

React Scss Bootstrap

PARKET

Parking System Program

25-06-23 ~ 25-06-30

team project | 김가람 남어진 박수빈 최유리

Timeline

Mo	Tu	We	Th	Fr
6/23	6/24	6/25	6/26	6/27
6/30				

6/23
_서치 및 스타일 가이드
_main와이어프레임
_main프로토타입
_main 코드 구현 시작

6/24
_main 코드 정리
_main 스타일 통합

6/25
_main 최종 수정
_sub 와이어프레임
_sub 프로토타입

6/26
_sub 코드 구현
_sub 코드 정리

6/27
_sub 스타일 통합
_sub 페이지 링크 연결

6/3
_오류 수정
_문서작업

Contents

Topic Selection

- 004 Target User
- 005 Brand Story

Design Process

- 006 Persona 1_오경아
- 007 Persona 2_정민우
- 008 Style Guide
- 009 Wireframe
- 010 Prototype

Code Implementation

- 011 main_HeroSection
- 012 main_DisableIfEmpty
- 013 main_DisableIfEmpty
- 014 main_Key Features 1
- 015 main_Key Features 2
- 016 main_AutoSwiper
- 017 main_onClick-Event
- 018 sub_DetailPage
- 019 sub_onClick-findIndex
- 020 sub_Search
- 021 sub_Remaining
- 022 sub_PageButton

Team Project

- 023 team-project_review

Target User

출퇴근으로 바쁜

주차 고민이 많은

짧은 시간도 아끼고 싶은

여성들

복잡한 절차를 싫어하는

전기차 충전 고민이 많은

몸이 불편한

도심 주차가 일상인

배려 주차구역을 자주 찾는

충전과 주차를 한 번에 해결하고 싶은

전기차를 사용하는 30~40대 직장인

Brand Story

ParKet - 주차의 모든 순간을 스마트하게

우리는 30~40대 전기차 오너의 시간을 소중히 생각합니다.

출근길, 아이 등·하원, 주말 장보기...

잠깐의 주차가 소중한 일상을 방해하지 않도록,

ParKet은 주차의 모든 과정을 더 쉽고, 빠르고, 똑똑하게 만듭니다.

ParKet은 Ticket과 Park의 합성어로,

단순한 주차 공간을 넘어 스마트한 이용권을 뜻합니다.

PARKET○

Persona 1 오경아

PROFILE



오경아(여)

나이 : 45세

거주 지역 : 경기 수원시

가족 관계 : 남편, 딸, 강아지

직업 : 유치원 교사

성격 : 따뜻하고 밝다

취미 : 베이킹

관심사 : 딸과 카페가기

기분 : 딸과 둘만의 추억이 생겨 행복하다

인터넷 활용 수준 : SNS 관리가 능숙하다

BIO

경아는 유치원 교사로 일한 지 벌써 20년 차가 다 되어 간다. 올해 16살이 된 딸이 하나 있는데 주말마다 함께 카페를 가는 것이 취미다. 딸이 좋아하는 느낌 좋고 분위기 좋은 카페를 자주 가는 편이다. 카페를 갈 때 항상 차를 직접 운전해 이동하는데 그럴 때마다 주차나 충전 문제가 생겨 고민이다. 인기가 많은 카페를 갈 때면 자리가 없어 근처를 몇 바퀴 돌 때도 있다. 전기차를 몰고 있는 경아는 가끔 충전 할 곳이 없어 마음이 조급해질 때도 많았다. 주차나 충전에 신경을 덜 쓸 수 있다면 더 여유롭게 시간을 보내지 않을까 생각한다.

무엇보다 딸과 보내는 주말이 소중하기 때문에, 불필요한 스트레스는 줄이고 싶은 마음이다. 경아 씨는 주차 문제 때문에 가끔 딸과 싸우게 될 때 마음이 굉장히 불편하다. 충전 가능 여부나 주차 공간을 미리 확인할 수 있으면 하는 마음이다.

PAIN

- 전기차 충전이 가능한 곳은 따로 찾아야 해서 복잡하다.
- 충전과 주차 정산을 따로해야 해서 번거롭다.
- 이것저것 복잡한 조작은 너무 귀찮다.

NEEDS

- 실시간으로 충전이 가능한 곳이 있는지 보고 싶다.
- 한 번에 정산해 이중 결제 되는 느낌을 받고 싶지 않다.
- 간단하고 빠르게 결제하고 싶다.

