

1. User

- `_id` (PK)
- `name`
- `email` (Unique)
- `password` (Hashed)
- `role` (worker, task-creator, admin)
- `photoURL`
- `coin`
- `createdAt`

2. Task

- `_id` (PK)
- `title`
- `description`
- `imageURL`
- `quantity`
- `payableAmount`
- `completionDate`
- `creatorEmail` (FK → User)
- `creatorName`
- `status` (active, expired)
- `createdAt`

3. Submission

- `_id` (PK)
- `taskId` (FK → Task)
- `taskTitle`
- `workerEmail` (FK → User)

- workerName
- submissionDetails
- payableAmount
- status (pending, approved, rejected)
- creatorEmail (FK → User)
- createdAt

4. Payment

- _id (PK)
- taskCreatorEmail (FK → User)
- amountPaid
- coinAdded
- transactionId (Stripe)
- createdAt

5. Withdrawal

- _id (PK)
- workerEmail (FK → User)
- workerName
- coin
- amount
- paymentSystem (bkash, rocket, nagad)
- accountNumber
- status (pending, completed)
- createdAt

6. Notification

- _id (PK)
- toEmail (FK → User)

- message
 - isRead
 - createdAt
-

Relationships

1. **User** → **Task** (1-to-Many):

- A **User** (role=task-creator) creates many **Tasks**.
- Foreign Key: creatorEmail in Task.

2. **Task** → **Submission** (1-to-Many):

- A **Task** receives many **Submissions** from Workers.
- Foreign Key: taskId in Submission.

3. **User** → **Submission** (Many-to-Many):

- A **Worker** submits many **Submissions**.
- A **Task-Creator** reviews many **Submissions**.
- Foreign Keys: workerEmail and creatorEmail in Submission.

4. **User** → **Payment** (1-to-Many):

- A **Task-Creator** makes many **Payments** to buy coins.
- Foreign Key: taskCreatorEmail in Payment.

5. **User** → **Withdrawal** (1-to-Many):

- A **Worker** requests many **Withdrawals**.
- Foreign Key: workerEmail in Withdrawal