

Student Name: Dwyanne

Date: November 28, 2025

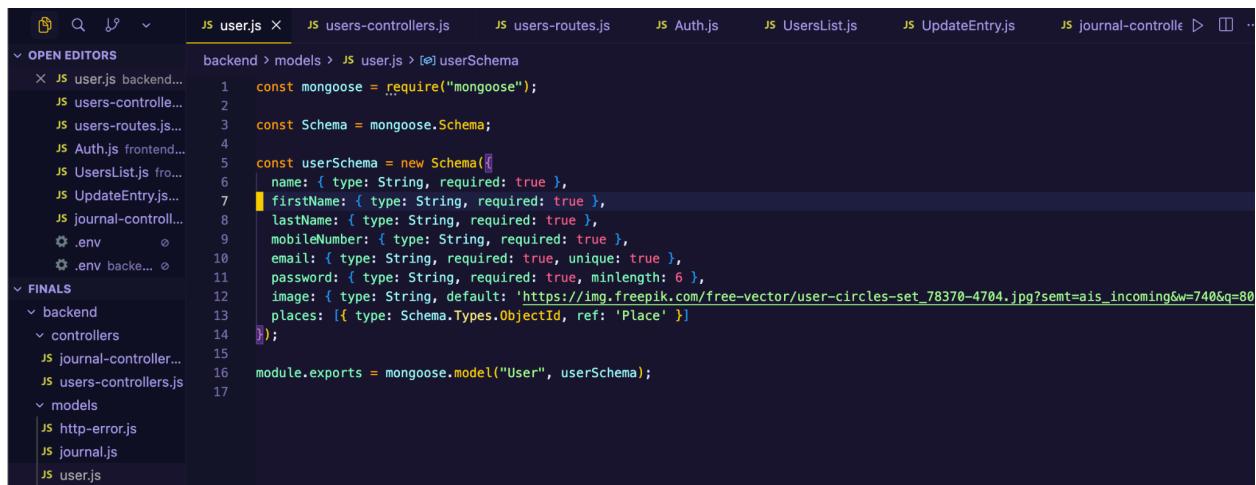
GitHub Repository: [Insert Public GitHub Link Here]

Part 1: New Task Implementation (Schema Update)

Objective: Added `firstName`, `lastName`, and `mobileNumber` to the User schema and updated the Signup flow to capture and store these fields.

1. Code Implementation

Screenshot of backend/models/user.js showing the new fields defined in the Mongoose schema.



