Allba 대타 스케줄러

Freshman 새내기들

김지은, 유영서, 이가은, 한수정

새내기들

TABLE of CONTENTS

프로젝트 소개

팀원 소개

김지은 FrontEnd

IT공학전공
경영학과
팀장

- 홈 화면 구현
- 스케줄러 페이지
- 지점 찾기 페이지
- 로그인/로그아웃
- 회원가입
- 대타 등록하기 기능 구현

유영서 FrontEnd

영어영문학전공
컴퓨터과학전공

- 마이페이지 구현
- 참가 요청 modal 및 조회 구현
- 로그인/로그아웃
- 회원가입

이가은 BackEnd

컴퓨터과학전공

- 스케줄러 구현
- 지점 등록 및 조회, 삭제 기능
- 지점 검색 기능
- 알바 대타 요청
- 마이페이지 구현

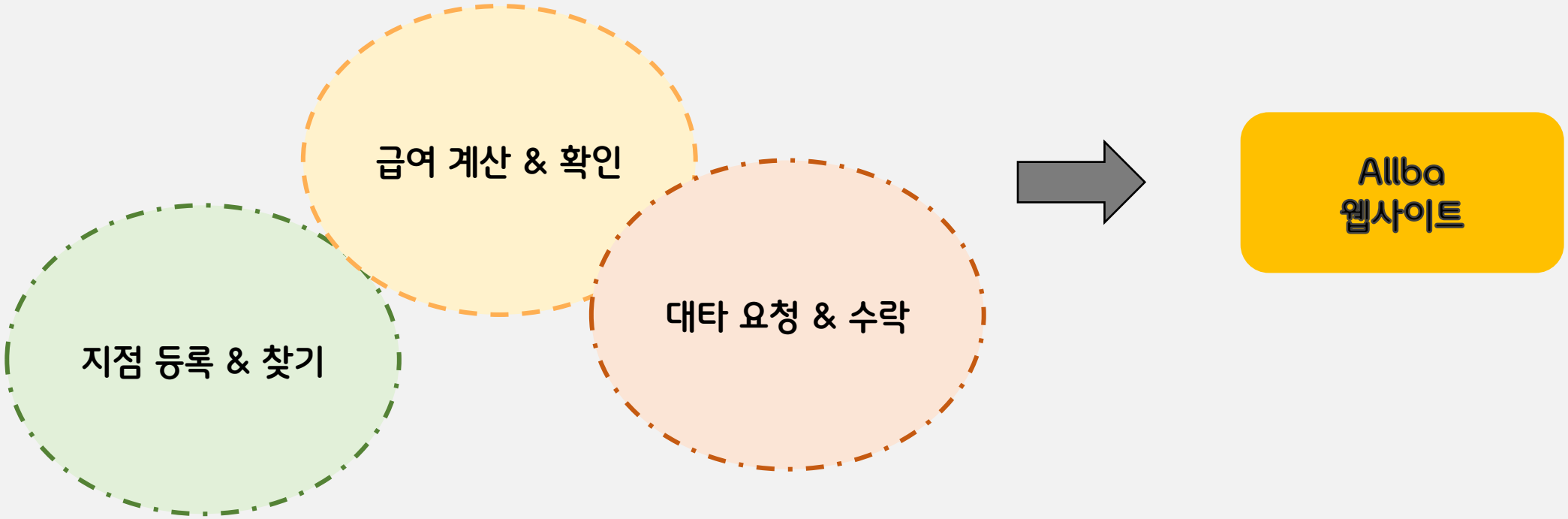
한수정 BackEnd

IT공학전공

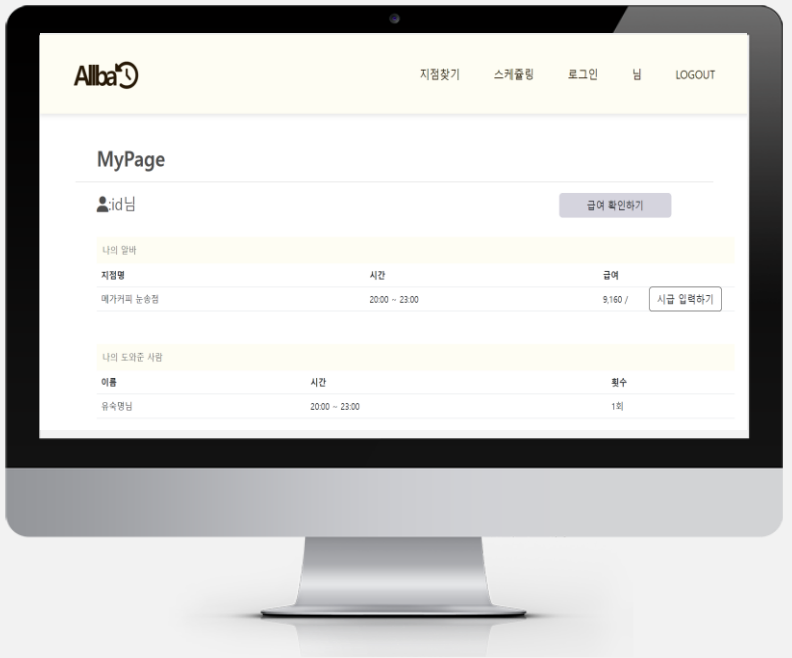
- 로그인/로그아웃
- 회원가입
- 지점 참가 요청 및 참가 조회 기능
- 알바 대타 요청/수락 기능
- 마이페이지 구현

