Thank you for your interest in Ryze AI. We appreciate you taking the time to apply.

🧠 Full-Stack Assignment
AI Agent → Deterministic UI Generator (Claude-Code Style)
Timebox

⏱ 72 hours (3 days)
This is a time-boxed exercise. We value clarity of thought, correctness, and tradeoffs over polish.

🎯 Goal

Build an AI-powered agent that converts natural language UI intent → working UI code + live preview, using a fixed, deterministic component library.

Think: Claude Code for UI — but safe, reproducible, and debuggable.

👤 User Story

A user should be able to:
Describe a UI in plain English
See a working UI rendered immediately

Ask the AI to iteratively modify the UI

Understand why the AI made each decision

Roll back to previous versions

🔒 Core Constraint (Very Important)
Deterministic Component System

All UIs must use the exact same component library every time.

Component implementation and styling must never change

The AI may not create new components

The AI may only:

Select components

Compose layouts

Set props

Provide content

📌 Visual consistency is mandatory
📌 This is a correctness constraint, not a suggestion

🧱 Fixed Component Library

You must define (or implement) a fixed set of components, for example:

Button

Card

Input

Table

Modal

Sidebar

Navbar

Chart (mocked data allowed)

Prohibited

Inline styles

AI-generated CSS

Arbitrary Tailwind class generation

External UI libraries

New components created by the AI

🧠 AI Agent Requirements

A single LLM call is not acceptable.

You must implement explicit agent steps, at minimum:

1️⃣Planner

Interprets user intent

Chooses layout structure

Selects components

Outputs a structured plan

2️⃣Generator

Converts the plan into UI code

Uses only allowed components

Produces valid React code

3️⃣Explainer

Explains decisions in plain English

References layout and component choices

📌 Prompt separation must be visible in code
📌 Hard-coded prompt templates are allowed

🖥️ Required UI (Claude-Style)

Your app must include:

Left panel: AI chat / user intent

Right panel: Generated code (editable)

Live preview: Rendered UI

Required Actions

Generate UI

Modify existing UI via chat

Regenerate

Roll back to previous versions

📌 Live reload preferred
📌 Diff view is optional

🔁 Iteration & Edit Awareness (Key Evaluation Area)

The system must support incremental edits.

Example:

“Make this more minimal and add a settings modal.”

The AI must:

Modify existing code (not regenerate everything)

Preserve component usage

Explain what changed and why

❌ No full rewrites unless explicitly requested

🛡️ Safety & Validation

Include lightweight but real protections:

Component whitelist enforcement

Validation before rendering

Basic prompt injection protection

Error handling for invalid outputs

🧰 Technical Stack

You may choose any stack.

Common choices:

Frontend: React / Next.js

Backend: Node.js or Python

AI: Any LLM API

Storage: In-memory or lightweight DB

## 📦 Deliverables

Working application (local or deployed)

Git repository with commit history

README.md including:

Architecture overview

Agent design & prompts

Component system design

Known limitations

What you’d improve with more time

Optional: Short demo video (5–7 minutes)

## 🧪 Evaluation Criteria

| Area | What We Look For |
|---|---|
| Agent Design | Clear multi-step reasoning |
| Determinism | Components render identically |
| Iteration | Modifies code correctly |
| Explainability | AI decisions are understandable |
| Engineering Judgment | Good scoping & tradeoffs |

## 🚫 Explicitly Not Required

Authentication

Multi-user support

Pixel-perfect design

Accessibility audit

Production infrastructure

Mobile edge-case handling

⭐ Optional Bonus

Streaming AI responses

Diff view between versions

Component schema validation

Replayable generations

Static analysis of AI output

🧠 What This Assignment Tests

AI agent orchestration

Deterministic code generation

UI systems thinking

Iterative reasoning

Trustworthy AI design

📤 Submission Instructions (Required)

By the end of the 72-hour period, candidates must submit all of the following:

GitHub Repository

Public or private (grant access if private)

Must include full commit history

Clear setup instructions in README.md

Deployed Web Application

Hosted on any platform (e.g. Vercel, Netlify, Render, Fly.io, etc.)

Must be accessible via a public URL

Local-only demos will not be accepted

Demo Video (5–7 minutes)
The video must show:

Initial UI generation from natural language

Iterative modification via chat

Live preview updating

Explanation output from the AI

Rollback or version change

Screen recording with voice-over is preferred, but not required.

📧 Submission Method

Send one email to:

jayant@get-ryze.ai

Include:

GitHub repository link

Deployed app URL

Video link (Loom, YouTube, Drive, etc.)

Salary can be negotiated later

📌 Please put "AI UI Generator Assignment – [Your Name]" in the subject line.