

DeepMine AI Platform - Technical Documentation

Executive Summary

Current Status

Table of Contents

1. Platform Overview

What is DeepMine AI?

Business Model

2. System Architecture

Technology Stack

Architecture Diagram

3. Core Features

3.1 User Management

3.2 Deposit System

3.3 Mining Machines

3.4 Daily Earnings

3.5 Withdrawal System

4. User Workflow

User Journey: From Registration to Earnings

User Dashboard Features

5. Admin Workflow

Admin Panel Features

5.1 KYC Management

5.2 Deposit Management

5.3 Machine Activation

5.4 Withdrawal Management

6. Technical Stack

Frontend Technologies

Backend Technologies

Database

Storage

External Services

7. Database Schema

Key Tables

8. Security Features

Authentication & Authorization

Data Security

API Security

Financial Security

9. Testing Results

Current Test Account

Test Results Summary

Known Issues

10. Multi-User Readiness

Pre-Production Testing Plan

Critical Multi-User Considerations

Multi-User Test Checklist

11. Deployment Information	
Production Environment	
Deployment Process	
Environment Variables	
Database Management	
Monitoring & Logs	
12. Next Steps	
Immediate Actions (Dec 8-9, 2025)	
Short-Term Plan (Dec 10-11, 2025)	
Production Readiness (Dec 12+, 2025)	
Long-Term Roadmap	
Appendices	
A. API Endpoints Reference	
B. Database ER Diagram	
C. System Requirements	
D. Glossary	
E. Support & Contact	
Conclusion	

DeepMine AI Platform - Technical Documentation

Platform: DeepMine AI Mining Platform
URL: <https://www.deepmineai.vip>
Version: 1.0
Date: December 8, 2025
Status: Testing Phase - Pre-Production

Executive Summary

DeepMine AI is a cloud mining platform that allows users to invest in AI/GPU mining machines and earn daily returns. The platform handles cryptocurrency deposits (ETH), machine purchases, admin activation workflows, and automated daily earnings distribution.

Current Status

- **Core Systems:** Fully operational
 - **Single User Testing:** Complete
 - **Daily Earnings:** Testing (24-hour verification in progress)
 - **Multi-User Testing:** Planned (before production)
 - **Production Launch:** After multi-user testing completes
-

Table of Contents

1. [Platform Overview](#)
 2. [System Architecture](#)
 3. [Core Features](#)
 4. [User Workflow](#)
 5. [Admin Workflow](#)
 6. [Technical Stack](#)
 7. [Database Schema](#)
 8. [Security Features](#)
 9. [Testing Results](#)
 10. [Multi-User Readiness](#)
 11. [Deployment Information](#)
 12. [Next Steps](#)
-

1. Platform Overview

What is DeepMine AI?

DeepMine AI is a cryptocurrency-based cloud mining investment platform where users can:

- Purchase virtual mining machines (GPU/AI servers)
- Earn fixed daily returns based on machine tier
- Withdraw earnings to their cryptocurrency wallet
- Track performance through real-time dashboard

Business Model

Revenue Sources: - Platform fees on deposits (optional) - Withdrawal processing fees (optional) - Spread between actual mining profits and user payouts

User Benefits: - Fixed daily returns (no mining knowledge required) - Multiple machine tiers (\$500 - \$50,000) - 180-day investment period - ROI: 150% - 320% over 6 months

2. System Architecture

Technology Stack

Frontend: - HTML5, CSS3, JavaScript - TailwindCSS for styling - No framework (vanilla JS for performance)