알바 대타 매핑 서비스란?

: 매번 대타를 구할 때마다 어색한 동료에게 연락을 취해야 했던 당신을 위한 알바 대타 매핑 서비스!



프로젝트 소개

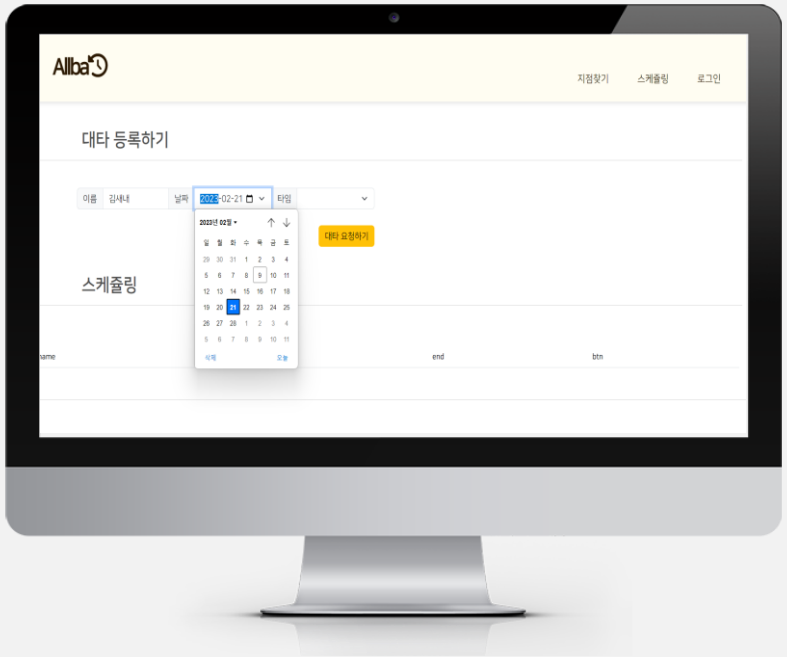


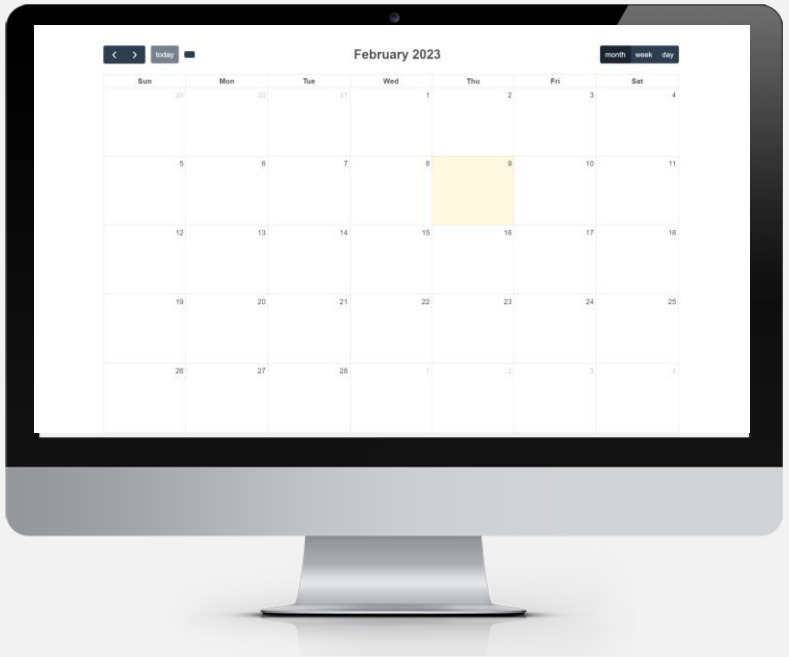
자신의 아이디와 비밀번호를 입력하고 로그인을 하세요!

마이페이지에서 본인의 알바 정보와 급여를 확인할 수 있답니다

나만의 캘린더로 자신의 일정을 관리 해보세요

프로젝트 소개





알바 대타를 손쉽게 구해봅시다!

