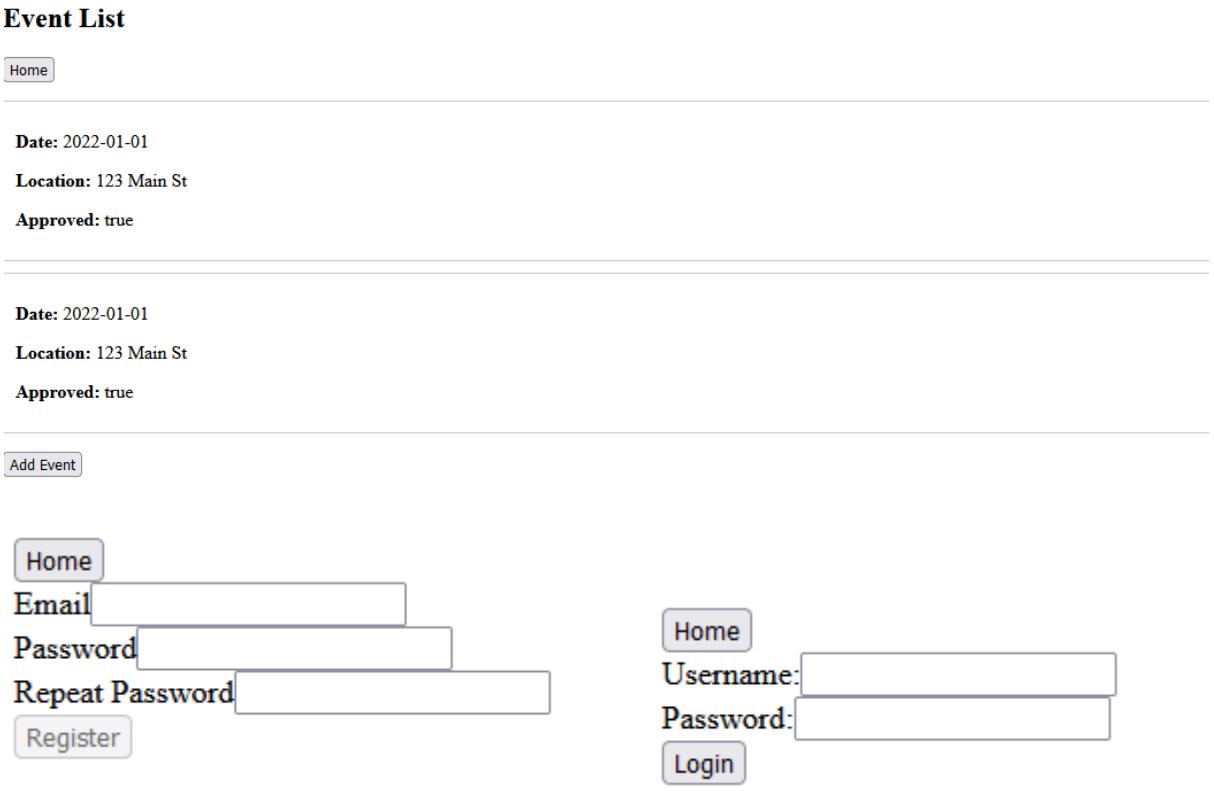
Project Overview

The aim of this project was to enhance our understanding of frontend development and aspect-oriented programming (AOP) by creating a practical application. We developed a frontend application using Angular to interact with a backend service for event management, and implemented AOP in Spring Boot to collect and expose usage statistics of the service.

Frontend Development

The frontend was developed using Angular, allowing users to interact with the backend service through a “user-friendly” interface. The key functionalities include viewing and inserting events, and a prototype of registering and logging in.



Aspect-Oriented Programming (AOP)

The AOP implementation was done using Spring Boot to monitor and collect statistics from the EventService. The primary goal was to track the number of times the getAllEvents method was called.

StatisticsAspect (Spring Boot)

- Uses AOP to intercept calls to getAllEvents and increments a counter each time the method is called.

```
@Aspect
@Component
public class StatisticsAspect {

    2 usages
    private AtomicInteger getAllEventsCallCount = new AtomicInteger( initialValue: 0);

    1 usage  new *
    @Pointcut("execution(* com.example.com_rucinski.service.EventService.getAllEvents(..))")
    public void getAllEventsPointcut() {}

    new *
    @AfterReturning(pointcut = "getAllEventsPointcut()", returning = "result")
    public void afterReturningGetAllEvents(List<EventDTO> result) {
        getAllEventsCallCount.incrementAndGet();
    }
```

StatisticsController (Spring Boot)

- Exposes an endpoint to retrieve the statistics collected by the StatisticsAspect.

```
new *
@RestController
public class StatisticsController {

    @Autowired
    private StatisticsAspect statisticsAspect;

    new *
    @GetMapping("/api/statistics")
    public ResponseEntity getStatistics() {
        return ResponseEntity.ok(statisticsAspect.getGetAllEventsCallCount());
    }
}
```

StatisticsDTO (Spring Boot)

- A simple DTO to structure the statistics data.