Question 1: Database Schema Design

Task: Design MongoDB schemas for:

- Users
- Investments
- Referrals / Level income
- ROI history

Requirements:

- Each user can have multiple investments.
- Each investment has `amount`, `plan`, `startDate`, `endDate`, `status`.
- Users can refer others; referrals create level income.
- Daily ROI must be tracked per user and per investment.

Expected output: MongoDB schemas (Mongoose models).

==========

Question 2: API Design

Task: Build APIs for:

1. User registration/login
2. Create investment
3. Fetch user dashboard (investments, ROI, level income)
4. Fetch referral tree

Requirements:

- Secure endpoints
- Efficient queries to calculate total income

==========

Question 3: Business Logic Implementation

Task: Implement a function that calculates:

- Daily ROI for all active investments
- Level income based on referral hierarchy
- Update the user balances accordingly

==========

Question 4: React Dashboard

Task: Build a simple React dashboard that shows:

- Total investments
- Daily ROI
- Level income
- Referral tree (nested)

Requirements:

- Fetch data from your API
- Display charts / tables
- Handle loading states

==========

Question 5: Cron Job / Scheduler

Task: Implement a scheduled job that:

- Runs daily at midnight
- Calculates ROI for all users
- Updates balances

Requirements:

- Use `node-cron` or similar
- Ensure idempotency (won't double-calculate if rerun)