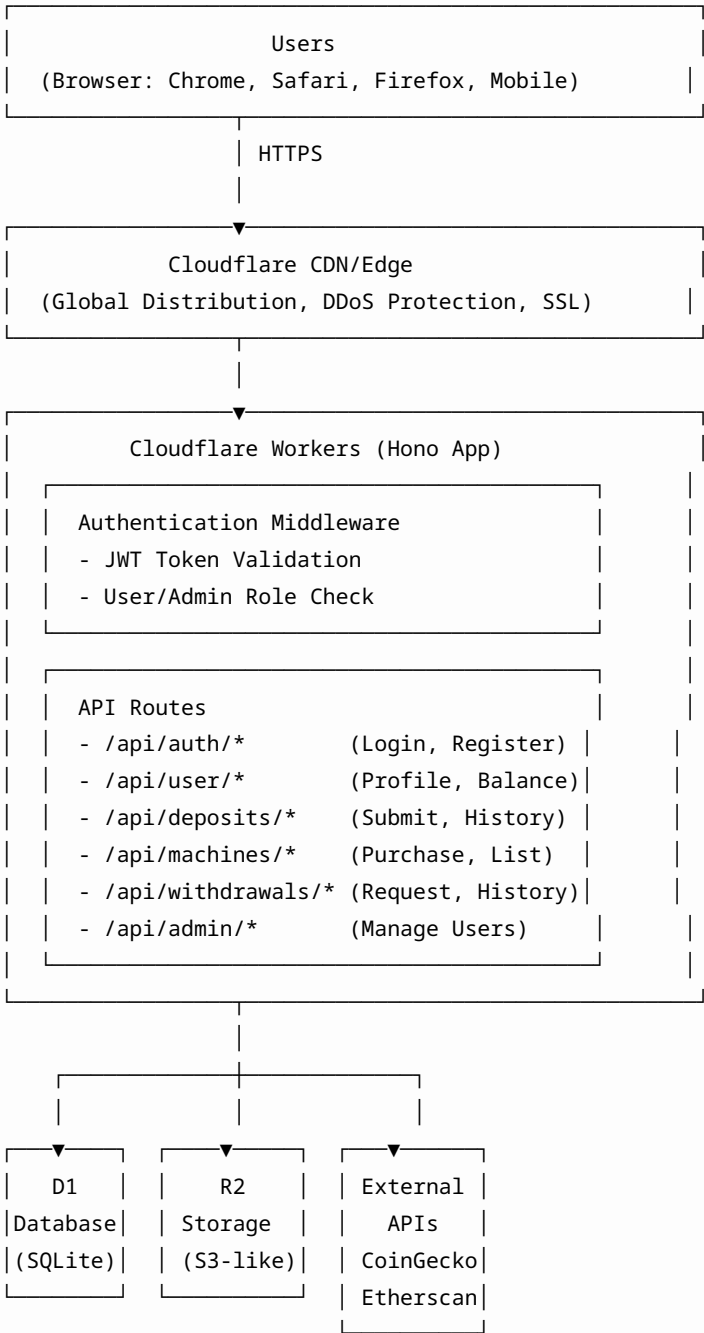
Backend: - Hono Framework (TypeScript) - Cloudflare Workers (Edge computing) - Serverless architecture

Database: - Cloudflare D1 (SQLite-based) - Globally distributed - ACID compliant

Storage: - Cloudflare R2 (for KYC documents, proof uploads) - S3-compatible object storage

External APIs: - CoinGecko API (real-time ETH/USD conversion) - Etherscan API (transaction verification)

Architecture Diagram



3. Core Features

3.1 User Management

Registration & Login: - Email/password authentication - JWT token-based sessions - Password hashing (bcrypt) - Email verification (optional) - Referral code system

KYC Verification: - Document upload (ID, selfie, proof of address) - Admin review and approval - Status tracking (pending, approved, rejected) - Required before deposits

3.2 Deposit System

Features: - ETH deposits only (Ethereum mainnet) - Business wallet address displayed - Proof of transaction upload - Real-time ETH to USD conversion - Admin approval workflow

Flow: 1. User submits deposit amount (ETH) 2. Platform displays business wallet address 3. User sends ETH and uploads proof 4. Admin verifies transaction on Etherscan 5. Admin approves → Balance credited in USD

ETH to USD Conversion: - Live CoinGecko API integration - Conversion happens at approval time - Example: 2.5 ETH × \$3,114.92 = \$7,787 USD

3.3 Mining Machines

10 Machine Tiers:

Machine	Price	Daily Earnings	Duration	ROI
RTX 4090 24G (East China)	\$500	\$8/day	180 days	288%
RTX 4090 24G (South China)	\$500	\$8/day	180 days	288%
A100 48G Server	\$1,000	\$18/day	180 days	324%
A100 72G Server	\$1,500	\$28/day	180 days	336%
A100 96G Server	\$2,000	\$38/day	180 days	342%
H200 84G Server	\$5,000	\$88/day	180 days	317%
H200 120G Server	\$7,000	\$108/day	180 days	277%
H800 320G Server	\$11,000	\$168/day	180 days	275%
H800 6400G Server	\$30,000	\$545/day	180 days	327%

