

## Day 22 – Forms, Database, and Authentication

Theme:

Implementing form handling, data validation, and secure user authentication.

---

### Challenge 1 (Easy): Form Handling

**User Story:**

As a user, I want to register via a form so that my data is submitted to the backend.

**Problem Statement:**

Create a registration form and handle form submission at /register using body-parser.

**Expected Outcome:**

Displays "Registration successful for John Doe".

---

### Challenge 2 (Average): Database Integration

**User Story:**

As a developer, I want to store registration data in MongoDB.

**Problem Statement:**

Use **Mongoose** to connect to MongoDB and store new user details in a users collection.

**Expected Outcome:**

Data successfully saved and confirmed via console log.

---

### Challenge 3 (Difficult): Authentication and RBAC

**User Story:**

As an admin, I want to log in securely and access protected routes based on role.

**Problem Statement:**

Implement **Passport.js** for authentication with sessions.  
Allow /admin route access only for users with role "admin".

**Expected Outcome:**

Unauthorized users → Access Denied  
Admin → "Welcome, Admin!"

---

## Best Practices for Day 22

- Always hash passwords before saving (use bcrypt).
  - Store secrets like DB credentials in .env files.
  - Use sessions or JWT for authentication.
  - Keep validation logic server-side, even if done client-side.
-