**AI prompt Arbitrage Crypto Web Application:**

**Prompt:**

"Create a fully functional **Arbitrage Crypto Web Application** that identifies and exploits real-time arbitrage opportunities across multiple cryptocurrency exchanges, automatically executes profitable trades, tracks historical performance, and allows users to customize and optimize their arbitrage strategy. The application should support both manual and automated trade execution, and integrate Artificial Intelligence (AI) to predict and optimize opportunities for maximizing profit.

**Key Requirements:**

1. **Arbitrage Opportunity Detection:**
   * Continuously scan multiple cryptocurrency exchanges for arbitrage opportunities.
   * Use **ccxt** (or API connections) to fetch market data from exchanges such as **Binance**, **Bybit**, **Kraken**, etc.
   * Identify price discrepancies between exchanges and calculate the arbitrage profit margin.
   * If the user has not connected exchange APIs, the app should automatically fetch data using **ccxt**.
2. **Database Integration:**
   * Store all detected arbitrage opportunities and related data (exchange names, coin pairs, margin, fees, timestamps) in a **MySQL** or **PostgreSQL** database.
   * Create a schema to track **arbitrage opportunities**, **trade logs**, and **user settings**.
   * Allow tracking of each opportunity over time and store historical data for future analysis.
3. **Opportunity Refresh & User Customization:**
   * The system should refresh the opportunity scan every second or according to a **user-defined time frame** (e.g., every minute or hour).
   * Allow users to select:
     + Which exchanges to monitor.
     + Which cryptocurrencies or coin pairs to track.
     + A minimum profit margin threshold for trades to execute.
     + A custom refresh interval for checking new opportunities.
4. **Live Tracking Interface:**
   * Create a **user-friendly web interface** that displays live arbitrage opportunities in real time, showing details like exchange names, price differences, profit margin, and the time the opportunity first appeared.
   * Display a **fee breakdown** for each opportunity (including trading fees, withdrawal fees, etc.).
   * Show an estimated **net profit** after executing the trade, considering all expenses.
5. **Trade Execution:**
   * When a profitable arbitrage opportunity is detected, the system should:
     + Automatically **execute the trade** (buy low on one exchange, sell high on another).
     + Calculate **transaction fees** (trading fees, withdrawal fees) for both exchanges.
     + Show the total **profit margin** after fees and expenses, and allow users to set a minimum profit threshold before the system executes the trade.
   * Provide **detailed reports** of each executed trade, showing the profit, fee breakdown, and execution time.
6. **AI Integration for Opportunity Optimization:**
   * Integrate an **AI model** (e.g., using **TensorFlow** or **PyTorch**) to enhance arbitrage prediction and optimization.
   * The AI should:
     + Learn from historical arbitrage opportunities and predict profitable coin pairs and exchanges.
     + Optimize the detection of profitable trades by analyzing patterns in price movements, volume, and fees.
     + Suggest the best arbitrage strategy based on real-time market conditions and user preferences.
   * Continuously improve over time by incorporating new data and feedback.
7. **Expense & Profit Calculation:**
   * Automatically calculate all **expenses** involved in executing an arbitrage trade, including:
     + Exchange trading fees
     + Withdrawal fees
     + Transaction slippage
     + Other related costs
   * Provide users with a **clear profit estimate** after all expenses are deducted, showing the net profit for each trade.
8. **User Interface (UI) and Dashboard:**
   * Build an intuitive dashboard that allows users to:
     + View live arbitrage opportunities.
     + Track their trade performance, including total profit, number of successful trades, and failed trades.
     + See detailed breakdowns of fees, net profit, and other relevant data for each trade.
     + View AI-powered insights and suggestions for optimal arbitrage strategies.
   * Display a historical **performance chart** with metrics like total profit, win/loss ratio, etc.
9. **Security and User Authentication:**
   * Implement **secure authentication** methods for users (OAuth, API keys, etc.).
   * Ensure **secure storage of API keys** and sensitive data using encryption or environment variables.
   * Implement **role-based access control** (if needed) for different users (e.g., admin vs. regular user).
10. **Backend Technology and Tools:**
    * Use **Python** for backend development (with **Flask** or **Django** as the web framework).
    * Utilize **ccxt** for exchange integration and market data fetching.
    * Integrate with databases like **MySQL** or **PostgreSQL** for storing opportunities and user data.
    * Use **SQLAlchemy** for ORM (Object-Relational Mapping) if using Python.
11. **Frontend Technology and Tools:**
    * Build the frontend with **HTML**, **CSS**, and **Bootstrap** for a responsive UI.
    * Use **JavaScript** (preferably with **React.js** or **Vue.js**) for real-time updates and live tracking functionality.
    * Implement dynamic graphs or charts to display arbitrage opportunity trends, profits, and historical performance.
12. **Scalability and Optimization:**
    * Design the system to handle **large volumes of real-time data** and multiple exchanges and coins.
    * Ensure the application is optimized for high-frequency scanning and trading without performance issues.
    * Implement **rate limiting** and other safeguards to prevent overwhelming the exchange APIs.
13. **Deployment:**
    * Deploy the web application to a cloud server (AWS, Heroku, or similar).
    * Set up CI/CD pipelines for deployment and testing.
    * Ensure that the production environment is stable, scalable, and secure.

**Final Deliverables:**

* A fully functional **Arbitrage Crypto Web Application** with automated arbitrage detection and execution.
* An integrated **AI system** for optimizing arbitrage opportunities.
* A user-friendly interface for monitoring and managing arbitrage opportunities.
* Full backend and frontend code with all required functionalities.
* Documentation on how to use and deploy the application.

The application should be **optimized for performance, security, and scalability**, and it should provide a seamless experience for users looking to profit from arbitrage opportunities across cryptocurrency exchanges."