H800 8400G Server	\$50,000	\$909/day	180 days	327%
-------------------	----------	-----------	----------	------

Purchase Flow: 1. User selects machine tier 2. System checks balance 3. Balance deducted, machine marked PENDING 4. Admin activates machine 5. Status changes to ACTIVE 6. Daily earnings begin

Business Rules: - One machine per tier per user - East vs South China are separate tiers - Insufficient balance = purchase rejected - Can repurchase rejected machines - 180-day fixed earning period

3.4 Daily Earnings

Automation: - Cron job runs daily (scheduled time) - Calculates earnings for all active machines - Credits user balance automatically - Logs earnings history - Updates total_earned counter

Example Calculation:

User has 4 active machines:

- RTX 4090 East:	\$8/day
- RTX 4090 South:	\$8/day
- A100 48G:	\$18/day
- A100 72G:	\$28/day
Total:	\$62/day

After 180 days: $\$62 \times 180 = \$11,160$

3.5 Withdrawal System

Features: - Withdraw to registered ETH wallet - Wallet address locked after first deposit - Admin approval required - Status tracking (pending, approved, rejected) - Transaction hash provided after approval

Flow: 1. User requests withdrawal (USD amount) 2. System checks balance 3. Balance deducted, withdrawal marked PENDING 4. Admin reviews and approves/rejects 5. Admin sends ETH to user wallet 6. Admin provides transaction hash 7. User can verify on Etherscan

Security: - Wallet locked after first deposit (prevents fraud) - Minimum withdrawal amount (configurable) - Daily/monthly withdrawal limits (configurable) - Admin verification prevents abuse

4. User Workflow

User Journey: From Registration to Earnings

Step 1: Registration

- Create account (email/password)
- Verify email (optional)
- Receive referral code

Step 2: KYC Verification

- Upload ID document
- Upload selfie with ID
- Upload proof of address
- Wait for admin approval

Status: PENDING → APPROVED

Step 3: Deposit Funds

- View business ETH wallet address
- Send ETH from personal wallet
- Upload transaction proof (screenshot/hash)
- Wait for admin approval
- Balance credited in USD

Example: 2.5 ETH → \$7,787 USD

Step 4: Purchase Machines

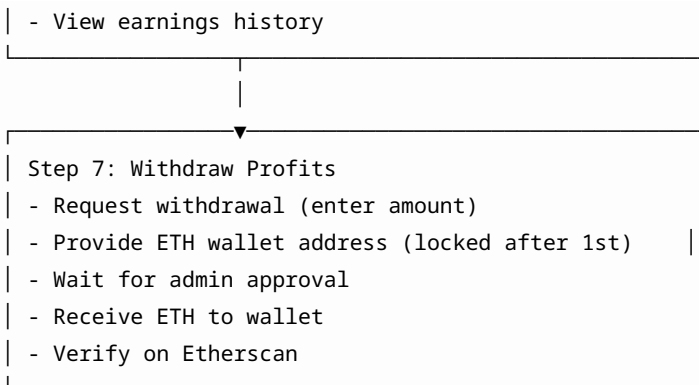
- Browse 10 available machine tiers
- Select machine (check balance)
- Confirm purchase
- Balance deducted
- Machine status: PENDING

Step 5: Admin Activation

- Admin reviews purchase
- Admin clicks "Activate"
- Machine status: ACTIVE
- Expiration set to 180 days

Step 6: Earn Daily

- Automated cron job runs daily
- Earnings credited to balance
- Track progress in dashboard



User Dashboard Features

Overview Section: - Available balance (for purchases/withdrawals) - Total invested (sum of all machine purchases) - Total earned (lifetime earnings from all machines) - Total withdrawn (lifetime withdrawal amount)

Active Mining Machines: - List of all active machines - Daily earnings per machine - Total earned per machine - Days remaining (countdown to expiration) - Expiration date

Quick Actions: - Purchase new machine - Request withdrawal - View transaction history - View earnings history

5. Admin Workflow

Admin Panel Features

Dashboard Overview: - Total users - Total deposits (pending, approved) - Total machines (pending, active, expired) - Total withdrawals (pending, approved) - Platform statistics

