

PORTFOLIO

기록하고 경험하는 것을
중요시 하는 백엔드 개발자 김가율입니다.

이름	김가율
생년월일	1991.02.27
이메일	gayulz@kakao.com
Git	https://github.com/gayulz
Tistory	https://yurizzo.tistory.com/



응급구조사 응급실 연계 플랫폼

Springframework 사용 경험 및 숙련도 향상을 위해 진행함

The screenshot displays two main sections of the application:

- 의료 서비스**: A search interface for medical services. It includes dropdown menus for '입원 병상 선택', '증환자실 목록', '소아/유아 격리포함', '수술 및 기타', and '필요 의료 장비'. Below these is a '격리/감염병/코호트' section. On the right is a map of Seoul and surrounding areas with a red marker indicating the user's current location.
- 병원 목록**: A list of hospitals. It shows three entries with their names, addresses, contact numbers, and links to their profiles:
 - 의료법인명지의료재단명지병원 (경기도 고양시 덕양구 화수로 14번길 55 (화정동))
 - 국민건강보험공단일산병원 (경기도 고양시 일산동구 일산로 100 (백석동, 백석1동 1241의 1필지 4층))
 - 순천향대학교부속부천병원 (경기도 부천시 원미구 조마루로 170 (중동))A large map of the same area is shown at the bottom of this section.

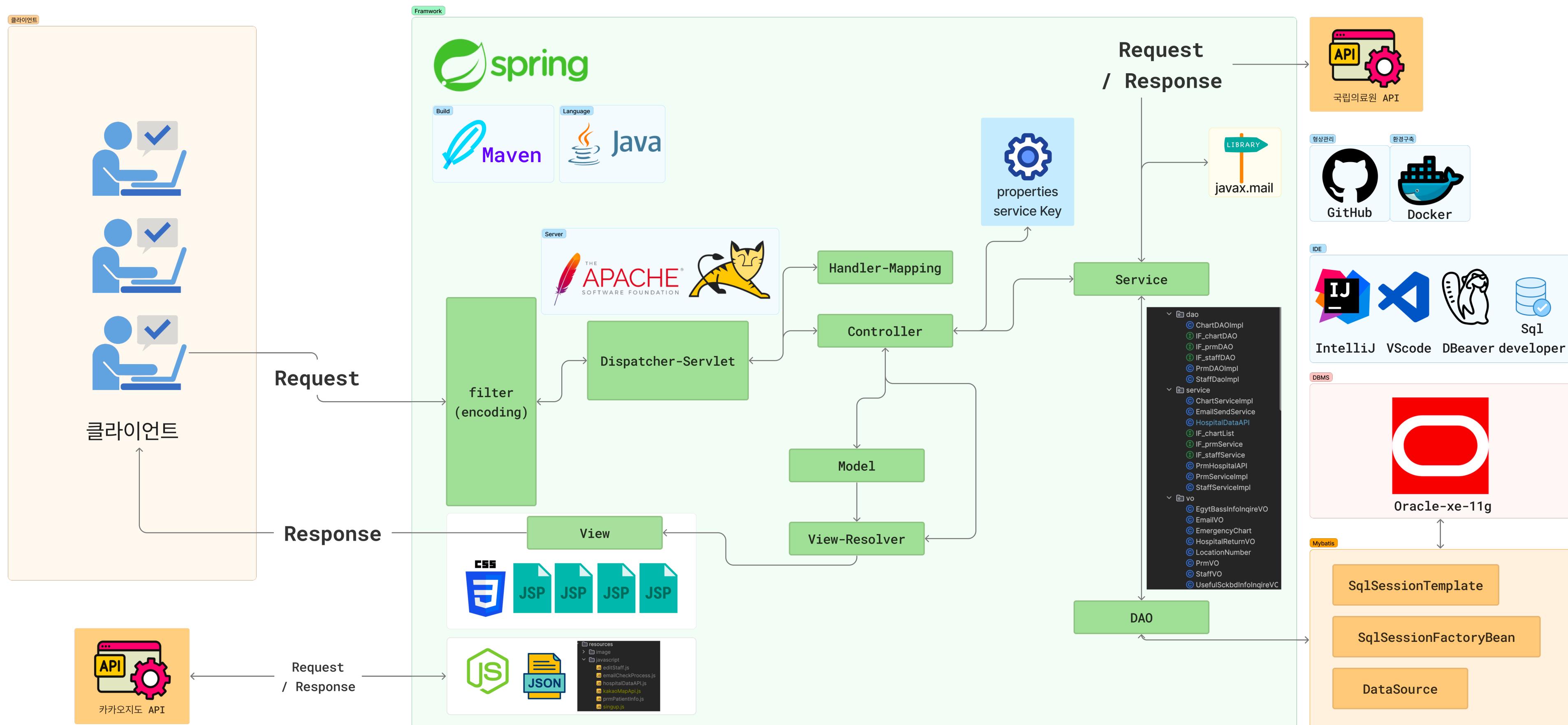
At the bottom of the interface, there is a copyright notice: "Copyright © Developer Yurii
Contact Us :: yurizz.ytistory.com".

