# AutoSnipe Capabilities and Guide to Creating a Similar Setup

## I. Capabilities of AutoSnipe

### Copy Trading

- Automatically mirrors trades from selected whale wallets.

- Configurable allocation (match full trade or percentage-based copy).

- Customizable take-profit and stop-loss settings.

- Ability to pause or stop copying at any time.

- Transparency via trade logs and real-time execution.

### Token Sniping (Trade Automation)

- AI-powered trade execution for newly launched Solana tokens.

- Configurable parameters such as liquidity filters, holder distribution, and contract safety checks.

- Auto-buy and auto-sell based on pre-set profit/loss rules.

- Priority transaction routing to ensure fast execution.

- Multiple sniper bots can run simultaneously with different risk profiles.

### Whale Wallet Analytics (Trader Lens)

- Tracks and analyzes successful trader wallets.

- Displays trade history, portfolio value, and profit trends.

- Helps users decide whom to copy-trade for better success rates.

### Meme Token Discovery (Meme Vision)

- Highlights trending and newly launched meme tokens.

- Tracks migrations from Pump.fun to Raydium.

- Provides insights into volume spikes and emerging trends.

### Trading Interface & Manual Trading

- Web-based platform with TradingView chart integration.

- Market and limit order options.

- Live price tracking for Solana tokens.

- Custom alerts and notifications.

### Risk Management & Security Measures

- AI-based scam detection and filtering of known rug-pull tokens.

- Take-profit and stop-loss settings to automate risk management.

- MEV protection to avoid front-running and sandwich attacks.

- Fast withdrawal mechanisms with minimal delays.

- Non-custodial approach with wallet integration (Phantom, Solflare, etc.).

### API & Advanced Customization

- Open API for advanced users to integrate trading strategies.

- Node.js SDK and REST API for automation.

- Webhooks for notifications on trade execution.

### Pricing Model

- No monthly subscription; instead, a 1% trade fee.

- Referral system offering fee discounts.

- Deposits accepted from multiple blockchains.

## II. Guide to Creating a Similar Setup

### Technology Stack

- Frontend: React.js or Vue.js for a modern, responsive UI.

- Backend: Node.js with Express.js for API handling.

- Database: PostgreSQL or MongoDB for storing user data and trade history.

- Blockchain Integration: Solana Web3.js and Raydium API for trade execution.

- Hosting: AWS, DigitalOcean, or Firebase for scalability.

### Blockchain & Smart Contract Integration

- Connect to Solana blockchain using Solana Web3.js.

- Utilize Raydium API for trading and liquidity pool data.

- Implement smart contract scanners to analyze new token contracts.

### Trading Execution Engine

- Develop a copy-trading engine that monitors and replicates trades.

- Implement a high-speed sniping engine that detects new token launches.

- Optimize transaction priority using Jito-Solana for faster trade execution.

### Risk Management & Security

- Implement automated scam detection based on contract metadata.

- Add MEV protection techniques like private transaction relays.

- Offer manual override options for copy traders.

- Ensure non-custodial wallet integrations with Web3 authentication.

### User Interface & Features

- Build an intuitive dashboard with TradingView charts.

- Create a wallet analytics page showing top traders' performance.

- Develop an alerts system for price movements and trade execution.

### Monetization & Pricing Model

- Charge a per-trade commission (e.g., 1%).

- Implement a referral system for organic growth.

- Consider offering premium features for advanced traders.

### Deployment & Maintenance

- Deploy frontend on Vercel or Netlify for speed.

- Use Kubernetes or Docker for backend scalability.

- Implement real-time monitoring with Prometheus & Grafana.

- Set up Telegram/Discord support channels for user engagement.

## Conclusion

- AutoSnipe is a robust Solana-based copy trading and sniping platform.

- To build a similar system, one needs a combination of blockchain expertise, real-time trading algorithms, and strong security measures.

- The key challenges include optimizing transaction speeds, ensuring smart contract safety, and maintaining a seamless user experience.

- With the right development stack and strategic execution, a competitive trading automation platform can be created in the Solana ecosystem.