Persona 2 정민우

PROFILE



정민우(남)

나이 : 33세

거주 지역 : 서울시 영등포구

가족 관계 : 엄마, 아빠, 남동생

직업 : 디지털 마케터

성격 : 걱정이 많지만 쾌활하다

취미 : 영화 특전 모으기

관심사 : 영화에 대한 이야기 하기

기분 : 요즘 주차할 자리가 없어 힘들다

인터넷 활용 수준 : SNS 관리가 능숙하다

BIO

민우는 5년 차 디지털 마케터이다. 걱정이 많은 성격이지만, 평소엔 쾌활하고 사람들과 잘 어울린다. 영화 특전을 모으는 것을 굉장히 좋아해 평일과 주말 상관없이 시간이 나면 영화관을 꼭 간다. 민우는 평소 자기 차를 몰고 다니는데, 주차할 공간이 부족해 시간을 허비하는 날들이 많다. 특히 무대인사를 하는 날이면 인파가 엄청 몰려 주차하기가 더 어려워진다. 무대인사는 오전부터 오후까지 하기에 그걸 다 찾아보고 영화를 예매하는 것도 힘들다. 민우는 평소 즉흥적으로 영화를 예매하고 보러 가기 때문에 실시간으로 주차장 잔여 자리를 알고 싶은데 그럴 수 없어 불편하다.

마음 편하게 주차하고 영화 시간에 늦고 싶지 않은 민우는 잔여 자리를 실시간으로 확인하고 싶다.

PAIN

- 영화관 근처 또는 건물 내에 주차 공간이 부족해 시간을 허비한다.
- 주차장의 자리 잔여 수를 미리 알 수가 없다.
- 주차 가격을 미리 알 수가 없어 불편하다.

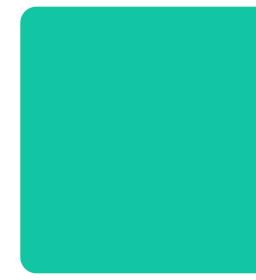
NEEDS

- 실시간으로 주차 및 충전 가능 여부를 알고 싶다.
- 잔여 자리가 몇인지 혼잡도가 어느정도인지 정확히 알고 싶다.
- 사전에 주차 요금을 미리 알고 싶다.