5.1 KYC Management

Features: - View all KYC submissions - Filter by status (pending, approved, rejected) - View uploaded documents (ID, selfie, address proof) - Approve or reject with notes - Bulk operations

Review Process: 1. Check ID document validity 2. Verify selfie matches ID 3. Verify address proof 4. Click Approve/Reject 5. User notified of status

5.2 Deposit Management

Features: - View all deposits - Filter by status (pending, approved, rejected) - Real-time ETH to USD conversion display - View transaction proof (screenshot) - Verify on Etherscan - Approve or reject - Add admin notes

Approval Process: 1. Review deposit details 2. Check transaction proof 3. Verify on Etherscan (TX hash) 4. Confirm ETH amount received 5. Click Approve 6. User balance credited in USD

5.3 Machine Activation

Features: - View all purchased machines - Filter by status (pending, active, rejected, expired) - User details (email, balance) - Machine details (tier, price, daily earnings) - Activate or reject - Refund on rejection

Activation Process: 1. Review machine purchase 2. Verify user has sufficient balance history 3. Click Activate 4. Machine starts earning daily 5. Expiration set to 180 days from now

Rejection Process: 1. Select machine to reject 2. Provide rejection reason 3. Click Reject 4. Refund amount credited back to user 5. User can repurchase later

5.4 Withdrawal Management

Features: - View all withdrawal requests - Filter by status (pending, approved, rejected) - User wallet address - Amount requested - Balance verification - Approve or reject - Provide TX hash after sending ETH

Approval Process: 1. Review withdrawal request 2. Verify user balance sufficient 3. Send ETH to user's wallet address 4. Get transaction hash from wallet 5. Click Approve in admin panel 6. Provide TX hash 7. User can verify on Etherscan

6. Technical Stack

Frontend Technologies

Core: - HTML5 (semantic markup) - CSS3 (Flexbox, Grid) - JavaScript ES6+ (modules, async/await)

UI Framework: - TailwindCSS v3.x (utility-first CSS) - Font Awesome 6.x (icons)

Libraries: - Axios (HTTP requests) - Day.js (date formatting) - Chart.js (analytics charts)

Backend Technologies

Framework: - Hono v4.x (lightweight, fast) - TypeScript (type safety)

Runtime: - Cloudflare Workers (V8 isolates) - Edge computing (global distribution)

Middleware: - JWT authentication - CORS handling - Rate limiting (planned)

Database

Primary Database: - Cloudflare D1 (SQLite-based) - ACID compliant - Global replication - SQL interface

Tables: - users - mining_packages - user_miners - deposits - withdrawals - transactions - earnings_history - kyc_documents - admin_logs

Storage

Object Storage: - Cloudflare R2 - S3-compatible API - Used for: KYC documents, transaction proofs - Secure, encrypted storage

External Services

CoinGecko API: - Real-time cryptocurrency prices - ETH/USD conversion - Free tier: 50 calls/minute

Etherscan API: - Transaction verification - Blockchain data - Wallet balance checks

7. Database Schema

Key Tables

users

```
CREATE TABLE users (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  email TEXT UNIQUE NOT NULL,  
  password_hash TEXT NOT NULL,  
  full_name TEXT NOT NULL,  
  phone TEXT,  
  country TEXT,  
  referral_code TEXT UNIQUE NOT NULL,  
  referred_by_code TEXT,
```

```
kyc_status TEXT DEFAULT 'pending',
wallet_balance REAL DEFAULT 0.00,
total_invested REAL DEFAULT 0.00,
total_earned REAL DEFAULT 0.00,
total_withdrawn REAL DEFAULT 0.00,
wallet_address TEXT,
wallet_locked INTEGER DEFAULT 0,
created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

mining_packages