Introduce project

작업 기간	2023.12.20 ~ 2024.01.17 (29일)
인력 구성(기여도)	1인 개발 (기여도 100%)
프로젝트 목적	PL/SQL 문법과 프로시저, 시퀀스를 활용한 자동 증가 처리로 PL/SQL 숙련도 향상 JAVA 인터페이스 활용으로 프로그래밍 경험 강화 Spring Framework와 MVC 패턴 적용으로 전반적인 웹 개발 숙련도 향상 동기/비동기 통신을 통한 네트워크 프로그래밍 이해도 증진 Open API 활용 및 통합으로 API 사용 경험 및 숙련도 향상
프로젝트 내용	1) 응급구조사가 필요한 의료 서비스를 검색하면, 해당 서비스를 제공하고 병상이 있는 응급실을 지도로 보여주는 웹 서비스 개발 2) 공공데이터포털의 국립중앙의료원 정보를 활용한 Open API 기반의 데이터 접속 웹 서비스 구현 3) 응급구조사의 접수 목록 확인 및 병원 관계자가 근무 병원에 접수된 응급환자 정보 관리 가능한 의료 서비스 구축
주요 업무 및 상세 역할	1) 데이터베이스 설계, 테이블 생성 및 관리 2) 프로시저 사용하여 Open API 데이터의 안전한 저장 및 관리 3) 시퀀스를 통한 직원번호 및 업체번호 자동 증가 로직 구현 4) 프론트엔드 구조 및 디자인 구현 5) Spring Framework 환경 구축 및 의존성 관리 6) MVC 패턴을 활용한 비즈니스 로직 구현 7) 동기/비동기 통신 로직 개발 8) 이메일 인증 및 6자리 랜덤 숫자 생성 로직 구현
사용언어 및 개발 환경	Front : HTML / CSS / Javascript Server : apache / tomcat Back : SpringFramework / mybatis / JAVA / JSP DB : oracle Tools : IntelliJ / Vscode / Dbeaver / Sqldeveloper Build : Maven SCM : github infra : Docker
참고 자료	https://github.com/gayulz/EMBAD_project

