**Database Structure and Backend API Endpoints**

# 1. Databases

## - Users

Columns:

- id (Primary key)  
 - first\_name  
 - last\_name  
 - email  
 - password  
 - role (ENUM: buyer, seller, admin)  
 - nick\_name

- isActive

## - Logs

Columns:   
 - id (Primary key)  
 - sender\_id (Users.id)  
 - recipient\_id (Users.id)  
 - message (Text)  
 - message\_type  
 - deed\_id (Deeds.id / Null)  
 - dispute\_id (Deeds.id / Null)

## - Dispute Table

Columns:   
 - id (Primary key)  
 - deed\_id (Escrow.id)  
 - raised\_by (Users.id)  
 - reason (Text)  
 - resolution (Text)  
 - status (ENUM: open, resolved, closed)

- resolved\_at (Time)

- closed\_at (Time)

## - Deed

Columns:   
 - title  
 - description  
 - payment\_method (ENUM: Ethereum, Solana, Ton)  
 - payment\_type (ENUM: one\_time, milestone)  
 - amount  
 - timeline  
 - status (ENUM: pending, in\_progress, requested, completed, cancelled, disputed)  
 - buyer\_id (Users.id)  
 - seller\_id (Users.id)  
 - category

## - DeedMilestone

Columns:   
 - deed\_id (Deeds.id)  
 - name  
 - amount  
 - timeline  
 - status (ENUM: pending, in\_progress, requested, completed, cancelled)

## - WorkSubmission

Columns:   
 - deed\_id (Deeds.id)  
 - submitted\_by (Users.id)  
 - file\_link  
 - description  
 - status (ENUM: pending, approved, revision\_requested, fraud\_reported)

# 2. Backend APIs

## - User

- POST /user/register – Register new users.  
 Body Parameters: first\_name, last\_name, email, password

- POST /user/login – User login and token generation.  
 Body Parameters: email, password

Response: id (with this user id, please call all APIs), and other user model variables (first\_name, last\_name, nick\_name, email, password)

- GET /user/:id – Get user profile.

- PATCH /user/:id – Update user detail.  
 Body Parameters: first\_name, last\_name, password

- GET /user/activity/:userId – Get the User activity (count of activeDeeds, completeDeeds, totalDeeds and totalMoney)

Response: totalDeeds, totalMoney, activeDeeds, completedDeeds

## - Deed

- GET /deed – Get all deeds

- POST /deed/create – Create a new deed.  
 Body Parameters: userId, title, description, buySellType(“BUY” or “SELL”), payment\_method, payment\_type, amount, timeline, milestones ([{name, amount, timeline}, { name, amount, timeline}, ...])

- GET /deed/:id – Retrieve a specific deed

- PATCH /deed/:id/update – Update deed details (change status).

- POST /deed/requestFunds – Buyer can request full funds or a milestone

Body Parameters: user\_id, deed\_id, milestone\_id

- POST /deed/releaseFunds - Seller can release full funds or a milestone

Body Parameters: user\_id, deed\_id, milestone\_id

- POST /deed/milestones/update/:milestone\_id - Update the milestone information

Body Parameters: milestone\_name, amount, timeline, status(‘pending’, ‘in\_progress’, ‘requested’, ‘completed’, ‘cancelled’)

- **Work Submission**

- POST /work/submit – Seller submits a file with its link and description.  
 Body Parameters: deed\_id, file\_link, description, userId

- PATCH /work/review/:deed\_id – Buyer can approve, request revision, or report fraud.  
 Body Parameters: status (‘pending’, ‘approved’, ‘revision\_requested’, ‘fraud\_reported’)

## - Dispute

- POST /dispute/create – Create a dispute.  
 Body Parameters: deed\_id, user\_id, reason

- GET /dispute/:id – Get dispute status.

- PATCH /dispute/:id/update – Update dispute status.  
 Body Parameters: status (‘open’, ‘resolved’, ‘closed’)

## - Log

- POST /log/create – Create a message for the recipient.  
 Body Parameters: sender\_id, recipient\_id, message, message\_type, deed\_id, dispute\_id

- GET /log/:id – Get all messages related to user Id