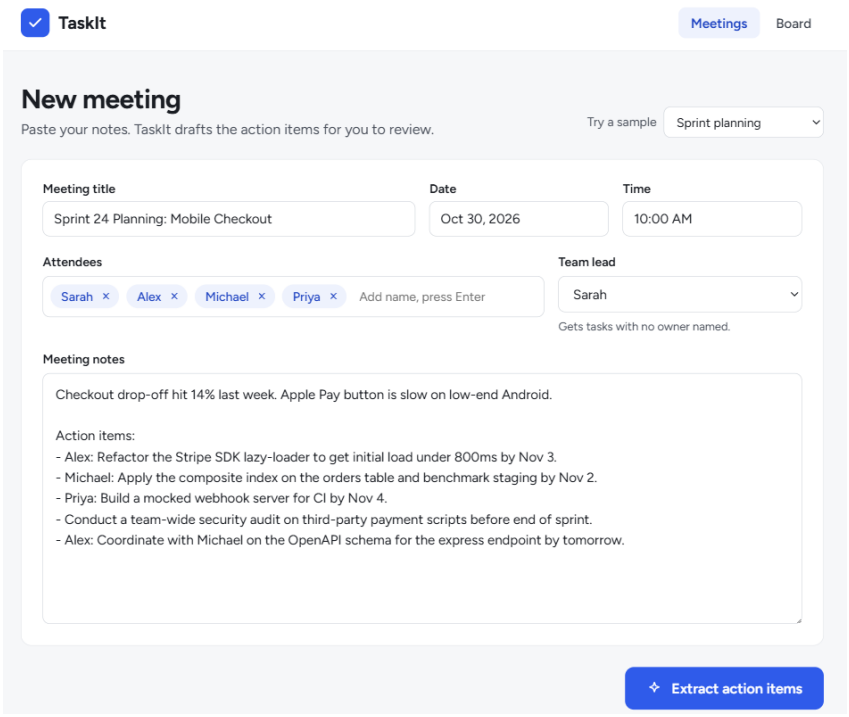
STEP 5: ARCHITECTURE & MOCKUPS



New meeting screen (mockup 1/2)



Review action items screen (mockup 2/2)



5.1 Big picture

- **Front-end:** React (Vite) web app built in Google Antigravity. Screens: Meetings, New Meeting, Review Items, Task Board.
- **Back-end:** One Vercel serverless function, `/api/extract`, that sends notes + attendees to the AI and returns action items as JSON. Keeps the API key off the browser.
- **Data store:** Browser localStorage for the MVP (no setup, full CRUD). Upgrade to Supabase only if time allows.
- **External services:** Google Gemini API (free tier key from Google AI Studio); Vercel for hosting; GitHub for code.

5.2 Screens and actions

Screen	Who uses it	Key actions	Data shown
Meetings	Team lead	Start new meeting, open past meeting	Title, date, # of tasks
New Meeting	Team lead	Enter title, date, time, attendees; paste notes; click "Extract action items"	Meeting form, notes box
Review Items	Team lead	Edit task, owner (attendee dropdown), deadline, priority; delete; add; "Confirm to board"	Draft items, "AI-assigned, please review" flags
Task Board	Team lead	Drag cards To do → In progress → Done; edit; delete	Cards with task, owner, deadline, priority, overdue flag

STEP 6: PRD

- **Goal:** Turn one set of meeting notes into a reviewed, assigned, trackable task list in under 2 minutes.
- **Users:** Team leads and project managers who run frequent meetings.
- **Scope:** Must-haves from 4.3 + missed-deadline flag + AI-suggested priority.
- **Out of scope:** Audio, integrations, multi-user login, notifications.

Core flows

1. New meeting → paste notes → Extract → Review → Confirm → Board.
2. Edit or delete an item (review screen or board).
3. Move a card To do → In progress → Done.