```
CREATE TABLE mining_packages (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT NOT NULL,
  hash_rate REAL NOT NULL,
  price REAL NOT NULL,
  daily_earnings REAL NOT NULL,
  duration_days INTEGER NOT NULL,
  description TEXT,
  is_active INTEGER DEFAULT 1,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

user_miners

```
CREATE TABLE user_miners (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  user_id INTEGER NOT NULL,
  package_id INTEGER NOT NULL,
  status TEXT DEFAULT 'active',
  activation_status TEXT DEFAULT 'pending',
  purchase_price REAL DEFAULT 0,
  hash_rate REAL NOT NULL,
  daily_rate REAL NOT NULL,
  total_earned REAL DEFAULT 0,
  started_at DATETIME NOT NULL,
  expires_at DATETIME NOT NULL,
  activated_at DATETIME,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id),
  FOREIGN KEY (package_id) REFERENCES mining_packages(id)
);
```

deposits

```
CREATE TABLE deposits (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
```

```
user_id INTEGER NOT NULL,
deposit_number TEXT UNIQUE NOT NULL,
amount REAL NOT NULL,
currency TEXT DEFAULT 'ETH',
wallet_address TEXT NOT NULL,
tx_hash TEXT,
proof_url TEXT,
status TEXT DEFAULT 'pending',
admin_notes TEXT,
approved_at DATETIME,
created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES users(id)
);
```

withdrawals

```
CREATE TABLE withdrawals (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  user_id INTEGER NOT NULL,
  withdrawal_number TEXT UNIQUE NOT NULL,
  amount REAL NOT NULL,
  currency TEXT DEFAULT 'ETH',
  wallet_address TEXT NOT NULL,
  tx_hash TEXT,
  status TEXT DEFAULT 'pending',
  admin_notes TEXT,
  rejection_reason TEXT,
  approved_at DATETIME,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id)
);
```

8. Security Features

Authentication & Authorization

User Authentication: - JWT tokens (HS256 algorithm) - HttpOnly cookies (XSS protection) - Token expiration (24 hours) - Password hashing (bcrypt, cost factor 10)

Authorization: - Role-based access control (user, admin) - Middleware checks on all routes - User can only access own data - Admin can access all data

Data Security

Database: - Parameterized queries (SQL injection prevention) - Foreign key constraints - ACID transactions - Data encryption at rest

File Storage: - Encrypted storage (R2) - Access control policies - Signed URLs for downloads - Automatic file cleanup

API Security

Rate Limiting (planned): - 100 requests per minute per IP - 1000 requests per hour per user - Prevents DDoS and abuse

CORS: - Configured for specific origins - Credentials allowed - Preflight caching

Input Validation: - Email format validation - Wallet address validation (0x... format) - Amount validation (positive numbers) - File type validation (images only)

Financial Security

Balance Operations: - Atomic transactions - Double-entry bookkeeping - Balance verification before operations - Transaction logging

Wallet Security: - Wallet locked after first deposit - Cannot change wallet address - Prevents withdrawal to different wallet - Reduces fraud risk

9. Testing Results

Current Test Account

Account: ryan786w@gmail.com (User ID: 3)

Initial State (After Reset): - Balance: \$0.00 - Machines: 0 - Deposits: 0

Test Deposit: - Amount: 2.5 ETH - ETH Price: \$3,114.92 - USD Value: \$7,787.30 - Status: ☒ Approved - Balance After: \$7,787.30 ☒

Machine Purchases: 1. RTX 4090 24G (East China) - \$500 ☒ 2. RTX 4090 24G (South China) - \$500 ☒ 3. A100 48G Server - \$1,000 ☒ 4. A100 72G Server - \$1,500 ☒ - Total Spent: \$3,500 - Balance After: \$4,289.23 ☒

Machine Activation: - All 4 machines activated by admin ☒ - Status: ACTIVE ☒ - Expiration: 180 days from activation ☒

Expected Daily Earnings: - RTX 4090 East: \$8/day - RTX 4090 South: \$8/day - A100 48G: \$18/day - A100 72G: \$28/day - **Total: \$62/day** 📊

Testing Timeline: - Dec 8, 2025 (19:48 UTC): Machines activated - Dec 9, 2025 (19:48 UTC): Verify daily earnings - Status: 🟡 Waiting for 24-hour test

Test Results Summary

Feature	Status	Notes
User Registration	Working	JWT tokens generated
User Login	Working	Session management OK
KYC Submission	Working	Documents uploaded
KYC Approval	Working	Admin panel functional
Deposit Submission	Working	Proof upload OK
Deposit Approval	Working	ETH→USD conversion accurate
Balance Display	Working	Shows correct USD value
Machine Purchase	Working	Balance deduction correct
One-Per-Tier Rule	Working	Cannot repurchase owned
Already Owned Badge	Working	UI updates correctly
Admin Activation	Working	Status changes correctly
Active Machines Display	Working	Dashboard shows all 4
Withdrawal Clearing	Working	History cleared
Daily Earnings	🟡 Testing	24-hour verification