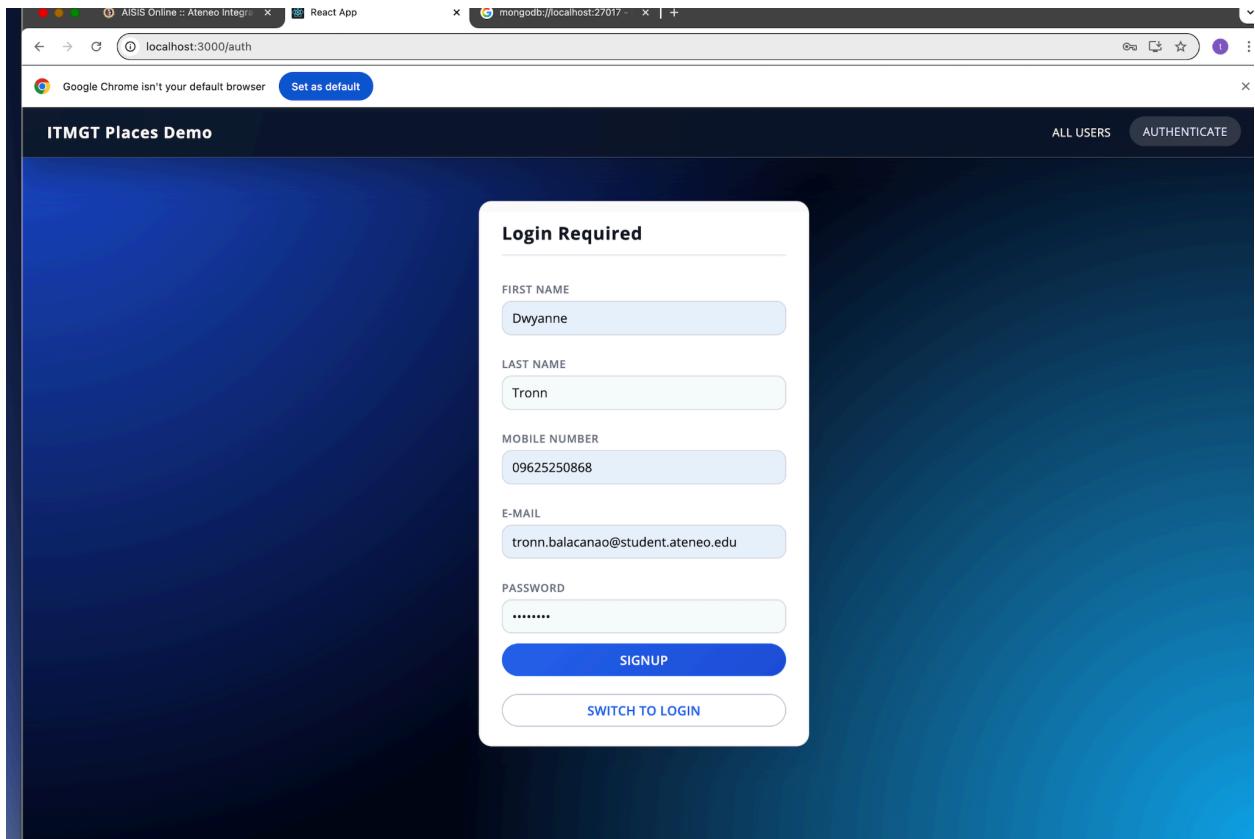
```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;

const userSchema = new Schema({
  name: { type: String, required: true },
  firstName: { type: String, required: true },
  lastName: { type: String, required: true },
  mobileNumber: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true, minlength: 6 },
  image: { type: String, default: "https://img.freepik.com/free-vector/user-circles-set_78370-4704.jpg?semt=ais_incoming&w=740&q=80" },
  places: [{ type: Schema.Types.ObjectId, ref: 'Place' }]
});

module.exports = mongoose.model("User", userSchema);
```

