Module 1 — Project skeleton, auth, DB schema (MVP foundation)

Goal: have backend + frontend skeleton, user auth, MongoDB integration and a working register/login flow.

What to do

1. **Backend kickoff**
    - Create Spring Boot project with: Web, Security, Data MongoDB, WebSocket, Lombok, Validation.
    - Setup MongoDB connection (local or MongoDB Atlas). Keep a .env or application-dev.yml.
    - Implement User model (Mongo document): id, name, email (unique), passwordHash, role, registeredAt.
    - Configure BCrypt password encoding and JWT filter (issue tokens on login). Implement role USER/ADMIN.
    - Endpoints: POST /api/auth/register, POST /api/auth/login, GET /api/users/me.
    - Basic unit tests for register/login.
2. **Frontend kickoff**
    - Create React app (Vite or Create React App). Install axios, react-router, Tailwind.
    - Pages: Register, Login, Dashboard (protected).
    - Auth flow: axios interceptors to include Authorization: Bearer <token>, store token in localStorage (or sessionStorage), protect routes.
    - Simple forms and validations.

Deliverables

- End-to-end register + login working (frontend ⇄ backend).
- Postman collection with register/login calls and saved JWT.
- README with start instructions.

Acceptance criteria (MVP)

- You can register a new user; password stored hashed in MongoDB.
- You can login and receive JWT; protected backend endpoints require valid JWT.
- Frontend stores token and can call protected GET /api/users/me.

Tips (fast wins)

- Use Spring Initializr and a JWT example library snippet (lots of examples). Keep token lifetime short for dev.
- Keep secret keys in environment variables.