AI rules

- Only extract tasks stated in the notes; never invent tasks.
- Owner: match to an attendee if named; else team lead, flagged.
- Deadline: convert phrases ("next Friday") to a date from the meeting date; else +5 business days, flagged.
- Return strict JSON.

Data model

- **Meeting:** id, title, date, time, attendees[], teamLead, notes, createdAt
- **ActionItem:** id, meetingId, title, owner, ownerSource (ai / default / user), deadline, deadlineSource, priority, status (todo / in_progress / done), createdAt

Non-functional: Works on laptop and phone width; AI errors show a clear message, never a blank screen; API key stored only in Vercel env vars.

STEP 7: PLATFORM AND STARTING PROMPT

7.1 Platform

- **Build tool:** Google Antigravity (agent-based code editor).
- **Hosting:** Vercel, connected to the GitHub repo (auto-deploys on every push).
- **AI model:** Gemini API; key in a `.env` file locally and in Vercel → Settings → Environment Variables as `GEMINI_API_KEY`.
- **Working notes:** Use Planning mode for the first prompt and review the agent's plan before it builds. Build one screen at a time, and test each in the browser before the next prompt. Commit to GitHub after each working step.

7.3 Starting prompt (paste into Antigravity)

```
Build a responsive web app called "TaskIt" with React + Vite + Tailwind, deployable
on Vercel.
Purpose: team leads paste meeting notes; AI extracts action items with owners and
deadlines; the lead reviews them; tasks are tracked on a board.

DATA (store in localStorage):
- Meeting: id, title, date, time, attendees (string[]), teamLead, notes, createdAt
- ActionItem: id, meetingId, title, owner, ownerSource ("ai"|"default"|"user"),
deadline (YYYY-MM-DD), deadlineSource ("ai"|"default"|"user"), priority
("High"|"Medium"|"Low"), status ("todo"|"in_progress"|"done"), createdAt

SCREENS:
1. Meetings: list past meetings (title, date, task count) + "New meeting" button.
2. New Meeting: fields for title, date, time, attendees (comma-separated), team lead
(pick from attendees), and a large notes textarea. Button "Extract action items".
3. Review Items: editable table of AI-drafted items (title, owner dropdown of
attendees, deadline date picker, priority). Show a yellow "AI-assigned, please
review" badge when ownerSource or deadlineSource is "default"; editing a field sets
its source to "user". Allow delete and "Add item". Button "Confirm to board" saves
items with status "todo".
4. Task Board: three columns To do / In progress / Done. Cards show title, owner,
deadline, priority. Drag-and-drop between columns (use @dnd-kit) plus a status
dropdown fallback. Edit and delete on each card. Red "Overdue" tag if deadline <
today and status != done.

AI (Vercel serverless function at /api/extract.js):
- POST {notes, attendees, teamLead, meetingDate}. Call Google Gemini (model gemini-
2.5-flash) with GEMINI_API_KEY from env.
- Instructions: extract only tasks stated in the notes, never invent tasks. Match
owners to attendees; if none named, owner = teamLead and ownerSource = "default".
Convert relative dates using meetingDate; if none, deadline = meetingDate + 5
```

```
business days and deadlineSource = "default". Suggest priority.
- Return strict JSON: {"items":[{title, owner, ownerSource, deadline, deadlineSource,
priority}]}. Validate it; on any error return a clear message.

UX: clean, simple layout; loading state while extracting; friendly error messages;
works at phone width.
Add a "Load sample notes" button on New Meeting for demos.
```

STEP 8: DAY 1 AND DAY 2 EXECUTION

Day 1 (Thu Oct 29 to Fri Oct 30, 3 pm)

- ☒ Ideation done and frozen
- ☒ Architecture sketched
- ☒ PRD written
- ☒ Starting prompt ready
- ☐ Gemini API key created in Google AI Studio
- ☐ GitHub repo created; teammates added (Mark)

Day 2 (Fri Oct 30, build)

- ☐ Antigravity project created; starting prompt run
- ☐ Data model working in localStorage
- ☐ `/api/extract` returns items for sample notes
- ☐ Must-have flow runs end-to-end once
- ☐ First deploy to Vercel; env var set; live link works
- ☐ Code pushed to GitHub