Style Guide

color-style

point-color



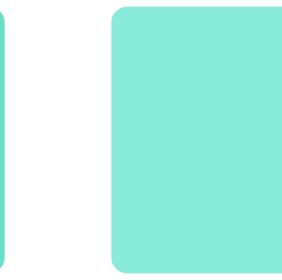
#14C6A4



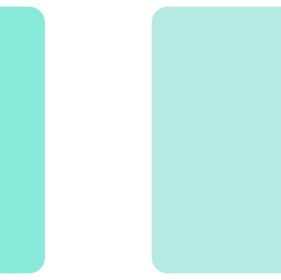
#39D5B8



#69DFC9



#88EADA



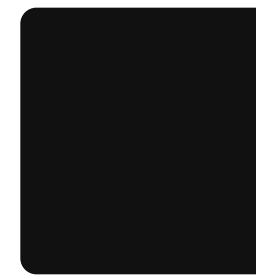
#B6EBE2

base-color



#C4F2E8

font-color



#121212



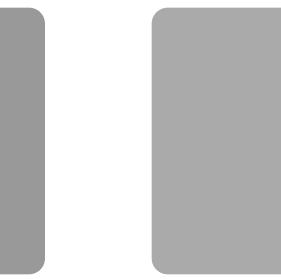
#333333



#666666



#999999



#AAAAAA



#EEEEEE

font-style

heading_Noto Sans h1 Bold 38px

heading_Noto Sans h2 Bold 32px

heading_Noto Sans h3 SemiBold 28px

heading_Noto Sans h4 Medium 24px

heading_Noto Sans h5 Medium 20px

heading_Noto Sans h6 Medium 18px

body_Noto Sans P Regular 16px

body_Noto Sans P Regular 15px

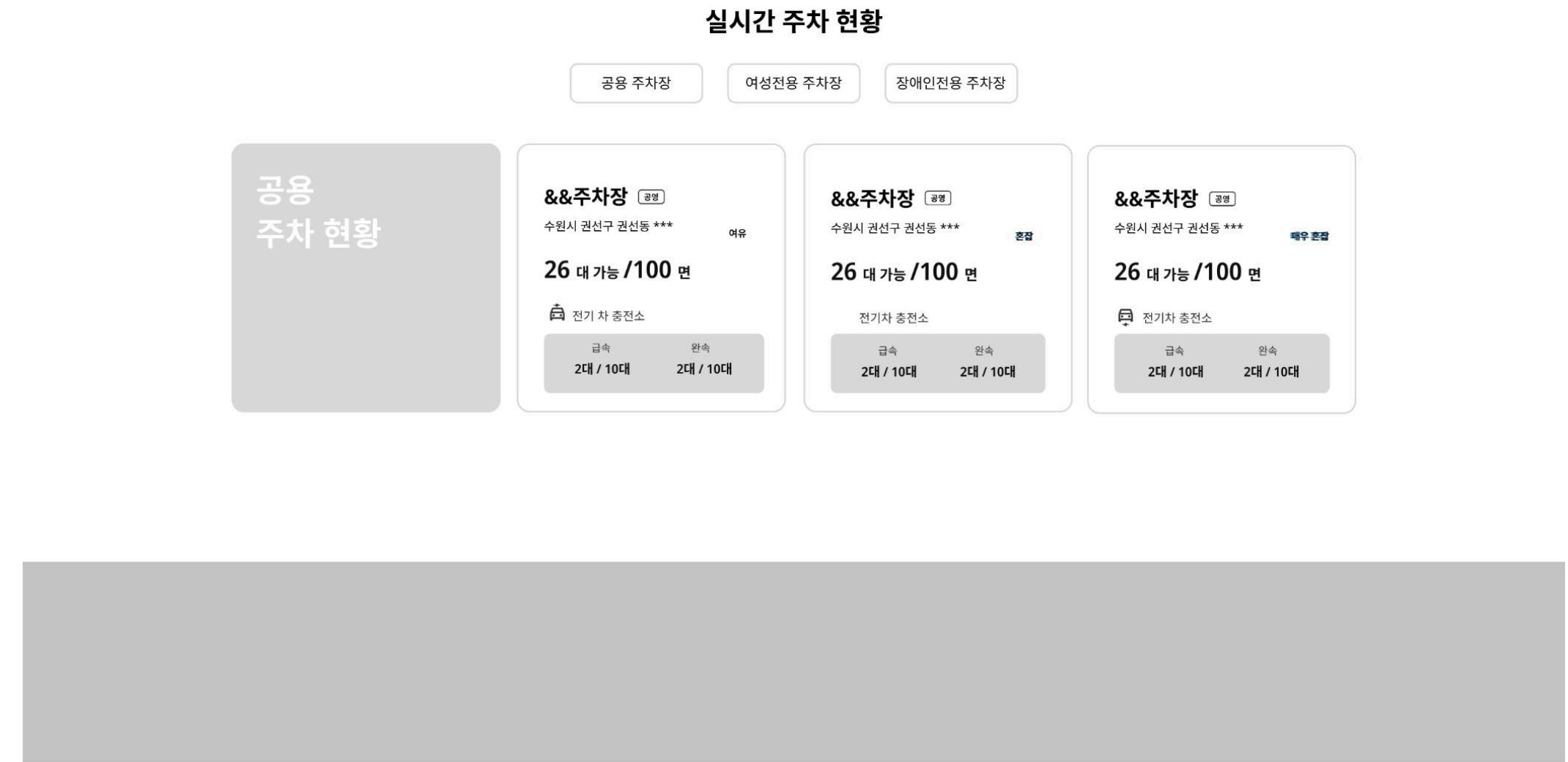
body_Noto Sans P Regular 13px

Wireframe

주차장
이용내역
정기권 신청/조회
공지사항

PARKET
파켓으로 편하게 충전하고 주차하세요!

