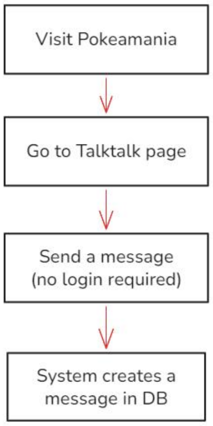
Debugging 1. Message System Enhancement: Sender Field Refactoring

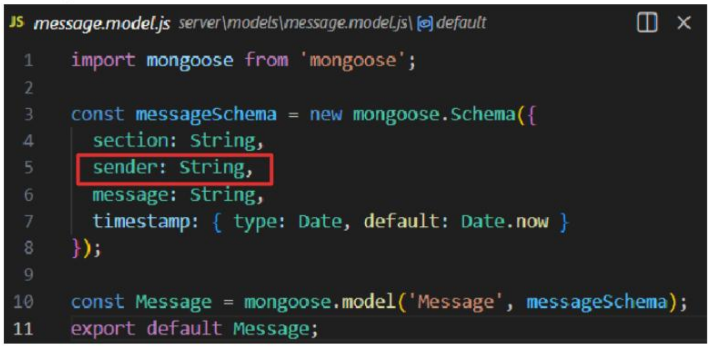
1) **Problem**: Messages stored sender as static string, causing inconsistency when user profiles changed.
2) **Solution**: Changed the guest user flow in the message system and refactored message schema.

<Before>

Guest & Member flow


Visit Pokeamania
Go to Talktalk page
Send a message (no login required)
System creates a message in DB

Message Schema



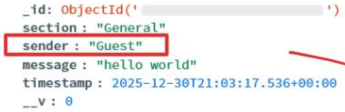
```
import mongoose from 'mongoose';

const messageSchema = new mongoose.Schema({
  section: String,
  sender: String,
  message: String,
  timestamp: { type: Date, default: Date.now }
});

const Message = mongoose.model('Message', messageSchema);
export default Message;
```

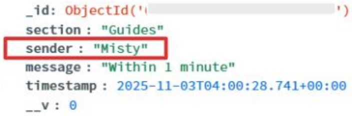
Message Objects

Guest



```
_id: ObjectId('[redacted]')
section : "General"
sender : "Guest"
message : "hello world"
timestamp : 2025-12-30T21:03:17.536+00:00
__v : 0
```

Member

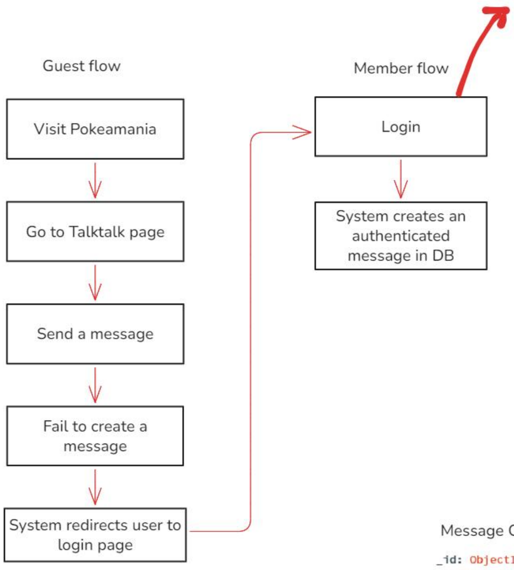


```
_id: ObjectId('[redacted]')
section : "Guides"
sender : "Misty"
message : "Within 1 minute"
timestamp : 2025-11-03T04:00:28.741+00:00
__v : 0
```

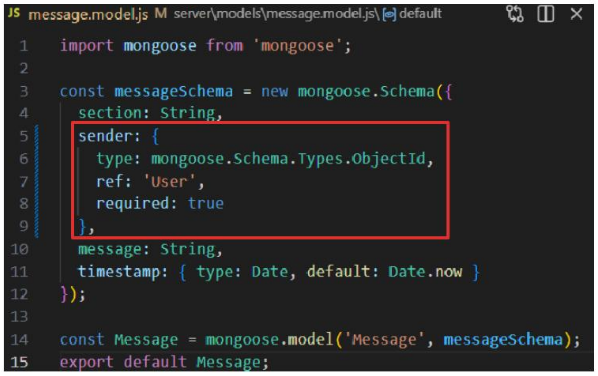
Stored sender as static string, causing inconsistency when user profiles changed.