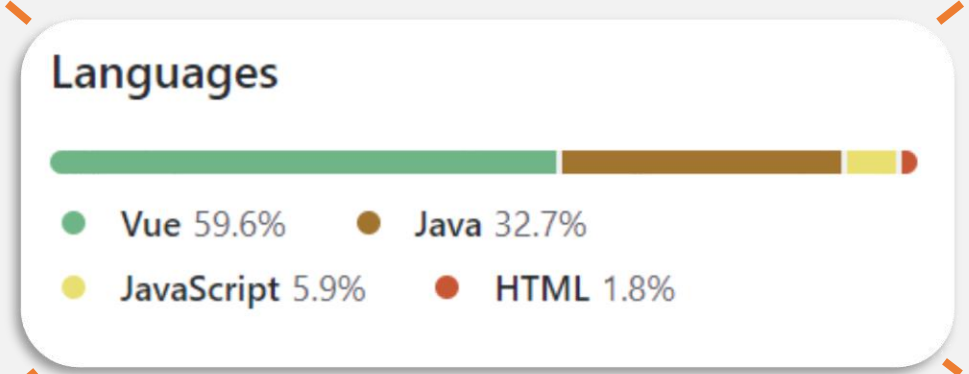
대타 요청과 대타 수락 기능으로 자동으로 변경되는 스케줄

변경된 스케줄로 계산된 급여를 확인하세요.

프로젝트 소개

기술 스택

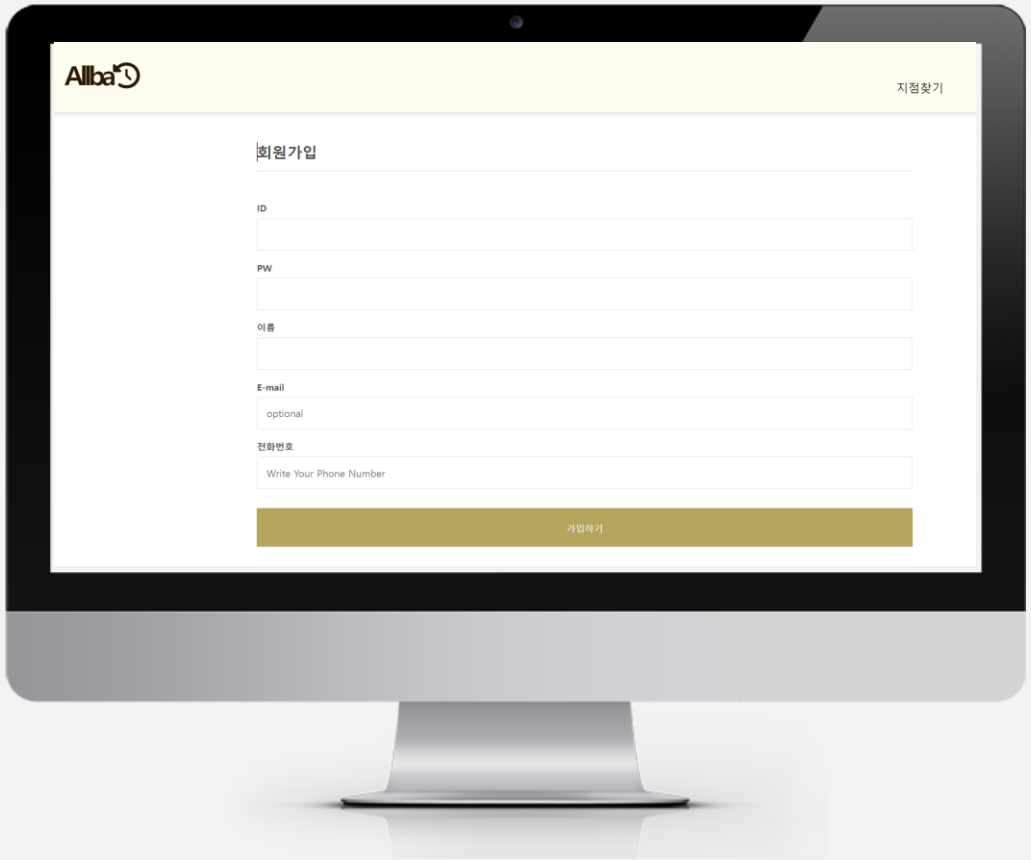
mongoDB



AXIOS

seok830621

회원가입 PAGE



회원가입

```
methods: {
    async submitForm() {
        axios.post('http://localhost:9090/user/new', {
            pid: this.pid,
            pwd: this.pwd,
            name: this.name,
            phone: this.phone,
            email: this.email,
        })
            .then(res => {
                console.log(res);
                alert(this.pid + '님 환영합니다.')
                this.$store.commit('setName', this.name);
                this.$router.push("/main");
            },


            ).catch(err => {
                console.log(err);
            })
    },
```