search...
실시간 충전예약
정기권 신청



실시간 주차 현황

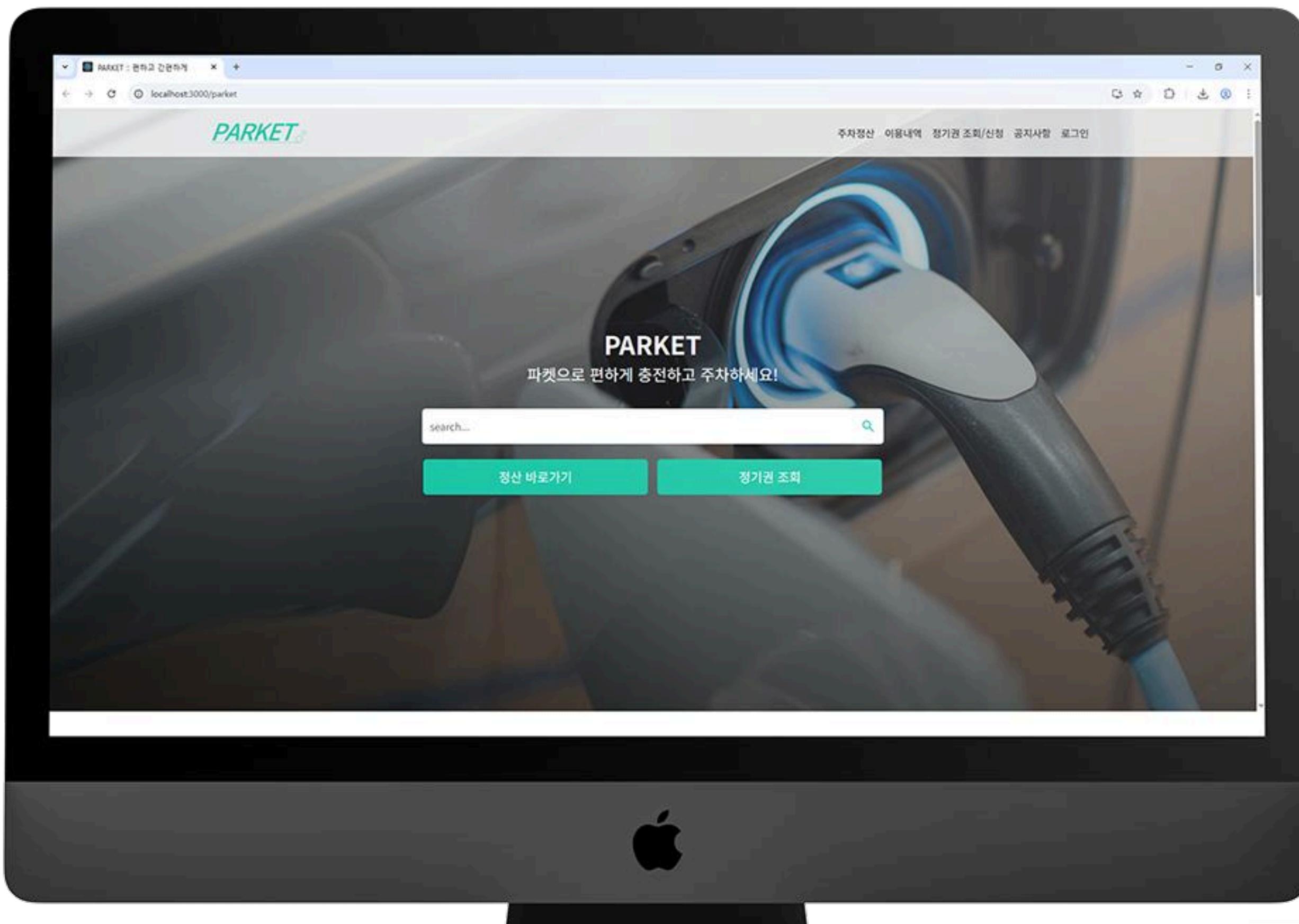
주차 요금 조회 / 정산
주차 예상 요금 조회

지역선택
주차장선택
입차일시
출차일시
차량번호
조회하기
정산하기

전기차 충전 여부
충전하시는 분만 체크해주세요

공지사항 자주 묻는 질문
+
전화문의
파켓 대표번호 031-1234-5678
주차 및 정산 통합 쿨센터 031-3214-8797
정기권 신청 및 환불 문의 031-3214-8797

Prototype



주차 요금 조회

주차 요금 조회 / 정산

지역선택: 지역을 선택해주세요!

주차장선택: 주차장을 선택해주세요!

