I want to submit a prototype for my mawin

What I can do

Great — submitting a prototype for **Mawin** is a solid step to show traction, capability, and clarity of your product direction. Since your product is a **plug-and-play AI agent platform for SMBs**, here's a tailored plan for what you can realistically submit as a **prototype**, even if you're still early:

✅ Minimum Viable Prototype (Web-Based)

You don't need to build everything. Focus on 1 agent that works **end-to-end** for 1 small business use case. This will demonstrate:

- Your core idea (easy-to-use AI agent for SMBs)
- Your ability to deploy fast
- The interface simplicity and onboarding experience

🧠 Choose 1 Agent to Implement

Pick one AI agent that's simple but impactful. Examples:

- **Email Auto-Responder**: Connect to a business Gmail, summarize inquiries, auto-draft responses
- **Customer Chatbot**: FAQ bot that works on their site or WhatsApp
- **Lead Scoring Agent**: Scores incoming leads from forms or emails
- **Social Media Post Generator**: Weekly post drafts from uploaded business updates

(Choose based on what's easiest to demo in isolation and still useful)

🧱 Build Prototype Structure

1. Landing Page (Mawin)

- Brief pitch: "Plug-and-play AI agents for your business. No setup needed."
- List 1–3 example agents with icons
- CTA: Try Agent Now (or Login to Dashboard)

2. Agent UI

Choose one and build a simple flow:

- Input: Form or connection (upload email, connect Gmail, enter website URL, etc.)
- Output: Response from agent (e.g., email reply, customer support answer, draft tweet)

- Allow the user to “edit + approve” the output

You can build this with:

- **Frontend**: React (or use Webflow with embedded iframe if very early)
- **Backend**: Flask, FastAPI, or Bun (since you’re using it)
- **Agent logic**: OpenAI API with prompt chaining or n8n/Make if you’re using automation tools
- **Storage**: Supabase (already in your stack)

3. Basic Dashboard (Optional)

- Simple user auth (Supabase)
- Saved past actions
- Allow user to re-run or fine-tune outputs

📦 Submission Materials

1. **Live prototype URL** (host on Vercel, Netlify, or Railway)
2. **Demo video (2–3 min)** – Show:
    - Your landing page
    - Using the agent end-to-end
    - Simplicity of the workflow
3. **One-pager or deck** that summarizes:
    - Problem
    - Mawin solution
    - Demo screenshots
    - How it works technically
    - What’s next

🔧 Tools You Can Use

Purpose	Tool Options
Backend API	FastAPI, Flask, Bun
Frontend	React (with Tailwind), Webflow
Database/Auth	Supabase
AI APIs	OpenAI, Gemini, Claude
Automation	n8n, Make.com
Hosting	Vercel (frontend), Railway (backend)
Recording Demo	Loom, OBS, Screen Studio

Want Help With...?

Let me know:

- Which agent you want to build?
- Do you want code scaffolding / prompt design help?
- Do you need a sample demo script?

Once I know the agent you're picking, I can help you create a basic demo fast.

Give me a basic agent I can implement

Here’s a **basic but impressive AI agent** you can implement quickly for your Mawin prototype:

📬 Agent: Email Inquiry Summarizer + Auto-Responder

🎯 What it does:

For small business owners who get customer inquiries (via email or contact form), this agent:

1. **Summarizes** the incoming message
2. **Suggests a professional reply**
3. Lets the user **edit or approve** the reply

👥 Why it works:

- Many SMBs struggle to respond quickly/professionally
- It’s easy to demo
- Looks impressive but technically simple
- Fits your value prop: "AI agent, no setup, saves time"

🔧 How It Works (MVP Flow)

1. User uploads an email or types a message they received

```txt
"Hi, I’d like to know if you offer catering for a party of 25 on July 21st in the evening."
```

---

2. Your app sends this to OpenAI (or Claude)

Prompt (sample):

```text
You are a customer service assistant for a small business.

Step 1: Summarize the customer's request in one sentence.
Step 2: Draft a polite and professional email reply.

Customer message: {{input}}
```

---

3. Display the result like this:

✅ Summary:

> "Customer is inquiring about catering services for a party of 25 on July 21st."

✉️ Suggested reply:

> "Hi there, thank you for reaching out! Yes, we do offer catering services, and we'd be happy to help with your party on July 21st. Could you please share the location and any specific menu preferences? Looking forward to assisting you!"

