### **Project Idea: "ChainMate - The Blockchain Companion"**

#### **Description:**

ChainMate is an AI-powered social agent designed for blockchain enthusiasts and community members. It provides real-time insights, answers questions, and helps users make informed decisions by analyzing blockchain data. Integrated with platforms like Telegram and Discord, it acts as a community assistant.

#### **Key Features:**

1. **Real-Time Blockchain Insights**: Scans blockchain transactions and provides data on token movements, swaps, or new trends.
2. **Community Engagement**: Answers questions about blockchain concepts, projects, or market trends.
3. **Personalized Notifications**: Alerts users about specific events like wallet movements, price changes, or voting opportunities.
4. **Cross-Platform Availability**: Available on Telegram, Discord, and other platforms via Eliza.
5. **Mode Integration**: Utilizes Mode's ecosystem for secure and seamless operations.

### **Project Workflow:**

1. **Setup**: Use the Eliza framework to create the chatbot and integrate it with Telegram/Discord.
2. **Data Source**: Connect to blockchain APIs (e.g., Mode’s API, Etherscan) to fetch real-time data.
3. **AI Module**: Use a natural language processing (NLP) model (like OpenAI's GPT) to interpret queries and provide responses.
4. **Mode Integration**: Enable the agent to interact with Mode Network for secure DeFi operations or governance tasks.
5. **User Interface**: Create a conversational interface for Telegram/Discord.
6. **Testing**: Deploy and test the agent in real-world scenarios.

### **Code for ChainMate:**

Below is a Python script to build a Telegram bot using Eliza and integrate blockchain data insights.

python

Copy code

import requests

from telegram.ext import Updater, CommandHandler, MessageHandler, Filters

from eliza import Eliza

# Initialize Eliza framework

eliza = Eliza(api\_key="YOUR\_ELIZA\_API\_KEY")

# Blockchain API details

BLOCKCHAIN\_API\_URL = "https://api.mode.network/v1"

BLOCKCHAIN\_API\_KEY = "YOUR\_MODE\_API\_KEY"

# Fetch blockchain data

def get\_blockchain\_data():

url = f"{BLOCKCHAIN\_API\_URL}/blockchain\_data"

headers = {"Authorization": f"Bearer {BLOCKCHAIN\_API\_KEY}"}

response = requests.get(url, headers=headers)

if response.status\_code == 200:

return response.json()

return {"error": "Failed to fetch data"}

# Telegram bot command handlers

def start(update, context):

welcome\_message = "Hello! I'm ChainMate, your blockchain companion. Ask me anything about blockchain!"

update.message.reply\_text(welcome\_message)

def help\_command(update, context):

help\_message = """

Here's what I can do:

- /insights: Get real-time blockchain data insights.

- /alert <event>: Set up personalized alerts.

- /help: Get help on using ChainMate.

"""

update.message.reply\_text(help\_message)

def insights(update, context):

data = get\_blockchain\_data()

if "error" in data:

update.message.reply\_text("Unable to fetch blockchain data. Please try again later.")

else:

insights\_message = f"Latest Blockchain Insights:\n{data}"

update.message.reply\_text(insights\_message)

def handle\_message(update, context):

user\_message = update.message.text

response = eliza.chat(user\_message)

update.message.reply\_text(response)

# Main function

def main():

# Telegram bot token

TOKEN = "YOUR\_TELEGRAM\_BOT\_TOKEN"

updater = Updater(TOKEN, use\_context=True)

dp = updater.dispatcher

# Command handlers

dp.add\_handler(CommandHandler("start", start))

dp.add\_handler(CommandHandler("help", help\_command))

dp.add\_handler(CommandHandler("insights", insights))

dp.add\_handler(MessageHandler(Filters.text & ~Filters.command, handle\_message))

# Start the bot

updater.start\_polling()

updater.idle()

if \_\_name\_\_ == "\_\_main\_\_":

main()

### **Submission Steps:**

1. **Build and Test**: Set up the bot using the above code and test it on Telegram or Discord.
2. **Integrate Mode**: Ensure the bot uses Mode's API for secure blockchain operations.
3. **Deploy**: Deploy the bot on a cloud platform (e.g., AWS, Heroku).
4. **Document**: Prepare a detailed project description, including setup instructions, features, and use cases.
5. **Submit**: Follow the hackathon submission guidelines to submit your project.