차량번호:
전기차 충전 여부: *충전하시는 분만 체크해주세요

입차일시:
출차일시:

조회하기
정산하기

**토스페이로 정기권 첫 결제시
2000 포인트 지급!**

2025.06.25 ~ 2025.07.14



PARKET · toss

실시간 주차 현황

공용 주차장 여성전용 주차장 장애인전용 주차장

공용 주차 현황	
<div style="display: flex; align-items: center; justify-content: space-between;"> <div style="flex: 1; border: 1px solid #00aaff; padding: 5px; border-radius: 10px; text-align: center;"> 교동 공영 <small>공영</small> 경기도 수원시 팔달구 교동 139 84대 가능 / 246면  <small>혼잡</small> </div> <div style="flex: 1; border: 1px solid #00aaff; padding: 5px; border-radius: 10px; text-align: center;"> 화서역 환승 <small>공영</small> 경기도 수원시 팔달구 화서동 644-1 174대 가능 / 290면  <small>여유</small> </div> <div style="flex: 1; border: 1px solid #00aaff; padding: 5px; border-radius: 10px; text-align: center;"> 성대 환승(구) <small>공영</small> 경기도 수원시 장안구 율전동 529 40대 가능 / 160면  <small>혼잡</small> </div> </div>	
<div style="display: flex; align-items: center; justify-content: space-between;"> <div style="flex: 1; border: 1px solid #00aaff; padding: 5px; border-radius: 10px; text-align: center;"> <small>전기차 충전소</small> 급속 6대 / 10대 완속 6대 / 10대 </div> <div style="flex: 1; border: 1px solid #00aaff; padding: 5px; border-radius: 10px; text-align: center;"> <small>전기차 충전소</small> 급속 6대 / 10대 완속 10대 / 10대 </div> <div style="flex: 1; border: 1px solid #00aaff; padding: 5px; border-radius: 10px; text-align: center;"> <small>전기차 충전소</small> 급속 0대 / 10대 완속 2대 / 10대 </div> </div>	

충전과 주차

이젠 **PARKET PAY**로 똑똑하게!

충전하러가기

공지사항

자주 묻는 질문

제목	작성일
환불 기간 및 수수료 관련 안내	2025-06-24
정기권 결제 포인트 사용법	2025-06-18
포인트 환불 관련	2025-05-22
조회 오류 점검	2025-05-19