---

4. User can:

- ✅ Approve and copy
- ✏️ Edit it manually
- 🔁 Regenerate with another tone

🧱 What You Need to Build

Component	Tech Option
UI Form	React + Tailwind or plain HTML
Backend API	Bun, Flask, or FastAPI
AI API	OpenAI `gpt-3.5-turbo`
Hosting	Vercel (frontend), Railway (API)
State mgmt	Local state or Supabase (optional)

🖥️ Bonus: Simple Frontend Layout

- **Left column**: Input box (paste email)
- **Right column**: Summary + Drafted reply
- Button: "Regenerate" / "Copy to Clipboard" / "Edit"

Want a starter codebase?

I can give you:

- A Bun backend route that takes in a message and returns a summary + draft
- A React component that shows input > response
- Deployment guide

Just say the word.

Make a architecture and tasks document for an LLM like cursor to make this

So like the website for Mawin, with a page for the agents

I want to first give it to lovable for front end

Perfect. Below is a structured **Architecture + Task Document** you can give to **Lovable (frontend)** or an LLM like Cursor to **build the frontend of the Mawin prototype**, focusing on the agent directory and the Email Responder agent.

🏗️ Mawin Prototype – Architecture & Task Document (Frontend)

🧠 Project Summary

Mawin is a plug-and-play AI agents platform for SMBs. This prototype includes:

1. A landing page describing the product and listing available agents
2. A working flow for a basic agent: **Email Inquiry Summarizer + Auto-Responder**

🗂️ Pages & Routes

Page	Route	Description
Landing Page	`/`	Hero section, brief about Mawin, agent list
Agent Page	`/agents`	Directory of all available agents (just 1 for now)
Agent Detail	`/agents/email-responder`	UI for user to use the Email Responder agent

---

🎨 UI Tasks by Page

🔹 1. Landing Page (`/`)

Sections:

- **Navbar**: Mawin logo (left), links: Agents, About
- **Hero**: "AI Agents that Just Work" + CTA button → “Explore Agents”
- **Value Props** (3 tiles/icons): e.g. "No Setup", "Plug-and-Play", "Trusted Results"
- **Featured Agent**: Highlight Email Responder agent
- **Footer**

🔹 2. Agent Directory Page (`/agents`)

Layout:

- Title: “Explore AI Agents”
- Grid layout (Card-based) with:
  - Agent name
  - 1-liner description
  - Icon
  - Button: “Try Now” → links to `/agents/email-responder`

For now, include 1 card:

Field	Value
Title	Email Responder
Description	Summarizes inquiries and drafts email replies
Icon	✉️
Link	`/agents/email-responder`

🔹 3. Agent Interaction Page (`/agents/email-responder`)

Layout:

- **Heading**: “Email Inquiry Responder Agent”
- **Subheading**: “Paste a customer inquiry and get a ready-to-send response.”

UI Components:

Component	Description
Textarea Input	Multiline, placeholder: “Paste the email inquiry here...”
Submit Button	Label: “Generate Response”
Output Area	Once submitted, show:
	1. Summary (in card or bordered box)

Component	Description
	2. Drafted Email Response (in styled box, editable if possible)
Action Buttons	- Regenerate - Copy to Clipboard - Optional: Edit & Save

---

⚙️ API Integration Plan (Placeholder)

- Make a POST request to `/api/generate-response` with the pasted input
- Receive JSON:

```json
{
  "summary": "Customer is asking about catering services for July 21.",
  "reply": "Hi, thank you for reaching out! Yes, we do offer catering..."
}
```

- No authentication needed
- Placeholder response can be mocked using static JSON or a delay

---

🧰 Tech Stack Suggestions (Frontend)

Layer	Tool
Framework	React (Next.js optional)
Styling	Tailwind CSS
State Mgmt	React useState
API Requests	`fetch` or Axios
Hosting	Vercel

✅ Deliverables

- ☐ Fully responsive website with `/`, `/agents`, `/agents/email-responder`
- ☐ Working input/output interaction flow on agent page (mock or live API)
- ☐ Clean and simple UI – no login/auth needed
- ☐ Ready for handoff to backend for real-time response integration

---

📦 Bonus Add-ons (Optional)

- Simple loading spinner while response is generating
- Toasts for “Copied!” or error handling
- Dark mode toggle

Would you like me to generate:

- A folder structure for the frontend repo?
- Sample mock JSON data for the agent output?
- The backend endpoint next (Bun/OpenAI)?

Let me know and I'll deliver that too.