<After>

Requires user authentication for messaging



Message Schema



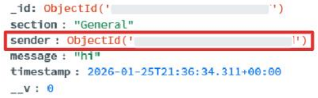
```
import mongoose from 'mongoose';

const messageSchema = new mongoose.Schema({
  section: String,
  sender: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'User',
    required: true
  },
  message: String,
  timestamp: { type: Date, default: Date.now }
});

const Message = mongoose.model('Message', messageSchema);
export default Message;
```

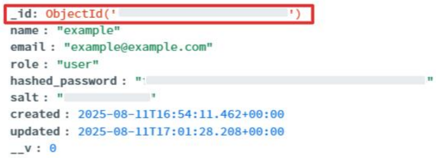
Messages remain linked to valid users

Message Object



```
_id: ObjectId('[redacted]')
section : "General"
sender : ObjectId('[redacted]')
message : "hi"
timestamp : 2026-01-25T21:36:34.311+00:00
__v : 0
```

User Object

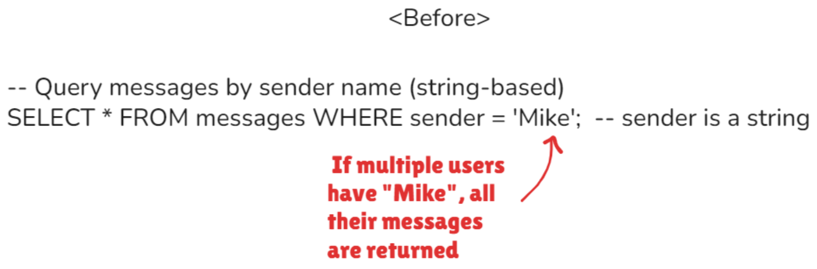


```
_id: ObjectId('[redacted]')
name : "example"
email : "example@example.com"
role : "user"
hashed_password : "[redacted]"
salt : "[redacted]"
created : 2025-08-11T16:54:11.462+00:00
updated : 2025-08-11T17:01:28.208+00:00
__v : 0
```

3) **Impact**:

   a. **Security:** Enforced user authentication for messaging, preventing anonymous or unauthorized message creation and ensuring all messages are traceable to verified users.

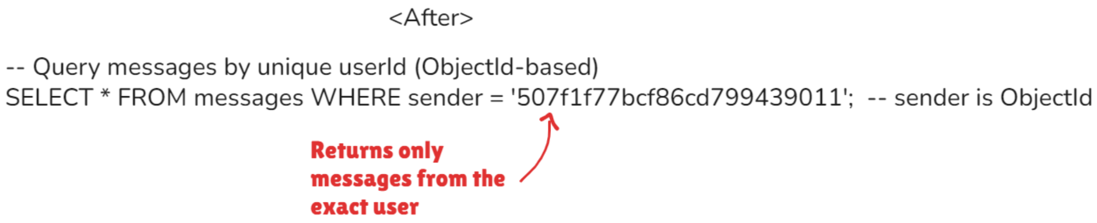
   b. **Data Accuracy:** Improved data accuracy by using UserId references instead of String names, enabled efficient JOINs and eliminated duplication risks.



<Before>

```
-- Query messages by sender name (string-based)
SELECT * FROM messages WHERE sender = 'Mike';  -- sender is a string
```

If multiple users have "Mike", all their messages are returned



<After>

```
-- Query messages by unique userId (ObjectId-based)
SELECT * FROM messages WHERE sender = '507f1f77bcf86cd799439011';  -- sender is ObjectId
```

Returns only messages from the exact user

Debugging 2. Frontend Message Display: Dynamic Profile Name Mapping

1. **Problem**: Frontend rendered sender names from cached login data (localStorage username), so profile name changes were not reflected in message threads, resulting in stale or inconsistent displays.
2. **Solution**: Resolved each message's sender by *userId* to fetch the current *profile.name* at render time, ensuring profile changes are reflected immediately without manual refreshes.

3. **Impact**:
    1. **Data consistency**: Profile name changes automatically reflect in message displays by resolving *userId* to *profile.name*, eliminating manual DB updates.

    2. **UX**: Messages always display current profile names, reducing user confusion.