응급구조사 응급실 연계 플랫폼

아키텍처 플로우 차트



의료 서비스

필요한 응급실을 찾아드립니다

[입원실] 외상 전용

중환자실 목록

소아/유아 격리포함

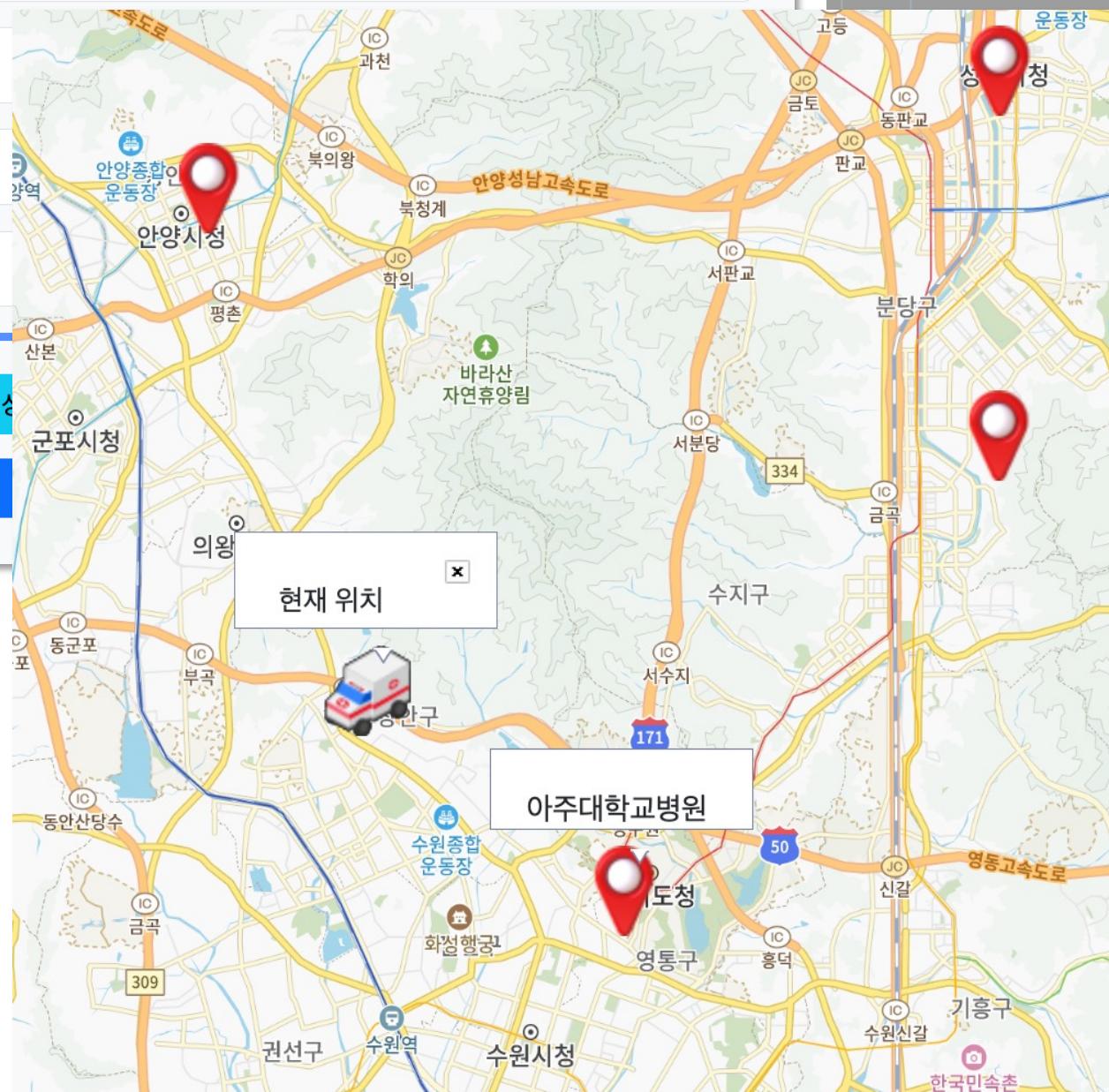
수술 및 기타

필수 의료 장비

격리/감염병/코호트

[입원실] 일반

[입원실] 외상



```
// 구급구조대 필요한 데이터로 파싱
// dom : api로 받아온 데이터를 document로 재조립된 객체
// resultData : 사용자 니즈가 들어가있는 객체 / Key 값이 사용자가 요구하는 값
public List<String> returnMatchHospital(Document dom, HashMap<String, Object> resultData) { Complexity is 12 You must be kidding
    List<String> matchedHpid = new ArrayList<>();
    try {
        XPathFactory xpathFactory = XPathFactory.newInstance();
        XPath xpath = xpathFactory.newXPath();

        // 모든 item 요소 가져오기
        NodeList items = (NodeList) xpath.evaluate( expression: "//item", dom, XPathConstants.NODESET );
        ...
    }
}
```