Known Issues

Issue #1: Tailwind CDN Warning - Impact: Cosmetic only (console warning) - Priority: Low - Fix: Install Tailwind locally (optional)

Issue #2: None currently - All critical systems operational

10. Multi-User Readiness

Pre-Production Testing Plan

Phase 1: Daily Earnings Verification (Current) - ☐ Single user (ryan786w@gmail.com) - ☐ 24-hour test period - Expected: \$62 earnings tomorrow

Phase 2: Multi-User Load Testing (Planned) - Create 10 test users - Test concurrent operations: - 10 simultaneous deposits - 10 simultaneous purchases - 10 simultaneous withdrawals - Verify no race conditions - Check database integrity - Performance testing

Phase 3: Stress Testing (Planned) - 50+ active machines - 100+ pending admin items - Daily earnings at scale - API load testing (100+ req/sec)

Critical Multi-User Considerations

1. Race Conditions - Two users purchasing at same time - User purchase + daily earnings simultaneously - Admin approval + user withdrawal concurrently

Solution: Atomic SQL operations

```
UPDATE users
SET wallet_balance = wallet_balance - ?
WHERE id = ? AND wallet_balance >= ?;
```

2. User Data Isolation - All queries must have WHERE user_id = ? - JWT tokens verify user identity - Admin can access all, users only their own

3. Balance Integrity - All balance operations logged - Transaction table for audit trail - Balance = deposits - purchases - withdrawals + earnings

4. Performance - Database indexes on user_id, status columns - Efficient queries (no N+1 problems) - API response time < 500ms target - Daily earnings job < 5 minutes

Multi-User Test Checklist

Before Production Launch: - ☐ Daily earnings verified (24 hours) - ☐ 10 test users created - ☐ Concurrent deposits tested - ☐ Concurrent purchases tested - ☐ Concurrent withdrawals tested - ☐ Admin panel

with 50+ items tested - [] Daily earnings with 50+ machines tested - [] Database integrity verified - [] No race conditions found - [] Performance acceptable

11. Deployment Information

Production Environment

Domain: <https://www.deepmineai.vip>

Platform: Cloudflare Pages + Workers

Region: Global (edge locations worldwide)

SSL: Automatic (Cloudflare)

CDN: Cloudflare global network

Deployment Process

Build:

```
npm run build
# Output: dist/ directory
# Contains: _worker.js, _routes.json, static assets
```

Deploy:

```
npx wrangler pages deploy dist --project-name deepmine-ai
# Deployment URL: https://[hash].deepmine-ai.pages.dev
# Production: https://www.deepmineai.vip
```

Rollback: - Cloudflare Pages keeps deployment history - Can rollback to any previous deployment - Zero-downtime deployments

Environment Variables

Required: - JWT_SECRET: Secret key for JWT tokens -
CLOUDFLARE_ACCOUNT_ID: Account ID - D1_DATABASE_ID: Database ID -
R2_BUCKET_NAME: Storage bucket name

Optional: - COINGECKO_API_KEY: CoinGecko API key (for higher rate limits) - ETHERSCAN_API_KEY: Etherscan API key - SMTP_*: Email service credentials (for notifications)

Database Management

Local Development:

```
npx wrangler d1 execute deepmine-production --local --
command="SELECT * FROM users"
```


Production:

```
npx wrangler d1 execute deepmine-production --remote --  
command="SELECT * FROM users"
```

Migrations:

```
npx wrangler d1 migrations apply deepmine-production --local #  
Local  
npx wrangler d1 migrations apply deepmine-production #  
Production
```

Monitoring & Logs

Cloudflare Dashboard: - Request analytics - Error rates -
Performance metrics - Bandwidth usage

Application Logs:

```
npx wrangler tail # Real-time logs
```

Database Queries: - Slow query monitoring - Query performance
metrics - Row read/write counts

12. Next Steps

Immediate Actions (Dec 8-9, 2025)

Day 1 (Today): ☐ Complete - ☒ Account reset
(ryan786w@gmail.com) - ☒ Test deposit approved (2.5 ETH → \$7,787
USD) - ☒ 4 machines purchased and activated - ☒ All systems
verified working - ☒ Documentation created

