Masroofy Team Contributions & Code Explanation

Project: Masroofy - Family Wallet Management System
Date: December 20, 2025

---

1. Moaz Abdelaleem (ID: 13007327)

Feature: User Authentication & Hierarchy

Contribution Overview

Moaz implemented the core security layer dealing with User Authentication (Register/Login) and the Parent-Child hierarchy foundation.

Code Logic Explanation

- **Registration Logic (`/auth/register`):**
  - Validates input fields (username, email, password).
  - Checks if user already exists in MongoDB.
  - Hashes passwords using `bcrypt` before saving for security.
  - Generates a JSON Web Token (JWT) upon successful signup for session management.
- **Login Logic (`/auth/login`):**
  - Finds user by username.
  - Compares input password with stored hash using `bcrypt.compare` .
  - Returns a JWT token and user profile if credentials match.

Code Snippet:

```
// Password Hashing (in User model)
userSchema.pre('save', async function(next) {
    if (!this.isModified('password')) return next();
    this.password = await bcrypt.hash(this.password, 10);
});

// JWT Generation
const token = jwt.sign(
    { userId: user._id, role: user.role },
    process.env.JWT_SECRET,
    { expiresIn: '7d' }
);
```

---

2. Eyad Ahmed Saeed (ID: 13005578)

Feature: Visual Budget Tracker

Contribution Overview

Eyad built the analytics engine that tracks spending against the total monthly allowance (budget).

Code Logic Explanation

- **Budget Calculation (`/analytics/budget`):**
  - **Total Budget:** Sums up all *incoming transfers* received by the user in the current month using MongoDB Aggregation pipeline (`$match` , `$group` , `$sum`).
  - **Spent Amount:** Calculates `Total Budget - Current Balance` .
  - **Status Indicator:** Returns a visual status color:
    - 🟢 Green: Spending < 50%
    - 🟠 Orange: Spending > 50%
    - 🔴 Red: Spending > 80%

Code Snippet:

```
// Aggregating total monthly allowance
const incomingTransfers = await Transaction.aggregate([
    { $match: {
        receiverId: user._id,
        type: 'transfer',
        timestamp: { $gte: firstDayOfMonth, $lte: lastDayOfMonth }
    }},
    { $group: { _id: null, totalBudget: { $sum: '$amount' } }}
]);
```

---

3. Omar Khaled (ID: 13003972)

Feature: Wallet Top-Up & Transfer System

Contribution Overview

Omar implemented the banking simulation features allowing parents to add funds and transfer money to children.

Code Logic Explanation

- **Deposit (`/wallet/deposit`):**
  - Restricted to 'parent' role only.
  - Updates parent's `balance` in User collection.
  - Creates a `type: 'deposit'` record in Transaction collection.
- **Transfer to Child (`/wallet/transfer`):**
  - **Atomic Transaction:** Deducts from Parent balance and Adds to Child balance.
  - **Security Checks:** Verifies the target child actually belongs to the logged-in parent using `parentId` check.
  - Records a transaction linking `senderId` (parent) and `receiverId` (child).

Code Snippet:

```
// Transfer Logic
parent.balance -= amount;
child.balance += amount;
```

```
await parent.save();
await child.save(); // Both balances updated

// Record the flow
const transaction = new Transaction({
    type: 'transfer',
    senderId: parent._id,
    receiverId: child._id,
    amount: amount
});
```

---

4. Omar Samer (ID: 13001857)

Feature: Transaction History & Expense Tracking

Contribution Overview

Omar developed the transaction ledger system, allowing users to record expenses and parents to audit their children's history.

Code Logic Explanation

- **Expense Recording (`POST /transactions`):**
    - Checks if user has sufficient balance.
    - Deducts amount from user balance.
    - Creates a transaction of type 'expense'.
- **View Child History (`/transactions/child/:childId`):**
    - **Access Control:** Ensures requestor is a parent and owns the child account.
    - **Query:** Fetches all transactions where the child is either the sender OR the receiver.
    - **Formatting:** Adds a virtual `direction` field (incoming/outgoing) based on whether the child was sender or receiver for frontend display.

Code Snippet:

```
// Determining flow direction for display
const formattedTransactions = transactions.map(tx => {
    const isIncoming = tx.receiverId && tx.receiverId.equals(childId);
    return {
        ...tx,
        direction: isIncoming ? 'incoming' : 'outgoing'
    };
});
```

---

5. Omar Mahmoud (ID: 13006696)

Feature: Smart Daily Calculator

Contribution Overview

Omar created a forecasting tool that advises users on their daily spending limits to last through the month.

Code Logic Explanation

- **Feature:** "Before you buy, check the calculator"
- **Safe Daily Spend Algorithm:**
  1. Calculates `daysRemaining` in current month.
  2. Formula: `Safe Spend = Current Balance / Days Remaining`.
- **Spending Analysis:**
  - Projects end-of-month balance based on average daily spending so far.
  - Provides textual advice (e.g., "Warning: You may run out of funds").

Code Snippet:

```
const today = new Date();
const lastDayOfMonth = new Date(today.getFullYear(), today.getMonth() + 1, 0);
const daysRemaining = (lastDayOfMonth - today) / (1000 * 60 * 60 * 24);

const safeDailySpend = (user.balance / daysRemaining).toFixed(2);
```

---

6. Bahaa Aldin Ahmed (ID: 13002233)

Feature: Child Accounts Management

Contribution Overview

Bahaa handled the CRUD operations for managing the child entities under a parent.

Code Logic Explanation

- **Create Child (`/auth/create-child`):**
  - Creates a new User with role 'child'.
  - Links the new child to the parent via `parentId` field.
  - Updates the Parent's `children` array to include the new Child ID (Two-way reference for efficient querying).
- **Middleware Integration:**
  - His routes heavily rely on `authMiddleware` to ensure strictly only parents can access these management endpoints.

Code Snippet:

```
// Linking Child to Parent
const child = new User({
    username,
    role: 'child',
    parentId: req.user._id // Link to creator
});
await child.save();
```

```
// Update Parent's list
await User.findByIdAndUpdate(req.user._id, {
    $push: { children: child._id }
});
```