* 기능 소개

- 1) 사용자로부터 요청받은 의료 서비스를 입력 받아, 선택된 의료 서비스를 검색 버튼 상단에 표시하여 사용자가 선택한 서비스를 확인할 수 있게 하는 기능
- 2) 입력된 데이터를 기반으로 서울 및 경기 지역 응급실 중, 요청받은 의료 서비스를 제공하는 병원을 찾아서 결과를 뷰(view)로 전달하는 기능
- 3) 뷰로 전달될 때, API로부터 받은 XML 형식의 응답 데이터 중 JSP에서 필요한 데이터만을 추출하여, JSON 형식으로 뷰단에 전달하는 기능
- 4) JSP에서 응답을 받은 경우, 자바스크립트 기능을 이용하여 카카오 맵 API로 지도를 랜더링하는 기능
- 5) 지도 위에 사용자의 현재 위치와 병원의 위치를 이미지로 표시하여 사용자에게 제공하는 기능
- 6) 표시된 병원 이름 위에 마우스가 이동하면 병원이름이 사용자에게 보여 지는 기능

자세한 소스 코드 및 프로젝트 소개는

☞ https://github.com/gayulz/EMBAD_project에서 확인하실 수 있습니다.

반려동물과 함께 가는 숙소 찾기

JAVA Swing 라이브러리를 적용, 다양성과 상속을 적용하여 숙련도 향상을 목표로 진행 함



Introduce project

작업 기간	2023.11.20 ~ 2023.12.01 (11일)
인력 구성(기여도)	1인 개발 (기여도 100%)
프로젝트 목적	<ul style="list-style-type: none">1) PL/SQL 문법과 프로시저 처리를 통한 숙련도 향상2) 객체지향 Java 언어의 Class, 상속, 인터페이스 활용으로 이해도 증진3) 디자인 패턴 적용을 통한 개발 경험 강화4) Java와 oracle DB의 연동을 통한 CRUD 작업 경험5) Java Swing와 javax.awt 라이브러리를 활용한 GUI 개발 경험
프로젝트 내용	<ul style="list-style-type: none">1) 반려동물 동반 가능 숙박업체 검색 및 연계 프로그램 개발2) 사용자의 숙소 선호도에 따른 분류, 지역, 날자 입력 기능 및 Oracle DB연동으로 일치하는 숙소 정보 제공3) 숙소 예약, 예약자 개인정보 입력 후 로그인을 통한 예약 정보 확인 기능4) 예약 취소 가능한 예약 시스템 구현
주요 업무 및 상세 역할	<ul style="list-style-type: none">1) 데이터베이스 설계 및 테이블 생성, 관리2) 제약 조건 설정 및 회원 정보 관리3) Java를 사용한 GUI 설계 및 구현4) State패턴 적용을 통한 MVC패턴 구현5) ojdbc 드라이버를 활용 한 DB연결 및 CRUD 기능 구현6) 상속을 이용한 DB연결 관련 드라이버 및 연결 리소스 관리7) 액션 리스너를 활용한 JPanel교체 기능 개발
사용언어 및 개발 환경	JAVA, Oracle, IntelliJ, SqlDeveloper, Dbeaver Docker를 이용한 개발 환경 구축
참고 자료	https://github.com/gayulz/JAVAGUI_project