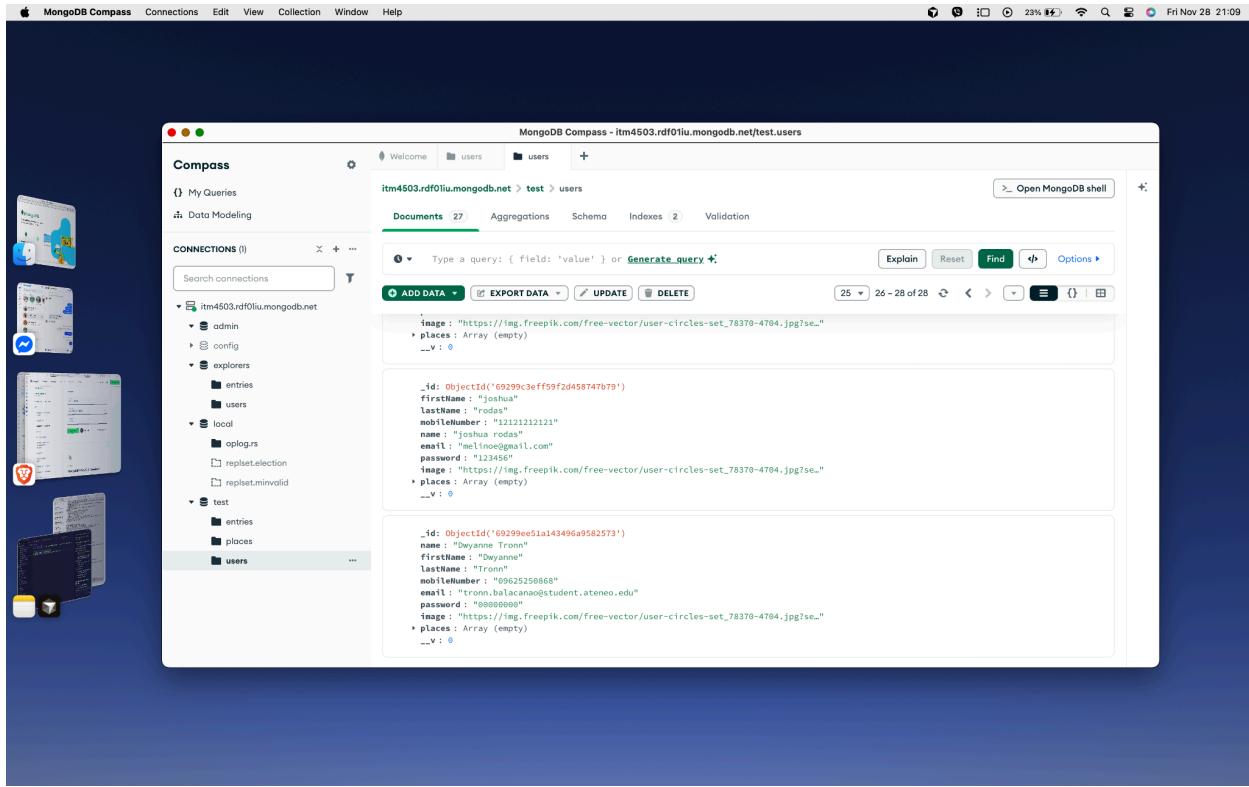
2. Frontend Interface

Screenshot of the Signup form showing the three new input fields.



3. Database Validation

Screenshot of MongoDB Compass/Atlas showing a newly created user document containing the new fields.



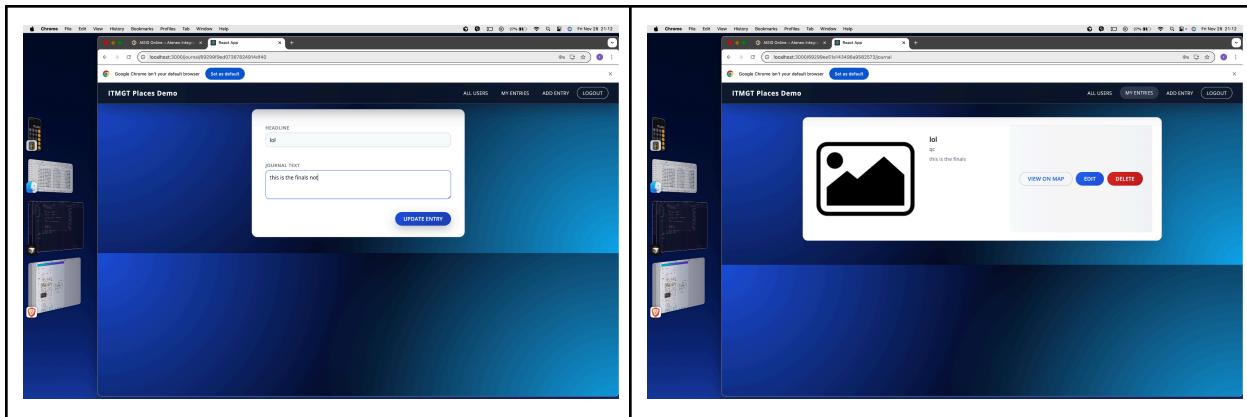
Part 2: Debugging Documentation

Bug A: "The Infinite Spinner" (Update Failure)

1. The Issue & Reproduction

Root Cause: The updateEntry controller modified the data but failed to persist it because the entry.save() method was missing/commented out. The frontend spinner hung indefinitely (or failed silently) because the response logic was flawed or the database operation never completed.

Fake update / update button not working



2. Investigation (Console Logs)

Screenshot of the VS Code Terminal showing the logs. It should show the request arriving, but NO "Save Successful" log following it.