더보기 >

전화문의

파켓 대표전화 031-123-4567

통합 콜센터 031-324-4632

정기권 문의 031-324-4557

주차 및 정산 문의 031-324-2134

kakao 카톡문의

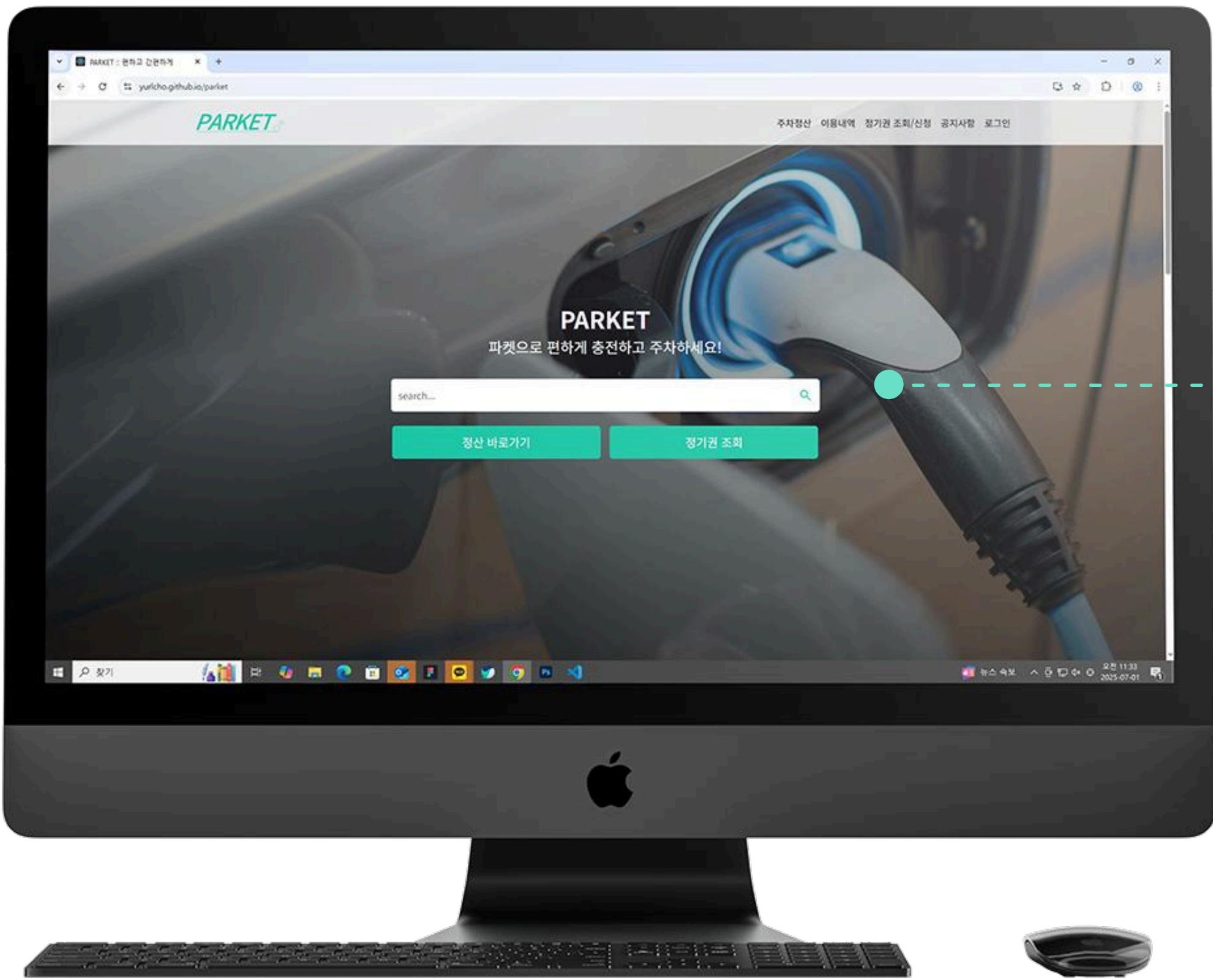
카카오톡 : Parket

INSTAGRAM @parket_kr

네이버 블로그 parket

YOUTUBE PARKET

main_Hero Section



브랜드 핵심 메시지와 시각적 이미지 배치로 첫 화면에서 PARKET을 각인시킬 수 있도록 구성하였습니다.

정산 기능과 정기권 기능은 페르소나를 기반하여 니즈를 반영한 핵심 서비스로, CTA 버튼과 함께 직관적으로 배치하여 사용자의 관심과 행동이 극대화되도록 구현하였습니다.

```
<h1>PARKET</h1>
<h4>파켓으로 편하게 충전하고 주차하세요!</h4>
<InputGroup className="mb-4">
  <Form.Control
    placeholder="search..."
    aria-label="Recipient's username"
    aria-describedby="basic-addon2"
  />
  <Button variant="outline-secondary" id="button-addon2">
    <span className="material-symbols-outlined">
      search
    </span>
  </Button>
</InputGroup>

<Button className='MBTN' variant="outline-primary">정산 바로가기</Button>
<Button className='MBTN' variant="outline-primary">정기권 조회 </Button>
```

main_DisableIfEmpty

주차 요금 조회

주차 요금 조회 / 정산

주차장선택: 주차장을 선택해주세요!

차량번호: 120가 3456

입차일시: 2025년 6월 26일 오전 8시 23분

이용 시간: 5일 2시간 8분

조회하기

정산하기

주차 예상 요금 조회

토스페이로 정기권 첫 결제시
2000 포인트 지급!
2025.06.25 ~ 2025.07.14

주차 요금 조회

주차 요금 조회 / 정산

지역선택: 경기도 수원시 장안구

주차장선택: 성대 환승

차량번호: 120가 3456

입차일시: 2025년 6월 26일 오전 8시 23분

이용 시간: 5일 2시간 8분

조회하기

정산하기

주차 예상 요금 조회

토스페이로 정기권 첫 결제시
2000 포인트 지급!
2025.06.25 ~ 2025.07.14



지역·주차장이 선택되어 있지 않은 상태에서 조회 버튼을 클릭하면 alert 창으로 ‘먼저 지역을 선택해주세요!’로 경고창이 뜨도록 했습니다.

