BUG Details

What is the bug?

In the login flow, the password comparison is done incorrectly. Instead of using `bcrypt.compare()` to compare the plain-text password provided by the user against the **hashed** password stored in the database, the code directly compares the plain-text password with the hashed password string using `===`.

This means **no user will ever be able to log in**, because `"mypassword123" === "$2b$10$hashedvalue..."` will always be `false`.

Where does it live?

File: `controllers/auth.controller.js` **Line:** ~52

```
// BUGGY CODE
const isMatch = password === user.password;
```

What is the correct fix and why?

The fix is to use the `comparePassword` instance method already defined on the User model (which internally calls `bcrypt.compare()`):

```
// CORRECT CODE
const isMatch = await user.comparePassword(password);
```

Why? When a user registers, their password is hashed using `bcrypt` (via the `pre('save')` hook in `user.model.js`) before being stored in the database. bcrypt hashes are one-way — you cannot reverse them. The only way to verify a password is to use `bcrypt.compare(plainText, hash)`, which re-runs the hashing algorithm and checks if the result matches the stored hash.

Using `===` compares two completely different strings (plain text vs. hash), so it will always return `false`, breaking authentication entirely.