LogicPatch Project Analysis

Project Overview

Ye project **Figma** aur **Codebase (GitLab)** ke darmian ek "intelligent bridge" hai. Iska maqsad ye hai ke jab bhi koi Designer Figma mein design change kare, to automatically code update ho jaye, bina developer ke manual kaam kiye.

Core Architecture (Kaise Kaam Karta Hai)

System 4 main "pillars" par khara hai. Har pillar ka apna khaas role hai:

1. The Listener (Kaan)
* **File:** `webhook_server.py`
* **Kaam:** Ye har waqt Figma ki taraf kaan laga kar baitha rehta hai.
* **Process:**
* Figma se message aata hai: *"File update hui!"*
* Server kehta hai: *"Note kar liya."*
* Ye foran us message ko **Database** mein likh deta hai aur wapas sojaata hai.
* Ye khud koi heavy kaam nahi karta, sirf message leta hai.

2. The Brain (Dimaag)
* **File:** `mcp_core/services/llm_coder.py`
* **Kaam:** Code likhna. (Intehai Hoshiyar AI).
* **Process:**
* Isay batao: *"Ye button laal karo."*
* Ye Google Gemini (AI) se baat karta hai.
* Ye wapas **React/Next.js Code** likh kar deta hai.
* Ye bilkul senior developer ki tarah sochta hai (Best Practices, Material UI rules follow karta hai).

3. The Manager (Worker)
* **File:** `automation_worker.py`
* **Kaam:** Sab ko control karna.
* **Process:** (Ye loop mein 24/7 chalta hai)
1. **Check:** Kya Database mein koi naya kaam (update) hai?
2. **Assign:** Agar kaam hai, to **Brain (Coder)** ko bulata hai.
3. **Locate:** `RepoSearch` se poochta hai: *"Ye code kahan paste karun?"*
4. **Execute:** Code generate karwa ke, GitLab par **Merge Request** banata hai.
5. **Fix:** Agar code mein error aaye, to dobara Brain se theek karwata hai.

4. The Connector (Taar)
* **File:** `scripts/register_webhook.py`
* **Kaam:** Connection jorna.
* **Process:**
* Ye one-time setup hai.
* Ye Figma ko batata hai: *"Mera address (URL) ye hai, future mein yahan chithi bhejna."*

The Complete Flow (Kahani Ki Tarah)

LogicPatch Project Analysis

Imagine karein Designer ne Figma mein ek **"Login Button"** ka color Blue se **Red** kiya.
1. **Figma:** *"Update hua!"* -> **Message bheja**.
2. **Listener (`webhook_server`):** Message pakra -> **Database (`events.db`)** mein save kiya -> Sukoon se baith gaya.
3. **Manager (`worker`):** (Jo har 2 second baad check karta hai) -> *"Arrey! Naya kaam aya hai."*
4. **Manager:** Figma se **Image** aur **Text** mangwata hai.
5. **Manager:** Repository (`RepoSearch`) mein dhoondta hai: *"Login wala code kahan hai?"* -> Jawab milta hai: `src/components/LoginButton.jsx`.
6. **Brain (`llm_coder`):** Manager kehta hai: *"Oye Brain! Ye purana code hai, aur ye naya Laal Wala design hai. Naya code likho."*
7. **Brain:** Code likh kar deta hai.
8. **Manager:** Code ko test karta hai (Validate).
9. **Manager:** GitLab par **Merge Request** bhejta hai.
10. **Developer (Aap):** Aapko email aati hai: *"Bot ne Login Button update kiya hai, Approve karein?"*

Key Files Quick Reference

File	Role	Nickname
`webhook_server.py`	Receives updates	**Receptionist**
`init_db.py`	Creates storage	**Store Room Builder**
`automation_worker.py`	Controls everything	**Project Manager**
`llm_coder.py`	Writes Logic	**Expert Coder**
`register_webhook.py`	Connects to Figma	**Telephone Operator**
`simulate_webhook.py`	Fakes an update	**Drill Instructor (Testing)**

Next Steps for You

Ab jab aapko flow samajh aa gaya hai, aap ye kar sakte hain:
1. **Test Run:** `start_logicpatch.bat` chalayen to sab (Server + Worker) start ho jayega.
2. **Fake Test:** `scripts/simulate_webhook.py` chala kar dekhein ke system kaise react karta hai.
3. **Real Test:** Figma mein change karein aur magic dekhein!