반려동물과 함께 가는 숙소 찾기

Main work - JPanel 별 버튼에 대한 리스너 처리

```
nSettingInf.java   © Context.java   © RemoteControl.java   © SearchFrame.java   © ActionHandler.java   © LoginFrame.java   © AddReservationDAO.java

private final cancelReservationDAO referencesCancelDao = cancelReservationDAO.getInstance();
private final CheckRoutine check = CheckRoutine.getInstance();
private List<customerANDbookingList> reservationList;
private CustomerDTO customerObject;
private JFrame tempFrame;
private Context tempContext;
private JButton tempBtn;

// 버튼 핸들러
public void handleButtonEvent(JButton source, JFrame frame, Context context) { Complexity is 13
    System.out.println("핸들러 클래스로 진입완료");
    // 멤버변수로 주소 복사
    tempContext = context;
    tempFrame = frame;
    // 진입된 버튼에 대한 리스너를 초기화 하기 위함으로 Context 클래스로 해당 버튼의 주소를 넘겨준다
    tempBtn = source;

    // 숙소검색 버튼 클릭
    if (tempBtn == SearchFrame.getInstance().getSearchBtn()) searchBtnProcess();
    // 내 예약 검색 버튼 클릭
    else if (tempBtn == SearchFrame.getInstance().getMyInfoBtn()) myInfoBtnProcess();
    // booking 버튼 클릭시
    else if (tempBtn == RoomViewFrame.getInstance().getBookingBtn()) bookingBtnProcess();
    // 예약 완료 버튼 클릭시
    else if (tempBtn == BookingFrame.getInstance().getSummitBtn()) summitBtnProcess();
    // 로그인 버튼 클릭시
    else if (tempBtn == LoginFrame.getInstance().getLoginBtn()) loginBtnProcess();
    // 예약 취소 버튼 클릭시
    else if (tempBtn == MyReservationFrame.getInstance().getCancelBtn()) cancelBtnProcess();
    // 모든 버튼이 아닐 때
    else backBtnProcess();

} // 핸들러 통한 프로세스 처리 메서드
// Search Frame에 숙소 검색 버튼 처리로직
private void searchBtnProcess() {
    CheckRoutine check = CheckRoutine.getInstance();
    if (check.validateSearchObject()) {
        tempContext.setState(RoomViewFrame.getInstance());
        tempContext.request();
    } else JOptionPane.showMessageDialog(parentComponent: this, message: "모든 필드를 입력하세요.", title: "오류", JOptionPane.ERROR_MESSAGE);
}
```



* 기능 소개

- 복잡한 사용자 인터페이스를 가진 Java GUI 어플리케이션에서 각 JButton이 수행해야 할 독립된 기능을 명확히 구별하고 관리할 수 있도록, 액션 리스너를 별도의 클래스로 분리하여 효율적으로 구성함.
- 사용자의 다양한 상호작용에 따라 동일한 JPanel 내의 버튼들이 서로 다른 작업을 수행할 수 있도록, State 패턴을 적용하여 상태에 따라 컨텍스트의 행동이 결정되고 변경될 수 있도록 함.
- State 패턴을 활용하여 애플리케이션의 다양한 상태(예: 검색 화면, 예약 화면 등) 간 전환을 원활하게 관리하며, 각 상태에 적합한 사용자 인터페이스를 제공하여 사용자 경험을 향상시킴.
- 코드의 가독성과 재사용성을 향상시키기 위해, JButton 별로 수행해야 할 기능을 메서드로 분리하여 모듈화함.
- 액션 리스너 내에서 정의된 메서드들은 사용자의 입력에 반응하여 다양한 데이터 접근 및 변화를 처리하며, 이를 통해 예약 추가, 예약 취소, 예약 조회와 같은 사용자 요구를 실시간으로 반영함.

자세한 소스 코드 및 프로젝트 소개는

🔗 https://github.com/gayulz/JAVAGUI_project에서 확인하실 수 있습니다.