아이디와 비밀번호, 이름, 전화번호,
이메일을 입력한 후
회원가입 버튼을 클릭하면

- Id와 함께 환영한다는 팝업창을 띄우는 axios 안의 alert문
- 홈 화면으로 이동

회원가입

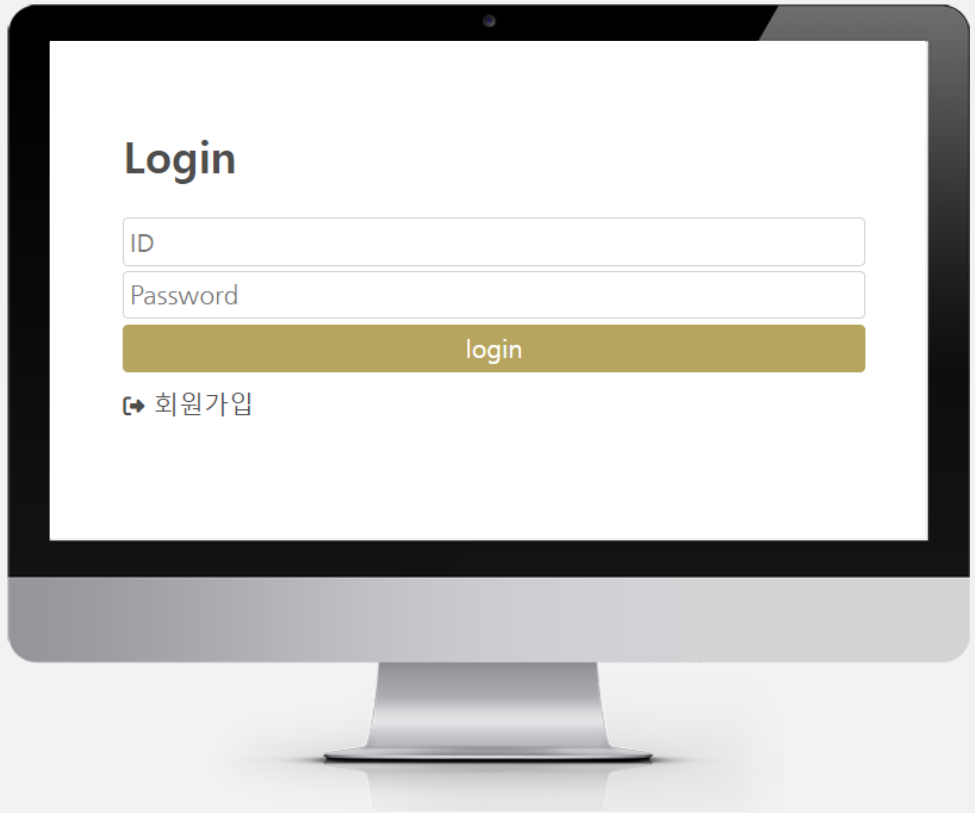
```java
@RequiredArgsConstructor
@RestController
public class MembersApiController {
    2개 사용 위치
    private final MembersService membersService;
    gaeunlee
    @PostMapping("/user/new")
    public String save(@RequestBody MembersSaveRequestDto requestDto) { return membersService.save(requestDto); }

    0개의 사용위치  gaeunlee
    @GetMapping("/user/{id}/mypage")
    public Object findById(@PathVariable String id) { return membersService.findByPid(id).get(); }

}
```

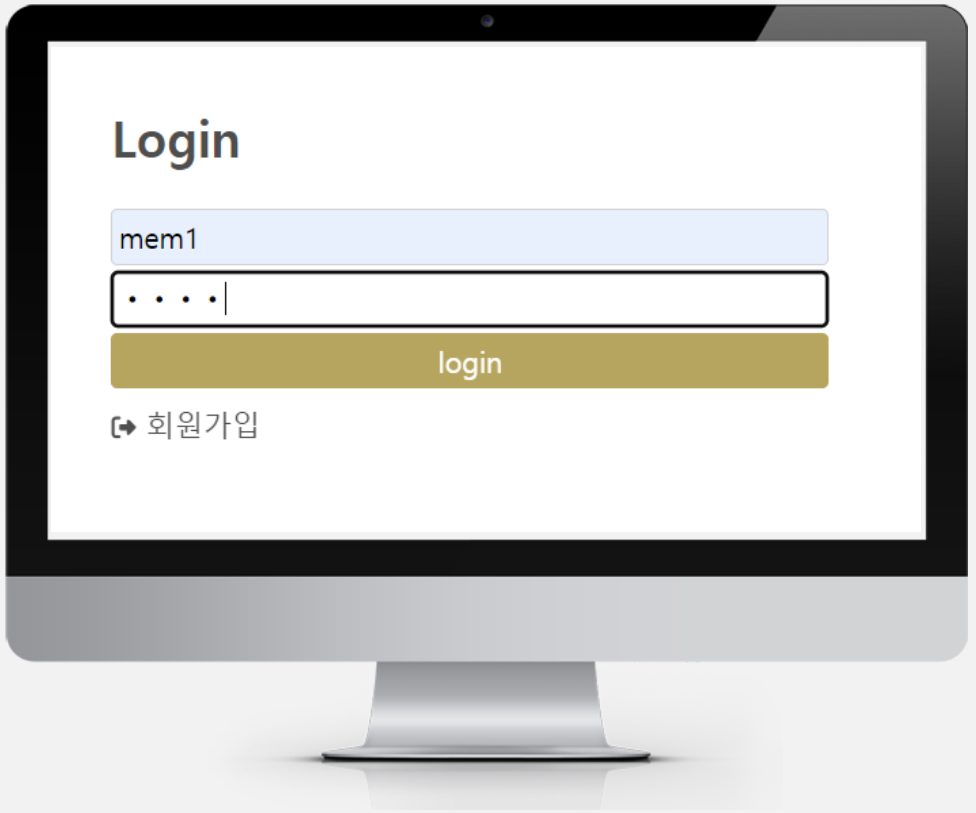
- 회원가입에 필요한 데이터들을 DTO 객체로 담아 전달
  (id, pw, email, name)
  -> 연결된 DB 에 저장