### **Built With**

Here are the technologies and tools used to build the **"ChainMate - Blockchain Companion"**:

### **Frameworks and Libraries:**

1. **Eliza**:
   * Framework for creating AI agents with personalities that can connect to Telegram, Discord, etc.
   * Used to handle user conversations and provide a natural language interface.
2. **Python Telegram Bot**:
   * Library to interact with Telegram APIs for bot integration.
   * Handles bot commands and user interactions.
3. **Requests**:
   * Python library for making HTTP requests to blockchain APIs like Mode's API for real-time data.

### **APIs:**

1. **Mode Network API**:
   * Provides access to blockchain data for insights and Mode ecosystem integration.
2. **Blockchain APIs** (Optional Examples):
   * **Etherscan**: For Ethereum data.
   * **CoinGecko**: For token price and market data.

### **AI and NLP:**

1. **Eliza NLP Module**:
   * Built-in conversational AI to interpret queries and generate human-like responses.
   * Powered by modern NLP techniques.

### **Development Tools:**

1. **Python**:
   * Core programming language used to develop the AI agent.
2. **VS Code**:
   * Code editor for writing and managing the Python project.
3. **Postman**:
   * Used for testing API requests and responses during development.

### **Hosting and Deployment:**

1. **Heroku / AWS / Google Cloud**:
   * Cloud platforms for hosting and running the AI bot.
2. **Mode Integration**:
   * Direct integration with Mode's blockchain to support seamless transactions and operations.

### **Version Control:**

1. **Git and GitHub**:
   * For version control and collaborative development.
   * GitHub repository for project submission.

### **Why This Stack?**

1. **Python's Versatility**: Simplifies integration with APIs and provides strong libraries for AI and data handling.
2. **Eliza's Simplicity**: Eases the creation of conversational AI agents without complex setups.
3. **Mode API**: Ensures blockchain-specific capabilities are available to the agent.
4. **Cross-Platform Compatibility**: Works seamlessly on popular messaging platforms (Telegram, Discord).

**ChainMate** is an innovative AI-powered agent designed to revolutionize how blockchain enthusiasts and community members interact with blockchain data. This agent serves as a **real-time blockchain data analyst** and **community companion**, providing insightful data, answering questions, and engaging with users on blockchain platforms such as Telegram, Discord, and other social channels. The agent offers **real-time blockchain insights**, making it a go-to tool for community engagement, cryptocurrency enthusiasts, and decentralized finance (DeFi) users.

The project focuses on integrating **Mode's blockchain technology** and **Eliza’s conversational AI** capabilities to create a chatbot that serves multiple purposes:

* It provides up-to-date blockchain transaction data and token analytics.
* It answers frequently asked questions related to blockchain, DeFi, and cryptocurrency markets.
* It enables personalized notifications for important events like wallet transactions, price changes, governance proposals, and more.

### **Key Features**

1. **Real-Time Blockchain Data Insights**:
   * **ChainMate** continuously fetches real-time data about blockchain transactions, token swaps, market trends, and more. For example, it can report the current value of a specific token, recent transactions involving specific addresses, or provide analytics on gas fees, token movements, etc.
   * The agent fetches data from **Mode Network’s API**, as well as third-party APIs like **Etherscan** and **CoinGecko**, ensuring that it stays up-to-date with the latest blockchain trends.
2. **Conversational AI with Eliza**:
   * The core of **ChainMate’s interaction** lies in **Eliza**, an AI framework for natural language processing (NLP). Users can ask questions like, "What’s the latest transaction for Token X?" or "How’s the Ethereum price today?" and receive well-informed responses based on real-time data.
   * The bot has a **friendly, approachable personality** that mimics a human-like conversation, making it more engaging for users and enhancing their experience on blockchain-focused platforms.
3. **Community Engagement and Support**:
   * The agent is designed to foster community engagement by answering questions on blockchain topics, providing educational information, and participating in conversations on various platforms like **Telegram** and **Discord**.
   * Community members can ask questions about new crypto projects, general blockchain topics, or the latest DeFi trends, and **ChainMate** will respond with helpful information.
4. **Event-Driven Alerts and Notifications**:
   * Users can set **personalized notifications** about events such as new governance proposals, price fluctuations, wallet activity, or even token swaps.
   * For example, users could receive an alert when a specific token price crosses a threshold, or when a wallet makes a significant transaction.
5. **Cross-Platform Integration**:
   * The bot is built to work across multiple platforms, allowing users to interact with it wherever they are. ChainMate can be deployed on **Telegram**, **Discord**, or **any chat platform** that supports integrations.
   * Users can interact with the bot in their community’s messaging group, receive updates, and even set custom alerts.
6. **Mode Network Integration**:
   * ChainMate is built to work seamlessly with **Mode’s blockchain ecosystem**. It enables agents to operate securely in the DeFi space and manage blockchain-based transactions or positions.
   * The bot leverages Mode’s smart contract and wallet features to allow users to interact with blockchain assets, facilitate token swaps, participate in governance, and perform other DeFi actions.