웹 스토리지 날씨 TODO-List

개인 사이드 프로젝트로 Javascript와 NodeJs 사용 경험 및 숙련도 향상을 목표로 진행



Introduce project

작업 기간	2023.12.23 ~ 2023.12.25 (3일)
인력 구성(기여도)	1인 개발 (기여도 100%)
프로젝트 목적	<ol style="list-style-type: none">Javascript 및 Node.js를 활용한 프로그래밍 경험 증진Geolocation API와 OpenWeather API 사용으로 API통합 개발 스킬 향상동기 및 비동기 API 통신에 대한 이해도 강화 및 실제 적용Web Storage를 사용한 데이터 관리 기법 습득
프로젝트 내용	<ol style="list-style-type: none">사용자의 실시간 위치를 Geolocation API로 파악한 후, 해당 위치를 기반으로 OpenWeather API를 통해 날씨 정보를 조회하고, 그 결과를 시각적으로 표현하는 웹사이트 개발Web Storage를 사용하여, 웹 페이지나 PC 재부팅 후에도 Todo-List 데이터를 유지하는 기능 개발사용자가 관리하는 Todo-List 항목을 개별적으로 삭제하거나, 완료 체크박스를 통해 관리할 수 있는 사용자 친화적인 웹사이트 구축
주요 업무 및 상세 역할	<ol style="list-style-type: none">Node.js 및 JavaScript로 소스 코드 작성프론트엔드 구현API 호출 및 데이터 통신 처리npm 및 yarn을 사용하여 라이브러리 관리Express와 dotenv 라이브러리를 이용한 백엔드 환경 구축.env 파일을 이용한 API 키 보안 관리Express를 통해 백엔드에서 API 키를 호출하고 프론트엔드로 전달하는 기능 구현
사용언어 및 개발 환경	HTML5, CSS3, JavaScript, Node.js Geolocation API, OpenWeather API VS Code Web Storage npm, yarn, express, dotenv
참고 자료	https://github.com/gavulz/TODOLIST_project

웹 스토리지 날씨 TODO-List

Main work - API 통신 처리 로직

The screenshot shows the browser's developer tools open to the Application tab. The 'Storage' section is selected. It lists two main items: 'saved-items' and 'saved-weather'. The 'saved-items' item contains an array of objects representing todos. The 'saved-weather' item contains an object representing the current weather in Suwon-si.

키	값
saved-items	[{"contents": "안녕하세요", "complete": true}, {"contents": "오늘은 일요일입니다", "complete": false}, {"contents": "좋은 하루입니다", "complete": false}]
__mantle_tile_meta_data	{"tileTypes": ["basic", "satellite", "terrain"], "version": 1}
saved-weather	{"location": "Suwon-si", "weather": "Clouds"}

*** 기능 소개**

- 1) 사용자의 편의성을 고려하여 데이터베이스 대신 Web Storage를 사용하여 데이터를 로컬에 저장함
- 2) 웹 사이트가 재부팅 되거나 PC가 재 시작 되어도 사용자의 정보와 설정이 유지 됨
- 3) 할일 목록에 대한 생성, 완료 표시, 삭제기능을 단일 페이지 어플리케이션(SPA)내에서 자연스럽게 처리할 수 있도록 함수를 이용해 기능을 세분화 하고 모듈화 했음
- 4) 웹 사이트에 접속 할 때마다 Web Storage에 저장된 할 일 목록과 날씨 데이터를 자동으로 불러와 사용자에게 즉각 표시하도록 하여, 사용자 경험을 향상 시켰습니다.

자세한 소스 코드 및 프로젝트 소개는 https://github.com/gayulz/TODOLIST_project에서 확인하실 수 있습니다.

```
const deleteAll = function () {
  const liList = document.querySelectorAll('li');
  for (let i = 0; i < liList.length; i++) {
    liList[i].remove();
  }
  saveItemsFn();
};

const saveItemsFn = function () {
  const saveItems = [];
  for (let i = 0; i < todoList.children.length; i++) {
    const todoObj = {
      contents: todoList.children[i].querySelector('span').textContent,
      complete: todoList.children[i].classList.contains('complete'),
    };
    saveItems.push(todoObj);
  }

  if (saveItems.length === 0
    ? localStorage.removeItem('saved-items')
    : localStorage.setItem('saved-items', JSON.stringify(saveItems)));
};

if (savedTodoList) {
  for (let i = 0; i < savedTodoList.length; i++) {
    createTodo(savedTodoList[i]);
  }
}
```