- GET을 통해 DB에 저장된 회원 정보 데이터 불러오기

seok830621

Log in/out PAGE





로그인/로그아웃

```
methods: {
  async submitForm() {
    axios
      .post("http://localhost:9090/login", {
        pid: this.pid,
        pwd: this.pwd,
      })
      .then((res) => {
        console.log(res);
        console.log(this.pid);

        this.$store.commit("setPid", this.pid);

        alert(this.pid + "님 어서오세요");
        this.initForm();
        this.$router.push("/main");
      })
      .catch((err) => {
        console.log(err);
      });
  },
  initForm() {
    this.pid = "";
    this.pwd = "";
  },
```

아이디와 비밀번호를 입력한 후
로그인 버튼을 클릭하면

- Id와 함께 팝업창을 띄우는 axios 안의 alert문
- 홈 화면으로 이동

로그인/로그아웃

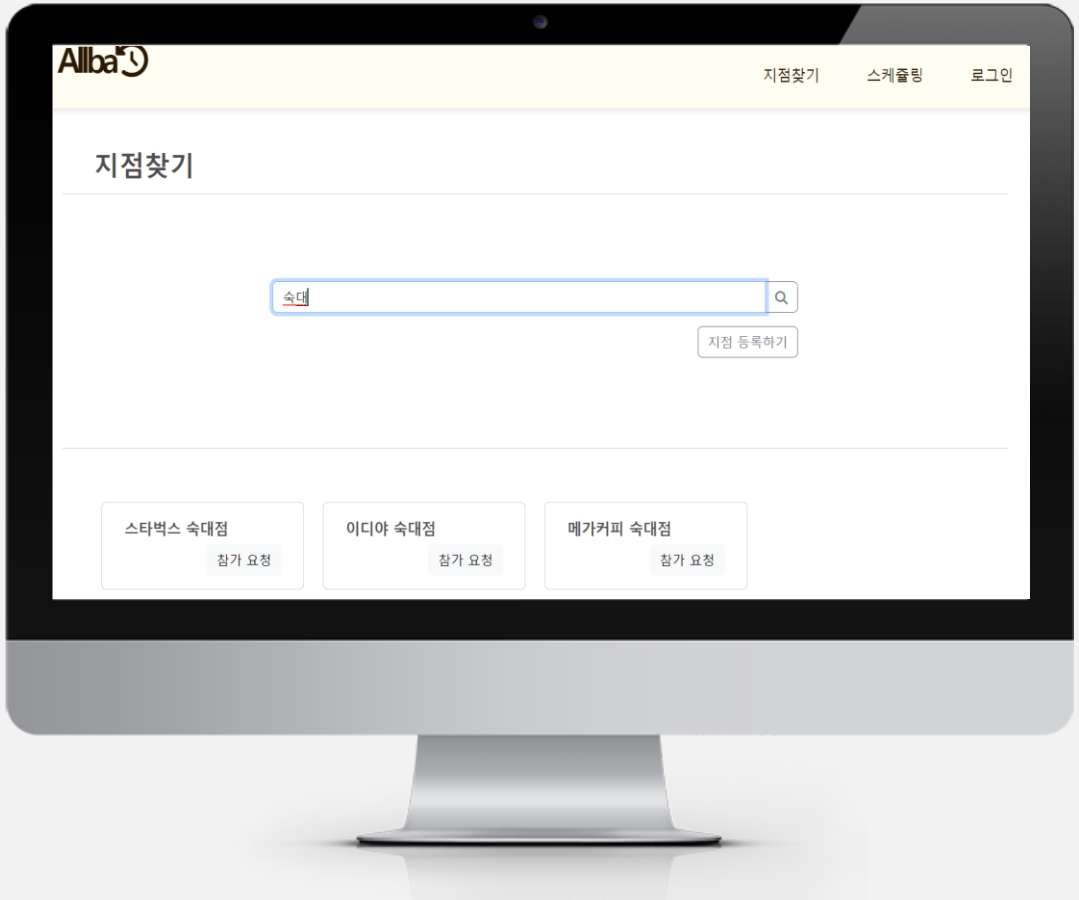
```java
@PostMapping("/login")
@ResponseBody
public String login(@RequestBody @Validated LoginDto loginDto,
                    BindingResult bindingResult,
                    @RequestParam(defaultValue = "/") String redirectURL, HttpServletRequest request) {
    Members loginMember = loginService.loginMem(loginDto.getPid(), loginDto.getPwd());
    if (loginMember == null) {
        bindingResult.reject( errorCode: "Fail",  defaultMessage: "아이디 또는 비밀번호가 맞지 않습니다.");
        System.out.println("입력값"+loginMember);
        return "false";
    }else if (bindingResult.hasErrors()) {
        String n = String.valueOf(bindingResult);
```

```java
public interface LoginSession {
```

