

Name: Jayaprakash

Experiment no: 8

Uid: 23BAI70240

Aim: To implement a banking transaction system with atomicity and rollback support using MongoDB sessions.

CODE:

Transaction Route:

```
const mongoose = require("mongoose");
```

```
const User = require("../models/User");
```

```
exports.transferMoney = async (req, res) => {  
  
  const session = await mongoose.startSession();  
  
  session.startTransaction();  
  
  
  try {  
  
    const { from, to, amount } = req.body;
```

```
const sender = await
```

```
User.findById(from).session(session);
```

```
const receiver = await
```

```
User.findById(to).session(session);
```

```
if (sender.balance < amount) {
```

```
    throw new Error("Insufficient balance");
```

```
}
```

```
sender.balance -= amount;
```

```
receiver.balance += amount;
```

```
await sender.save();
```

```
await receiver.save();
```

```
await session.commitTransaction();
```

```
res.send("Transfer successful");
```

```
} catch (error) {
```

```
    await session.abortTransaction();

    res.status(500).send("Transaction failed: " +
error.message);

  } finally {

    session.endSession();

  }

};
```