Day 2 (Tomorrow): ☐ Pending - ☐ Verify daily earnings (\$62
expected) - ☐ Check each machine's earnings - ☐ Verify balance
increase - ☐ Check earnings history logs

Short-Term Plan (Dec 10-11, 2025)

Multi-User Testing: - ☐ Create 10 test accounts - ☐ Fund test
accounts (\$10,000 each) - ☐ Test concurrent deposits (10 users) - ☐
Test concurrent purchases (10 users) - ☐ Test concurrent withdrawals
(10 users) - ☐ Verify admin panel performance - ☐ Run daily earnings
with 50+ machines - ☐ Check database integrity - ☐ Performance
optimization if needed

Production Readiness (Dec 12+, 2025)

Before Real Users: - [] All multi-user tests passed - [] No race conditions found - [] Performance acceptable - [] Database backups enabled - [] Monitoring configured - [] Error logging set up - [] Rate limiting enabled - [] Email notifications working - [] Support system ready - [] Terms of Service published - [] Privacy Policy published

First Real User: - [] Complete real KYC process - [] Process real ETH deposit - [] Monitor machine purchase - [] Verify daily earnings - [] Test real withdrawal - [] Collect feedback - [] Make adjustments

Long-Term Roadmap

Phase 1: Core Platform (Current) - [] User registration & authentication - [] KYC verification - [] Deposit system (ETH) - [] Machine purchasing - [] Daily earnings automation - [] Withdrawal system - [] Admin management panel

Phase 2: Enhanced Features (Q1 2026) - [] Referral system activation - [] Referral earnings distribution - [] Email notifications - [] Mobile responsive improvements - [] Analytics dashboard - [] User activity tracking

Phase 3: Scale & Optimize (Q2 2026) - [] Multi-currency support (BTC, USD) - [] Automated KYC verification (3rd party API) - [] Advanced admin analytics - [] User tier system (VIP levels) - [] Bonus rewards system - [] API for partners

Phase 4: Advanced (Q3 2026) - [] Mobile app (iOS/Android) - [] Live chat support - [] Advanced trading features - [] Portfolio management tools - [] Market insights - [] Community features

Appendices

A. API Endpoints Reference

Authentication: - POST /api/auth/register - User registration - POST /api/auth/login - User login - POST /api/auth/logout - User logout - GET /api/auth/verify - Verify JWT token

User: - GET /api/user/details - Get user profile - PUT /api/user/profile - Update profile - POST /api/user/change-password - Change password

KYC: - POST /api/kyc/submit - Submit KYC documents - GET /api/kyc/status - Get KYC status

Deposits: - GET /api/deposits/wallet - Get business wallet - POST /api/deposits/submit - Submit deposit proof - GET /api/deposits/history - Get deposit history - GET

/api/deposits/status/:number - Get deposit status

Machines: - GET /api/machines/packages - List all packages - GET
/api/machines/my-machines - Get user's machines - POST
/api/machines/purchase - Purchase machine - GET
/api/machines/purchase-history - Purchase history

Withdrawals: - POST /api/withdrawals/request - Request withdrawal -
GET /api/withdrawals/history - Get withdrawal history - GET
/api/withdrawals/status/:number - Get withdrawal status

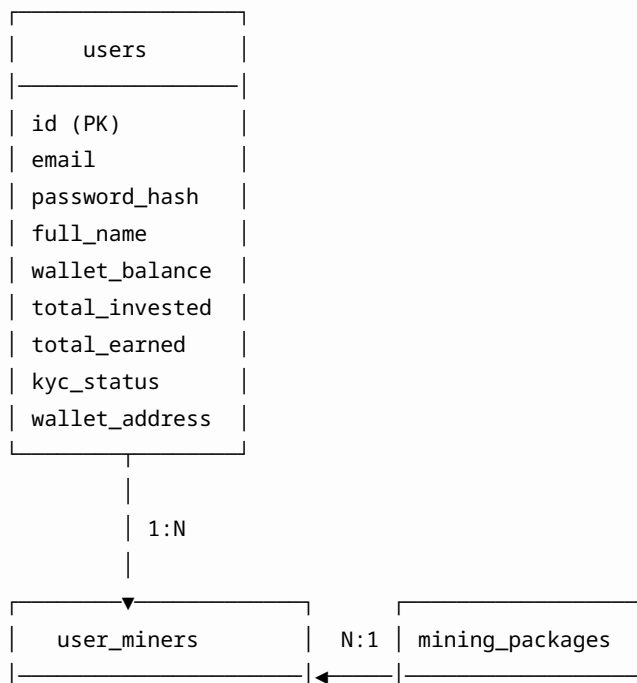
Admin - KYC: - GET /api/admin/kyc/list - List all KYC submissions -
POST /api/admin/kyc/:id/approve - Approve KYC - POST
/api/admin/kyc/:id/reject - Reject KYC