- 아이디와 패스워드를 입력받아 실시간 DB 조회를 거쳐 계정을 확인하고, 조합이 맞는지 검증 과정을 거침
- 로그인에 성공하면, 세션을 통해 값 정보 저장/전달

seok830621

지점찾기 PAGE



실시간 지점 검색 기능

```
<hr class="line" />
<div class="p-5 10" style="width: 88%; margin-left: 6%">
  <div class="row g-5">
    <div
      v-for="user in users"
      v-if="user.companyName.includes(searchName)"
      class="col-3"
    >
      <div class="row">
        <!-- <div class="col-sm-4 col-lg-3"> -->
        <div class="card">
          <div class="card-body">
            <h5 class="card-title mt-1">{{ user.companyName }}</h5>
            <p class="card-text mt-1 mb-2">{{ user.companyDescription }}</p>
            <div class="d-flex flex-row-reverse gap-2 mt-2">
              <button
                class="btn btn-light"
                id="show-modal"
                @click="showModal = true"
              >
                참가 요청
              </button>
            </div>
```

User 데이터를 가져온 다음,
찾고자 하는 지점명(searchName) 을
회사명이 포함하고 있다면 실시간으로
확인할 수 있게 구현

지점 등록 및 실시간 지점 검색 기능

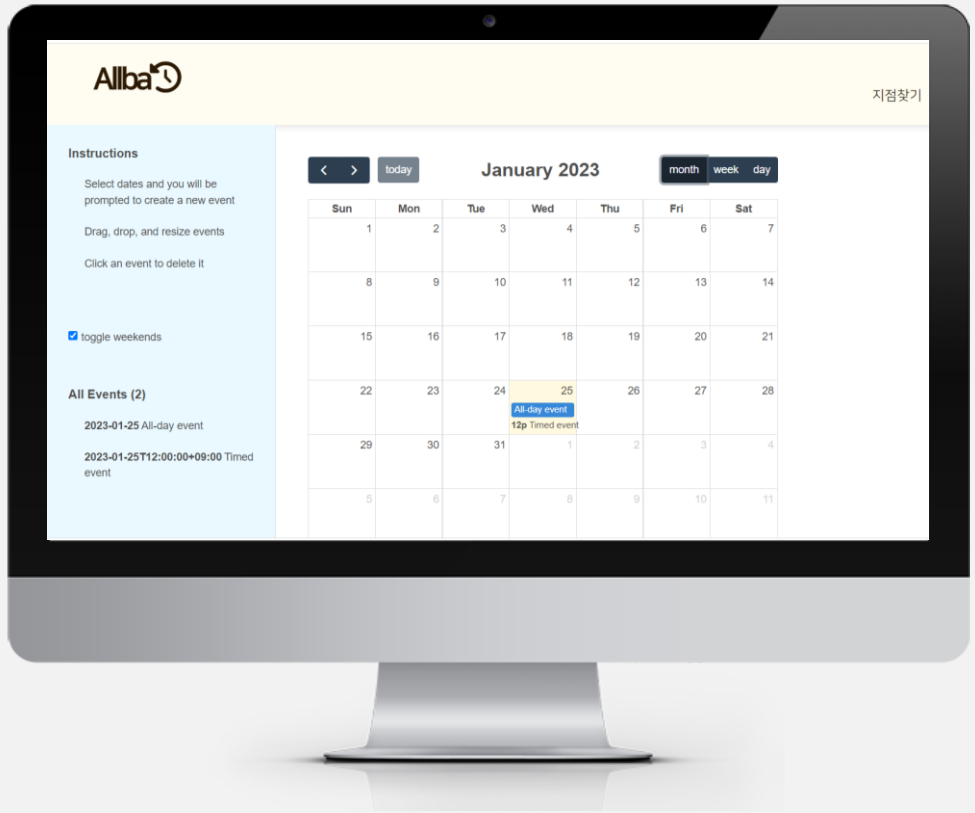
```java
@PostMapping(value = "/new")
public String createNewCompany(@RequestBody Company company) {
    System.out.println("createNewCompany");
    companyService.registerCompany(company);
    return "company";
}
```

