FlowMind Al: Prompt Detalls and Development Plan

FlowMind Al is a workflow automation platform that allows users to visually create, manage, and execute workflows using a drag-and-drop interface. It integrates Al models, data processing tools, and third-party integrations to enable users to automate tasks, process data, and build Al-powered solutions without extensive coding.

Core Features

1. Drag-and-Drop Workflow Builder.

Users can create workflows by connecting nodes on a canvas.

Nodes represent specific actions or

operations (e.g., data import, AI processing, conditional logic).

2. Node-Based Architecture:

Data Import Nodes File Loader, API Loader, CSV Processor, URL Loader, etc.

Al Nodes: OpenAI, Anthropic, DALL-E, Whisper.

Logic Nodes:

Condition, Merge, Time.

Integration Nodes: Google Drive, Slack, Salesforce.

3. Al Model Integration:

Supports leading Al models like OpenAI

GPT-4, DALL-E, and others.

Enables tasks like text generation, image generation, and natural language processing.

4. Conditional Logic:

Nodes like the Condition Node allow users to define branching logic based on conditions (e.g. IF, ELSE IF, ELSE)

5. Real-Time Workflow Execution:

A

Executes workflows in real-time with support for multi-step pipelines and conditional logic.

6. Role-Based Access Control (RBAC):

Different roles (eg, Admin, Developer, Business User, Viewer) control access to features and nodes.

7. Marketplace for Pre-Built Workflows:

Includes a marketplace where users can browse and import pre-built workflows and templates.

8. Third-Party Integrations:

Integrates with tools like Google Drive, Salesforce, Slack, and GitHub for seamless workflow automation.

Development Plan

1. Frontend Development

Framework: React with TypeScript.

Key Components:

o FlowCanvas.tsx: Main canvas for creating workflows.

o NodePanel.tsx: Displays available nodes categorized by type.

o Toolbar.tsx: Provides workflow management actions (e.g., save, run, export).

o ConditionNode.tsx: Handles branching logic with dynamic input/output handles.

[] BaseNode.tsx: Reusable base component for all nodes.

State Management:

Use Zustand for managing application state (e.g., nodes, edges, user roles).

Styling:

Use Tailwind CSS for consistent and responsive styling.

2. Backend Development

eate

Framework: FastAPI (Python) or Node.js (JavaScript/TypeScript).

eNoc

cre

Key Responsibilities:

reat

Workflow Execution:

Implement a PipelineExecutor to process workflows sequentially.

Noc

Handle input/output data flow between nodes.

eate

API Integration:

crea

Provide endpoints for Al model execution (e.g., OpenAI, DALL-E).

atel

Manage third-party integrations (e.g., Google Drive, Salesforce).

Noc

Role Management:

Implement role-based access control (RBAC) for users.

Noc

Database:

Use PostgreSQL for structured data (e.g., workflows, user roles).

Use MongoDB for unstructured data (e.g., logs, templates).

3. Node Development

Node Types:

Data Import Nodes:

file-loader: Load files (CSV, JSON, XML, TXT, PDF).

api-loader: Fetch data from APIs.

csv-processor: Process CSV files.

url-loader: Scrape data from websites.

Al Nodes

openai: Use GPT-4 for text generation.

dalle: Generate images from text prompts.

whisper: Convert speech to text.

Logic Nodes:

condition: Branch workflows based on conditions.

merge: Combine multiple inputs into one.

Integration Nodes:

google-drive: Manage files in Google Drive.

slack: Send messages to Slack channels.

Node Configuration:

Define inputs, outputs, and icons for each node in index.tax.

Implement execution logic for each node in nodeExecutors.ts.

4. Role-Based Access Control (RBAC)

Roles:

Admin: Full access to all features and workflows.

Developer. Access to advanced features like low-code scripting.

Business User. Access to no-code features and pre-built templates.

Viewer: Read-only access to workflows.

Implementation:

Store user roles in a database or Zustand store.

Restrict access to nodes and features based on roles.

5. Marketplace for Pre-Built Workflows

Features:

Browse and import pre-built workflows (e.g., Al-powered search engines, chatbots).

Store templates as JSON files in a templates/ directory.

Implementation:

Add a Marketplace tab in the sidebar.

Implement a loadTemplate function to load templates into the canvas.

6. Deployment

Frontend:

Use Vercel or Netlify for deployment.

Backend:

Use AWS, Azure, or Google Cloud for

deployment.

Use Docker for containerization and Kubernetes for orchestration.

Key Files and Their Responsibilities

1. Frontend:

o FlowCanvas.tsx: Main canvas for creating workflows.

NodePanel.tex: Displays available nodes.

ConditionNode.tax: Handles conditional logic.

oindex.tax: Registers all node types.

2. Backend:

o pipelineExecutor.ts: Executes workflows sequentially.

o nodeExecutors.ts: Implements execution logic for each node.

o roles.ts: Defines roles and permissions.

3. State Management:

o flowstore.ts: Manages workflow state (nodes, edges, templates).

o userStore,ts: Manages user roles and authentication.

Target Customers

1. Non-Technical Business Users:

Automate tasks without coding.

Use pre-built templates for common workflows.

2. Developers and Data Scientists:

Build custom workflows with low-code scripting.

Integrate Al models and APIs.

3. Enterprises:

Scale operations with Al-powered workflows.

Integrate with tools like Salesforce and Slack.

4. Content Creators and Marketers:

Generate text, images, and videos using AI models.

Automate content creation and publishing.

Monetization Strategies

1. Subscription Plans:

Free, Pro, and Enterprise tiers with varying features.

2. Pay-As-You-Go:

Charge based on workflow executions or API usage.

3. Marketplace:

Sell pre-built workflows and templates.

4. White-Label Solutions:

Offer a customizable version of FlowMind AI for enterprises.

1. Core Settings Nodes

Total Nodes: 13

input

o output

text

o web-extractor

o youtube-transcript

o document-to-text

google-search

o http-get

ofile-upload

o file-save

note

transform

condition

2. LLM Nodes

Total Nodes: 6

[openai

anthropic

- gemini

cohere

[] perplexity

[] custom

3. Knowledge Base Nodes

Total Nodes: 4

o kb-reader

o kb-loader

o kb-search

kb-sync

4. Database Nodes

Total Nodes: 3

mysql

mongodb

github

. Productivity Tool Nodes

Total Nodes: 12

[] airtable

[] notion

[] hubspot

[] gmail

[] outlook

[] discord

google-drive

[] onedrive

google-docs

merge

[] time

[] ttsql

6. New Nodes (Data Import and Processing)

Total Nodes: 12

ofile-loader

o

api-loader

csv-processor

o

url-loader

o

wikipedia-loader

o

youtube-loader

orss-reader

google-alerts

o

serp-api-loader

o

A

S

arxiv-loader

rss-loader

o web-searchNavigation Categories

1. Core Settings: 13 nodes

日

2. Al Models (LLMs): 6 nodes

3. Knowledge Base: 4 nodes

4. Database: 3 nodes

5. Productivity Tools: 12 nodes

6. Data Import and Processing (New Nodes): 12 nodes