Coding a Funded Trading Platform: Full Checklist

1. Core Architecture

- Choose your stack: Frontend (React.js/Vue.js), Backend (Node.js, Django, etc.), DB (PostgreSQL, MongoDB)

- Use a clean folder structure (MVC or service-based)

- Manage environment configs via .env files

2. Authentication & User Management

- User registration/login system

- Email verification and password reset

- Role-based access control (Admin, Trader)

- JWT or session-based authentication

- 2FA (optional for security)

3. Trader Dashboard

- Display stats (balance, drawdown, profit %, etc.)

- Account status (active, breached, qualified)

- Challenge countdown timer

- Charts and metrics for trades

4. Admin Dashboard

- Manage traders and accounts

- Create and assign demo/funded accounts

- View challenge performance

- Manual account resets or overrides

5. Challenge Logic Automation

- Implement challenge rules (drawdown, profit target, etc.)

- Real-time trade monitoring

- Breach/pass status updates

- Notifications for traders and admins

6. Broker / Platform Integration

- Integrate with MT4/MT5, cTrader, or DXTrade APIs

- Create demo and funded accounts automatically

- Fetch trade data and sync to DB

7. Payment Integration

- Set up gateways: Stripe, Razorpay, PayPal, Crypto

- Secure webhook processing

- Purchase challenge + auto account assignment

8. Email & Notifications

- Send onboarding, breach, payout emails

- Use SendGrid, Mailgun, or SMTP for delivery

- In-app or dashboard alerts

9. Account Scaling & Payouts

- Show payout history and status

- Payout request form for traders

- Admin approval and processing

- Integration with Wise, Payoneer, crypto wallets

10. Logging & Monitoring

- Track trade logs and rule breaches

- Use error logging tools (e.g. Sentry)

- Monitor uptime and performance

11. Security Best Practices

- Input validation and sanitization

- Use HTTPS (SSL certificate)

- Encrypt sensitive data (e.g. JWT, bcrypt)

- Implement rate limiting and CORS policies

12. Database Schema Essentials

- Users, Trading Accounts, Challenges, Trades, Payments

- Payouts, Notifications, Logs

13. Frontend Pages

- Landing page, Pricing, FAQ, Contact

- Login/Signup

- Trader Dashboard

- Admin Panel

- Help Center, Payout Page

14. Testing & DevOps

- Unit and integration testing

- Use Git + GitHub/GitLab

- Dev/Staging/Production environments

- CI/CD pipelines (GitHub Actions, etc.)

- Use Docker if needed

15. VS Code Tools & Extensions

- Prettier, ESLint, REST Client, GitLens

- Docker and SQL Tools extensions

- Integrated terminal and Postman for API testing

16. Optional Advanced Features

- Referral system, Discord bot integration

- WebSocket for live updates

- Dark mode and responsive design

- Partner API (external devs)