3. The Fix

Screenshot of the code in journal-controllers.js showing the added await entry.save() line.

```

js Auth.js frontend... 160
js UsersList.js fro... 161
js UpdateEntry.js... 162
x js journal-controll... 163
  .env      o 164
  .env backe... o 165
FINALS
166
167
168
169
170
171
172
173
174
175
176
177
  .env
  .env backe... o
try {
  console.log("Updating entry fields...");
  entry.headline = headline;
  entry.journalText = journalText;

  await entry.save();
  console.log("Entry saved successfully");
} catch (err) {
  console.log("Error saving entry:", err.message);
  const error = new HttpError(
    "Something went wrong, could not update entry.",
    500
  );
  return next(error);
}

console.log("Sending success response");

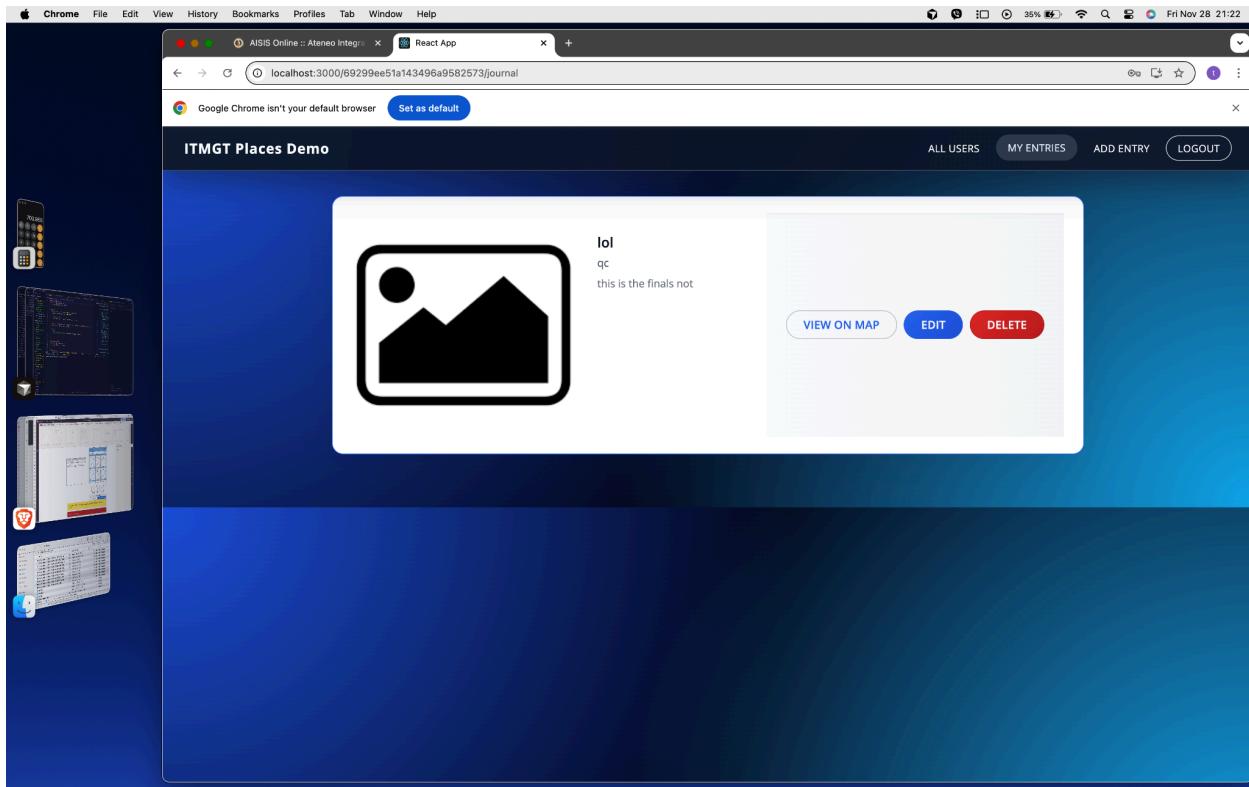
```

Problems Output Debug Console Terminal Ports

Updating entry fields...
Saving entry to database...
Entry saved successfully
Sending success response
==== FETCHING ENTRIES BY USER ID ====
Fetching entries for user: 69299ee51a143496a9582573
User ID type: string
Query: Journal.find({ author: userIdString })
Query filter object: { author: '69299ee51a143496a9582573' }
Database result - Number of entries found: 1
Database result - Entries: [
 {
 coordinates: { latitude: 52.4760892, longitude: -71.8258668 },
 id: new ObjectId('69299f9ed07387824914df40'),
 headline: 'lol',
 journalText: 'this is the finals',
 photo: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSpPkm3Hfm2fa7zZFgK0HQrD8yvwSBmmn_Gw&s',
 locationName: 'qc',
 author: '69299ee51a143496a9582573',
 __v: 0
 }
]
First entry author field: 69299ee51a143496a9582573
Does this userIdString match entry.author? true
Type comparison - userIdString: string entry.author: string
Found 1 entries, preparing response
Sending response with entries
[nodemon] restarting due to changes...
[nodemon] starting node app.js
[dotenv@17.2.3] injecting env (3) from .env -- tip: ✅ audit secrets and track compliance: https://dotenvx.com/ops
[nodemon] restarting due to changes...
[nodemon] starting node app.js
[dotenv@17.2.3] injecting env (3) from .env -- tip: ⚡ run anywhere with `dotenv run -- yourcommand`
[nodemon] restarting due to changes...
[nodemon] starting node app.js
[dotenv@17.2.3] injecting env (3) from .env -- tip: 🛡 prevent committing .env to code: https://dotenvx.com/precommit