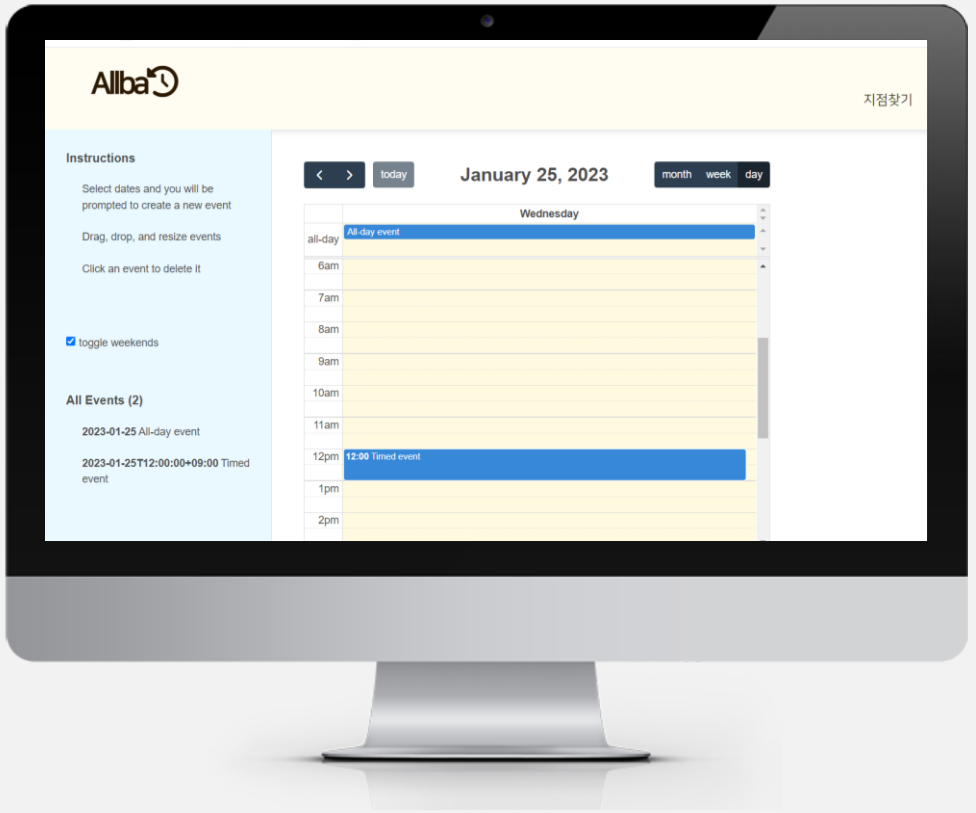
```java
@GetMapping("")
public List<Company> viewAllCompany() { return companyService.findAllCompanies(); }

0개의 사용위치  gaeunlee
@GetMapping("/{companyName}/find")
public List<Company> searchByCompanyName(@PathVariable String companyName) {
    return companyService.findByCompanyName(companyName);
}
```

DB에 지점 등록하고
companyName을 통해
DB에 등록된 지점을 조회하고
GET 출력

seok830621

캘린더 PAGE





캘린더 api 구현

```
async getFull() {
  var cal = this;
  axios
    .get("http://localhost:9090/scheduler/{companyCode}")
    .then(function (response) {
      console.log(response.data);
      cal.calendarOptions.events = response.data;
    })
    .catch(function (error) {
      console.log(error);
    });
},
handleWeekendsToggle() {
  this.calendarOptions.weekends = !this.calendarOptions.weekends; // update a property
},
handleDateSelect(selectInfo) {
  let title = prompt("Please enter a new title for your event");
  let calendarApi = selectInfo.view.calendar;
  calendarApi.unselect(); // clear date selection
  if (title) {
    calendarApi.addEvent({
      id: createEventId(),
      title,
      start: selectInfo.startStr,
      end: selectInfo.endStr,
      allDay: selectInfo.allDay,
    });
  }
```