### **Tech Stack**

1. **Python**:
   * **Python** is the primary language for building this AI agent. Python’s rich ecosystem of libraries and frameworks makes it the perfect choice for combining AI, web scraping, API interaction, and chatbot development.
2. **Eliza Framework**:
   * **Eliza** is a conversational AI framework, providing the **natural language processing (NLP)** capabilities required to interpret user messages and generate appropriate responses. It simplifies the process of creating a chatbot with a personality, making it easy for developers to design interactive bots that communicate in a human-like way.
3. **Telegram Bot API**:
   * The **Telegram Bot API** is used to create and manage the bot on the Telegram platform. This allows the bot to interact with users, respond to commands, and send real-time messages with blockchain insights and notifications.
4. **Mode API**:
   * The **Mode Network API** provides access to blockchain data and operations. The bot uses Mode’s API to fetch real-time transaction data, token swaps, and other blockchain-related insights to be shared with the users.
   * The Mode API also allows integration with **DeFi protocols** on Mode, enabling the bot to perform decentralized finance operations.
5. **Blockchain Data Providers (Optional)**:
   * **Etherscan API** and **CoinGecko API** can be used for additional blockchain data, such as token prices, gas fees, and transaction details that may not be provided by the Mode Network API.
6. **Heroku / AWS / Google Cloud**:
   * The bot is deployed on **cloud platforms** like **Heroku**, **AWS**, or **Google Cloud** for scalability and availability. These platforms provide the infrastructure to host and run the bot, ensuring it can handle multiple users and run continuously.
7. **GitHub**:
   * The **GitHub repository** is used for version control and collaborative development. It also serves as the submission point for the project, allowing other developers to contribute or fork the project.

### **Detailed Workflow**

1. **User Interaction**:
   * Users interact with the bot by sending text commands (e.g., /insights, /help, /set\_alert) or free-form questions about blockchain topics.
2. **API Data Fetching**:
   * When a user requests blockchain insights, the bot fetches real-time data from **Mode’s API** and additional APIs (Etherscan, CoinGecko, etc.) for token statistics, market trends, transaction details, etc.
3. **Response Generation**:
   * The bot then uses **Eliza’s NLP engine** to interpret user inputs (if they ask questions or engage in conversation) and generate a response.
   * If a user requests specific data, such as the latest token swap or price, the bot pulls the relevant blockchain data and responds accordingly.
4. **Alert System**:
   * Users can set alerts for specific blockchain events (e.g., price thresholds, transaction types, or wallet activity).
   * When the event occurs, the bot sends an instant notification to the user through Telegram or Discord.
5. **Blockchain Operations**:
   * ChainMate can integrate Mode’s **DeFi protocols**, allowing users to initiate actions like **swapping tokens, lending, or voting** on governance proposals directly through the bot.

### **Challenges Faced**

1. **Real-Time Data Fetching**:
   * Continuously fetching real-time blockchain data can be challenging due to rate limits, especially when interacting with public APIs. Implementing proper caching and handling API rate limits efficiently were key to optimizing performance.
2. **Natural Language Understanding**:
   * While Eliza provides basic conversational AI, developing an effective conversation flow that understands blockchain-specific jargon and can provide accurate responses required fine-tuning.
3. **Cross-Platform Deployment**:
   * Ensuring that the bot works seamlessly across multiple messaging platforms like **Telegram** and **Discord** presented deployment challenges, especially with authentication and maintaining consistent user experience.

### **Future Improvements**

1. **Machine Learning**:
   * Integrating machine learning models to predict market trends based on blockchain data, which can help users make better decisions.
2. **Expanded Blockchain Integration**:
   * Adding more blockchain networks (e.g., Solana, Polkadot) for broader data coverage.
3. **Voice Integration**:
   * Adding **voice interactions** to ChainMate, allowing users to communicate with the agent using voice commands.

### **Conclusion**

**ChainMate** is a versatile, interactive AI agent that bridges the gap between blockchain data and community engagement. By leveraging **Eliza’s conversational AI**, **Mode’s blockchain infrastructure**, and real-time data APIs, the bot creates an engaging and informative experience for users interested in blockchain, cryptocurrency, and decentralized finance. Whether used for real-time insights, community interactions, or personalized alerts, **ChainMate** has the potential to transform the way blockchain communities interact with their data and make decisions.