Agent terminals are read-only

4. Validation



Console logs of it working

```

tree ./.m2e-workspace
  routes
    JS journal-routes.js
    JS users-routes.js
  > util
  ⚡ .env
  ⚡ .gitignore
  JS app.js
  {} package-lock.json
  {} package.json
  ⓘ README.md
  < frontend
    > node_modules
    > public
    < src
      < journals
        > components
        < pages
          JS NewEntry.js
          # PlaceForm.css
          JS UpdateEntry.js
          JS UserJournal.js
        > shared
      < user
        < components
        OUTLINE
        TIMELINE
      Agent terminals are read-only
    
```

Problems Output Debug Console Terminal Ports

```

[nodemon] starting `node app.js`
[dotenv@17.2.3] injecting env (3) from .env -- tip: 🛡 prevent committing .env to code: https://dotenvx.com/precommit
== UPDATE ENTRY REQUEST ==
Update payload received: { headline: 'lol', journalText: 'this is the finals not' }
Request body received: { headline: 'lol', journalText: 'this is the finals not' }
Entry ID from params: 69299f9ed07387824914df40
Update data - headline: lol journalText: this is the finals not
Attempting to find entry by ID: 69299f9ed07387824914df40
Entry found: Yes
Updating entry fields...
Saving entry to database...
Entry saved successfully
Sending success response
== FETCHING ENTRIES BY USER ID ==
Fetching entries for user: 69299ee51a143496a9582573
User ID type: string
Query: Journal.find({ author: userIdString })
Query filter object: { author: '69299ee51a143496a9582573' }
Database result - Number of entries found: 1
Database result - Entries: [
  {
    coordinates: { latitude: 52.4760892, longitude: -71.8258668 },
    id: new ObjectId('69299f9ed07387824914df40'),
    headline: 'lol',
    journalText: 'this is the finals not',
    photo: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSpPkm3Hfm2fa7zZFgK0HQrD8yvwSBmmn_Gw&s',
    locationName: 'qc',
    author: '69299ee51a143496a9582573',
    <v: 0
  }
]
First entry author field: 69299ee51a143496a9582573
Does this userIdString match entry.author? true
Type comparison - userIdString: string entry.author: string
Found 1 entries, preparing response
Sending response with entries
  
```

Bug B: "The Identity Crisis" (Login Mismatch)

1. The Issue & Reproduction

Root Cause: The Signup route normalized emails (lowercase), but the Login route did not. Users who signed up with capitals (e.g., Test@Gmail.com) could not log in with lowercase (test@gmail.com).

Screenshot of the Login failed error message.

Error LOG:

==== LOGIN ATTEMPT ====

Login Email Input: student@ateneo.edu

Query: User.findOne({ email: email }) - using raw email (case-sensitive)

User query result: User not found

No user found with exact email match (case-sensitive): student@ateneo.edu

Login failed - User not found or password mismatch

2. The Fix

Screenshot of users-controllers.js showing the .normalizeEmail() function added to the login logic.

```
JS journal.js      86  console.log("Querying database for user with email:", email);
JS user.js        87  console.log("Login Email Input:", email);
> node_modules    88  // Normalize email to lowercase for case-insensitive search
  ✓ routes          89  const normalizedEmail = email.toLowerCase().trim();
  JS journal-routes.js 90  console.log("Normalized email for query:", normalizedEmail);
  JS users-routes.js 91  console.log("Query: User.findOne({ email: normalizedEmail })");
  > util            92  existingUser = await User.findOne({ email: normalizedEmail });
  .env              93  console.log("User query result:", existingUser ? "User found" : "User not found");
  ◆ .gitignore       94  if (existingUser) {
  JS app.js          95    console.log("Found user ID:", existingUser.id);
  ◆ package-lock.json 96    console.log("Found user email:", existingUser.email);
  {} package-lock.json 97    console.log("Stored password matches:", existingUser.password === password ? "Yes" : "No");
  }                  98  }
} catch (err) {
```

3. Validation

Screenshot of the User Dashboard successfully loaded after logging in.