- 데이터 등록 및 조회
- 데이터 베이스 연결을 통해 이름, 근무 시간, 대타 요청 여부 확인 가능

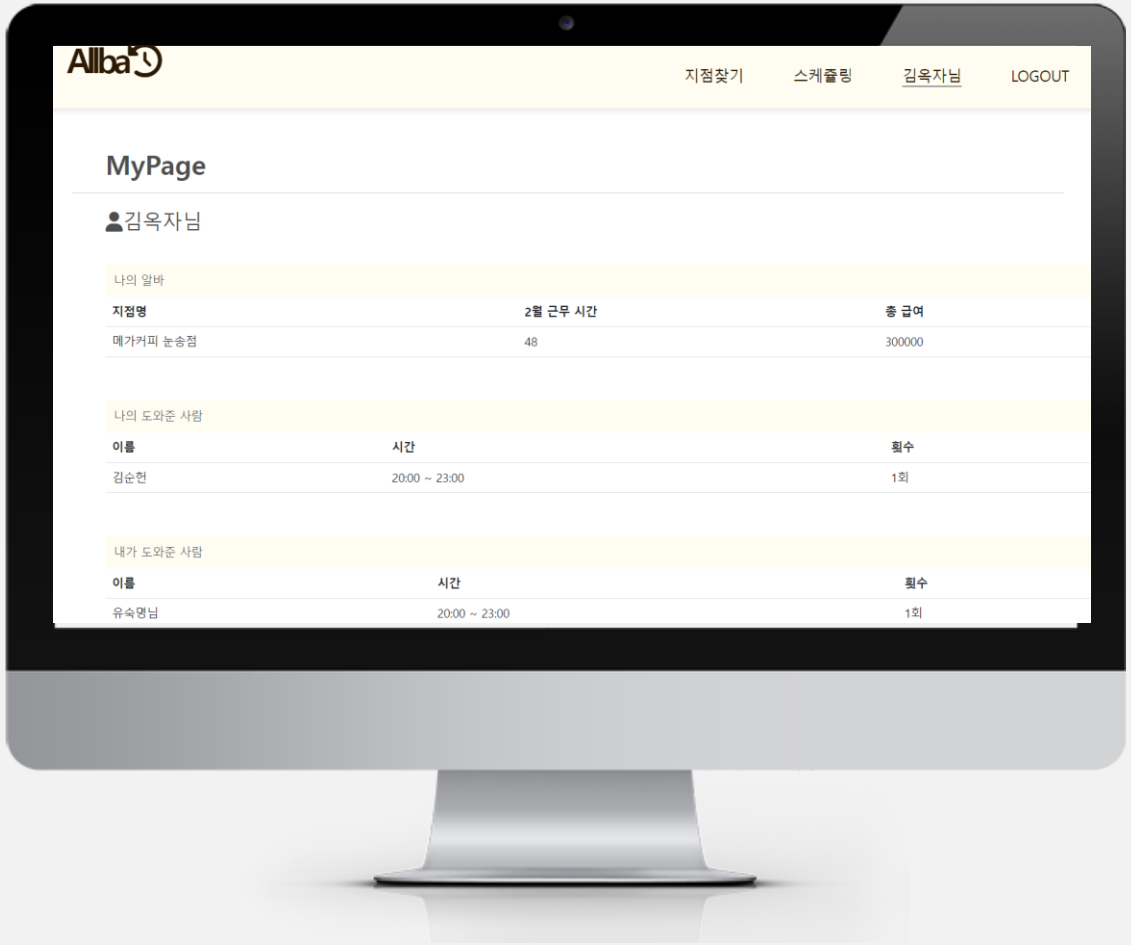
캘린더 api 구현

```
@PostMapping("/{companyName}/new")
public String createNewSchedule(@PathVariable String companyName, @RequestBody Help scheduler) {
    scheduler.setCompanyName(companyName);
    schedulerService.addSchedule(scheduler);
    return "newSchedule";
}

0개의 사용위치   gaeunlee
@GetMapping("/{companyName}")
public List<SchedulerFullCalendar> viewAll(@PathVariable String companyName) {
    return schedulerService.findSchedulers(companyName);
}
```

- 데이터 베이스의 이용자의 스케줄과 대타 요청을 받아 캘린더에 출력

seok830621

MY PAGE



대타 요청 및 수락

```
// 대타 요청
hancrysta1
@PostMapping("scheduler/help/request/")
public Help save(@RequestBody HelpSaveRequestDto requestDto){
    return helpService.save(requestDto);
}
```

```
// 대타 수락
gaeunlee +1
@PatchMapping("scheduler/help/{id}/allow")
public void allow(@PathVariable String id, @RequestBody HelpAllowRequestDto dto) {
    helpService.allow(id, dto);
}
```

- 대타 요청

세션을 통해 이용자의 id와 이름을 받아와 대타 요청을 하고,
대타 수락을 한 이용자가 그 요청 코드를
받아 DB 업데이트

감사합니다