PHASE 3: BACKEND, AUTOMATIONS, ITERATION AND DEMO

STEP 9: DATABASE AND CRUD

9.1 Entities (localStorage keys `taskit_meetings`, `taskit_items`)

- **meetings:** id, title, date, time, attendees, teamLead, notes, createdAt
- **action_items:** id, meetingId, title, owner, ownerSource, deadline, deadlineSource, priority, status, createdAt

9.2 CRUD coverage (tick as you test)

Entity	Create	Read	Update	Delete	Notes
meetings	☐	☐	☐	☐	Edit title/notes optional for MVP
action_items	☐	☐	☐	☐	Via Review screen and Board

If a box stays empty, prompt Antigravity: "Enable full CRUD for [entity]. Add [missing operations]. Keep UX consistent with existing screens."

STEP 10: AUTOMATIONS AND AI

10.1 Backend and automations

Trigger	Tool / service	What happens
"Extract action items" clicked	Vercel function + Gemini	Send notes, attendees, team lead, date; get items as JSON; open Review screen
Item has no owner or deadline	Gemini + app rules	Apply team lead / +5 business days; set "AI-assigned, please review" badge
"Confirm to board" clicked	App + localStorage	Save items with status To do
Board loads or day changes	App	Tag cards past deadline and not Done as Overdue
Code pushed to GitHub	Vercel	Auto-deploy new version to live link

10.2 Wow AI feature

- **Name:** Smart Owner & Deadline Detection
- **What it does:** Reads messy notes, matches "Priya will…" to an attendee, turns "next Friday" into a date, and flags anything it had to assume.
- **Prompt core:** "Extract only tasks stated in these notes. Match owners to this attendee list. Convert relative dates using the meeting date. If no owner, use the team lead and mark ownerSource 'default'. If no date, use +5 business days and mark deadlineSource 'default'. Return JSON only."

STEP 11: TESTING AND ITERATION

11.1 Scenario

A team lead finishes a 30-minute product sync with 4 attendees. She pastes the notes (some tasks with names and dates, some with neither), extracts items, fixes one flagged owner, confirms, then moves one task to In progress and one to Done.

New Meeting → Review Items → Task Board → done.

11.2 Bug and gap log

#	Issue	Where	Prompt used to fix	Fixed?
1				
2				

11.3 Stability checklist

- ☐ Must-have flow works twice in a row on the live Vercel link
- ☐ No dead buttons or dead-end screens
- ☐ CRUD works for meetings and action items
- ☐ AI never invents tasks on all 4 sample notes
- ☐ Defaults and flags appear when owner or date is missing
- ☐ AI error shows a message, not a blank screen

STEP 12: DEMO AND REFLECTION

12.1 Demo script (2–3 min Loom)

1. **User and problem:** "TaskIt is for busy team leads who lose follow-ups because action items stay buried in meeting notes."
2. **What we built:** "In 3 days we built TaskIt, which turns meeting notes into reviewed action items so every follow-up has an owner and a deadline and gets done."
3. **Live walkthrough:**
    - Open the live link; create a meeting with 4 attendees.
    - Paste sample notes; click Extract.
    - **Wow moment:** AI matched owners and turned "next Friday" into a date; flagged two items it had to assume.
    - Fix one flagged owner; confirm to board.
    - Drag a card to In progress, another to Done; point out an Overdue tag.
4. **Close:** "Next, we plan to add team sign-in, Slack and email reminders, and audio upload."

Demo checklist

- ☐ Live Vercel link opens with no login
- ☐ Must-have flow runs live, no hidden setup
- ☐ Wow feature works on demo notes

12.2 Reflection (fill after the hackathon)

- What judges liked most:
- One choice that worked well:
- Top 3 issues:
- Next 7 days: improve stability and UX; add 1–2 parked features; test with 3–5 users; clean PRD for v2.