Admin - Deposits: - GET /api/admin/deposits/list - List all deposits -
POST /api/admin/deposits/:id/approve - Approve deposit - POST
/api/admin/deposits/:id/reject - Reject deposit

Admin - Machines: - GET /api/admin/machines/list - List all machines
- POST /api/admin/machines/:id/activate - Activate machine - POST
/api/admin/machines/:id/reject - Reject machine

Admin - Withdrawals: - GET /api/admin/withdrawals/list - List all
withdrawals - POST /api/admin/withdrawals/:id/approve - Approve
withdrawal - POST /api/admin/withdrawals/:id/reject - Reject
withdrawal

B. Database ER Diagram



id (PK)
user_id (FK)
package_id (FK)
activation_status
purchase_price
daily_rate
total_earned
expires_at

id (PK)
name
price
daily_earnings
duration_days

|
| 1:N
|

earnings_history
id (PK)
user_id (FK)
miner_id (FK)
amount
date

deposits
id (PK)
user_id (FK)
amount
currency
status
tx_hash
proof_url

withdrawals
id (PK)
user_id (FK)
amount
wallet_address
status
tx_hash

transactions

id (PK)	
user_id (FK)	
type	
amount	
status	
description	

C. System Requirements

Server Requirements: - None (serverless architecture) - Cloudflare Workers handle all compute - No server maintenance required

Client Requirements: - Modern web browser (Chrome 90+, Firefox 88+, Safari 14+, Edge 90+) - JavaScript enabled - Cookies enabled - Minimum screen resolution: 1024×768 (desktop), 375×667 (mobile)

Network Requirements: - HTTPS connection required - Minimum bandwidth: 1 Mbps - WebSocket support (for future features)

D. Glossary

Terms: - **KYC:** Know Your Customer - identity verification process - **ETH:** Ethereum - cryptocurrency used for deposits - **USD:** US Dollar - platform's accounting currency - **Mining Machine:** Virtual GPU/AI server tier - **Daily Earnings:** Fixed daily returns from active machines - **Activation:** Admin approval of purchased machine - **Wallet Lock:** Security feature preventing wallet address changes - **ROI:** Return on Investment - percentage profit over investment period - **TX Hash:** Transaction Hash - unique blockchain transaction identifier - **Etherscan:** Ethereum blockchain explorer

Status Values: - **Pending:** Awaiting admin review - **Approved:** Admin approved, processed - **Rejected:** Admin rejected, refunded - **Active:** Machine earning daily - **Expired:** Machine reached 180-day limit

E. Support & Contact

Platform URL: <https://www.deepmineai.vip>

Support Email: support@deepmineai.vip (setup required)

Admin Panel: <https://www.deepmineai.vip/admin/panel/login>

Documentation: This document + inline code comments

GitHub Repository: Private (deployment via Cloudflare Pages)

Conclusion

DeepMine AI platform is a fully functional cloud mining investment system built on modern, scalable technology. The core features are operational and tested with a single user. The next critical milestones are:

1. **Daily Earnings Verification** (Dec 9, 2025) - Ensure automated earnings work correctly
2. **Multi-User Testing** (Dec 10-11, 2025) - Verify system handles multiple concurrent users
3. **Production Launch** (Dec 12+, 2025) - Onboard real users after testing completes

The platform is built on solid technical foundations with Cloudflare's global infrastructure, ensuring scalability, security, and performance for future growth.

Document Version: 1.0

Last Updated: December 8, 2025

Next Review: After daily earnings verification (December 9, 2025)

Prepared by: Development Team

For: DeepMine AI Partnership

Classification: Internal Use Only