Problem 4: Student Management System ----------

```
// Approach: Build a CRUD UI with Thymeleaf, backed by a Student entity and Spring Data JPA.

@Entity
public class Student {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long studentId;
    private String name;
    private int age;
    private String studentClass;
    private String email;
    private String address;

}


public interface StudentRepository extends JpaRepository<Student, Long> {
    List<Student> findByNameContainingIgnoreCase(String name);
    List<Student> findByStudentClass(String studentClass);
}


@Controller
public class StudentController {
    @Autowired
    private StudentService service;

    @GetMapping("/students")
    public String list(@RequestParam(defaultValue = "0") int page, Model model) {
        Page<Student> students = service.getAllStudents(page);
        model.addAttribute("students", students);
        return "student_list";
    }


}
```