Task: Secure the Contact Management API with Spring Security

Background

Our Contact Management API currently exposes all endpoints publicly. This is a security risk because anyone can access and modify contact information without authentication.
We need to implement basic authentication with role-based access control so only authorized employees can interact with the API.

Current State

The API has the following controller available:

```
@RestController
@RequestMapping("/contacts")
public class ContactController {

    @GetMapping
    public String getContacts() {
        return "Returning all contacts";
    }

    @PostMapping
    public String addContact() {
        return "New contact added!";
    }

    @DeleteMapping("/{id}")
    public String deleteContact(@PathVariable int id) {
        return "Contact " + id + " deleted!";
    }

    @GetMapping("/public/info")
    public String publicInfo() {
        return "This is a public endpoint";
    }
}
```

All of these endpoints are currently open to the public.

Requirements

Add Spring Security

- Include Spring Boot Starter Security dependency in the project.
- Use Basic Authentication (username/password).

Authentication Rules

- All endpoints must be secured except /contacts/public/info.
- /contacts/public/info should remain accessible without login.

In-Memory Users

Create the following in-memory users:

Admin user
Username: admin
Password: admin123
Role: ADMIN

Regular user
Username: user
Password: user123
Role: USER

Authorization Rules

GET /contacts → Accessible to both USER and ADMIN.

POST /contacts → Only accessible to ADMIN.

DELETE /contacts/{id} → Only accessible to ADMIN.

GET /contacts/public/info → Accessible to everyone (no login required).

Acceptance Criteria

Public access:

```
curl http://localhost:8080/contacts/public/info
```

✅ Works without authentication.

Regular User access:

```
curl -u user:user123 http://localhost:8080/contacts
curl -u user:user123 -X POST http://localhost:8080/contacts   # Should fail (403)
```

Admin access:

```
curl -u admin:admin123 http://localhost:8080/contacts
curl -u admin:admin123 -X POST http://localhost:8080/contacts
curl -u admin:admin123 -X DELETE http://localhost:8080/contacts/1
```

✅ Admin can read, add, and delete contacts.
✅ Regular user can only read.
✅ Public endpoint works without authentication.

Deliverables

- Updated project with Spring Security enabled.
- In-memory users configured.
- Security rules applied as per requirements.
- All acceptance criteria passing.