A screenshot of a developer terminal window. The left sidebar shows a file tree with files like NewEntry.js, PlaceForm.css, UpdateEntry.js, UserJournal.js, and components. The main pane displays a log of a login attempt:

```
    === LOGIN ATTEMPT ===
    Request body received: { email: 'student@ateneo.edu', password: '00000000' }
    Login attempt for email: student@ateneo.edu
    Password provided: Yes (hidden)
    Querying database for user with email: student@ateneo.edu
    Login Email Input: student@ateneo.edu
    Query: User.findOne({ email: email }) - using raw email (case-sensitive)
    User query result: User found
    Found user ID: 6929a3cd437a80176a26d912
    Found user email: student@ateneo.edu
    Stored password matches: Yes
    Login successful! Sending response
```

At the bottom right, it says "Agent terminals are read-only".

Bug C: "The Phantom List" (Data Retrieval)

1. Investigation (Console Logs)

Screenshot of the Console showing the query running but returning [] (empty array).

Error log

```
==== FETCHING ENTRIES BY USER ID ====
```

```
Querying for UserID: 6929a3cd437a80176a26d91
```

```
Query: Journal.find({ author: userId })
```

```
Database result - Number of entries found: 0
```

```
Query ran but returned 0 results
```

```
Total entries in database: [some number > 0]
```

```
Sample entry author: 6929a3cd437a80176a26d91
```

```
Requested userId: 6929a3cd437a80176a26d91
```

```
Type mismatch possible - userId type: [type] vs entry.author type: [type]
```

2. The Fix

Screenshot of the code in journal-controllers.js showing the fixed query logic (e.g., casting userId or fixing the filter).

```

.env backend ...
FINAL
backend
  controllers
    journal-controller...
  users-controllers.js
models
http-error.js
journal.js
user.js
node_modules
routes
  journal-routes.js
  users-routes.js
util
.env
.gitignore
app.js
package-lock.json
package.json
README.md
frontend
  node_modules
  public
  src
    journals
    components
    pages
      NewEntry.js
      # PlaceForm.css
      UpdateEntry.js
      UserJournal.js
  shared
  user
    components
OUTLINE
 37 // Ensure userId is a string for consistent querying
 38 const userIdString = String(userId);
 39
 40 let entries;
 41 try {
 42   console.log("Query: Journal.find({ author: userIdString })");
 43   console.log("Query filter object:", { author: userIdString });
 44   entries = await Journal.find({ author: userIdString });
 45   console.log("Database result - Number of entries found:", entries.length);
 46   console.log("Database result - Entries:", entries);
 47   if (entries.length > 0) {
 48     console.log("First entry author field:", entries[0].author);
 49     console.log("Does this userIdString match entry.author?", entries[0].author === userIdString);
 50     console.log("Type comparison - userIdString:", typeof userIdString, "entry.author:", typeof entries[0].aut
 51   } else {
 52     console.log("No entries found. Checking if any entries exist in database...");
 53     const allEntries = await Journal.find();
 54     console.log("Total entries in database:", allEntries.length);
 55     if (allEntries.length > 0) {
 56       console.log("Sample entry author:", allEntries[0].author);
 57     }
 58   }
 59   console.log("Requested userIdString:", userIdString);
}

```

Problems Output Debug Console Terminal Ports

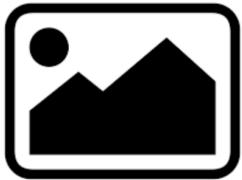
```

== FETCHING ENTRIES BY USER ID ==
Fetching entries for user: 6929a3cd437a80176a26d912
User ID type: string
Query: Journal.find({ author: userIdString })
Query filter object: { author: '6929a3cd437a80176a26d912' }
Database result - Number of entries found: 1
Database result - Entries: [
  {
    coordinates: { latitude: 41.8071182, longitude: -6.7589839 },
    _id: new ObjectId('6929a61cc2b2a9450edae1cc'),
    headline: 'lololll',
    journalText: 'ksjdhfj',
    photo: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSpKm3Hfm2fa7zZFgK0HQrD8yvwSBmnM_Gw&s',
    locationName: 'bgc',
    author: '6929a3cd437a80176a26d912',
    __v: 0
  }
]
First entry author field: 6929a3cd437a80176a26d912
Does this userIdString match entry.author? true
Type comparison - userIdString: string entry.author: string
Found 1 entries, preparing response
Sending response with entries

```

3. Validation

Screenshot of the Frontend showing the list of items correctly rendered.



lololl
bgc
ksjdhfj

[VIEW ON MAP](#) [EDIT](#) [DELETE](#)