지역·주차장 선택 후 탭 변경 시 입력·선택 내용이 초기화 및 불활성화 되게 구현하였습니다.

```
// 지역 선택하면 주차장선택 가능하게 하고 선택 목록 생성
const bringParkingLotsOnThisArea = area => {
  setSelectedParkinglot('');
  setSelectedEnterHour('');
  setSelectedEnterMinute('');
  setSelectedExitHour('');
  setSelectedExitMinute('');
  setResultCost('');
  setSelectedArea(area);

  // 검색 내용이 비어있으면 검색 초기화하기
  setDisableParkinglot(setDisableIfEmpty(area));
  setDisableSelectTime(true);

  if (area!="") {
    setParkingLots(
      lpData.filter(data => data.address.indexOf(area) >= 0)
    );
  }
}

turn (
<Form className='paymentForm' onLoad={() => bringParkinglotsOnThisArea('')}>
  <FormGroup className='selectLong'>
    <Form.Label className='paymentLabel'>지역선택</Form.Label>
    <Form.Select
      onChange={(event) => bringParkinglotsonThisArea(event.target.value)}
      value={selectedArea}>
      <option value="">지역을 선택해주세요!</option>
      {locationOptions.map((area, i) =>
        <option value={area} key={i}>{area}</option>
      )}
    </Form.Select>
  </FormGroup>
  <FormGroup className='selectLong'>
    <Form.Label className='paymentLabel' defaultValue="">주차장선택</Form.Label>
    <Form.Select
      disabled={disableParkinglot}
      value={selectedParkinglot}
      onChange={(event) => {
        setSelectedParkinglot(event.target.value);
        setDisableSelectTime(setDisableIfEmpty(event.target.value));
        setSelectedEnterHour('');
        setSelectedEnterMinute('');
        setSelectedExitHour('');
        setSelectedExitMinute('');
        setResultCost('');
      }}
    >
      <option value="" checked>주차장을 선택해주세요!</option>
      {parkingLots.map((lot, i) =>
        <option value={lot.title} key={i}>{lot.title}</option>
      )}
    </Form.Select>
  </FormGroup>

```

main_Key Features

*주차 요금 비용 산출 과정

1. 공통: 지역, 주차장 선택

```
// 주차요금 조회 탭에서 "조회하기" 버튼 누르면 조회 메세지 표시
const checkPayment = action => {
  let resultMinute = 0; // 산출된 주차 시간 (분)
  let toShowResult = false; // 결과를 보여주려면 true

  // 지역 선택이 안 되어있다면
  if (selectedArea == "") {
    alert("먼저 지역을 선택해주세요!");
  }

  // 주차장 선택이 안 되어있다면
  } else if (selectedParkinglot=="") {
    alert("주차장을 선택해주세요!");
  }
```

2. 주차 요금 정산: 차량 번호 입력 입차 여부 확인

```
// 차량 번호 입력을 잘못했다면
} else if (!carNumRE.test(writtenCarPlate)) {
  alert(
    "차량 번호를 다시 입력해주세요!\n"
    + "차량 번호 입력 예: "
    + "'12가 3456' 혹은 '789황 6543'"
  );
  setSelectedEnterTime('');
  setSelectedExitTime('');

// 입력에 문제가 없다면 parkingPaymentData 와 비교
// 입력에 문제가 없다면 parkingPaymentData 와 비교
} else {
  const oneCar = parkingPaymentData.filter(oneCar => oneCar.carPlate == writtenCarPlate);
  // 해당 차량의 입차 기록이 없다면 (parkingPaymentData에서 carplate 없을 때)
  if (oneCar.length < 1 || selectedParkinglot != oneCar[0].parkinglotId) {
    alert("해당 차량, " + writtenCarPlate + "은 입차하지 않았습니다.\n차량 번호를 다시 확인해주세요.")
    setSelectedEnterTime('');
    setSelectedExitTime('');

  } else {
    toShowResult = true;
    // 입차일시 입력력
    const enterTime = new Date(oneCar[0].enterTime);
    setSelectedEnterTime(setDateTimeForm(enterTime));

    // 출차일시 입력력
    const currentTime = new Date();
    setSelectedExitTime(setDateTimeForm(currentTime));

    resultMinute = Math.floor((currentTime - enterTime)/60000);
  }
}
```

3. 주차 요금 정산 입출차 시간 입력

```
const isExitTimeBeforeEnterTime = (enterHour, enterMinute, exitHour, exitMinute) => {
  let isStrangeTime = false;
  enterHour = Number(enterHour);
  enterMinute = Number(enterMinute);
  exitHour = Number(exitHour);
  exitMinute = Number(exitMinute);

  if (exitHour < enterHour) {
    isStrangeTime = true;
  } else if (exitHour == enterHour && exitMinute < enterMinute) {
    isStrangeTime = true;
  }

  return isStrangeTime;
}
```

```
// 주차 예상 요금 조회 탭에서
} else {
  if (selectedEnterHour == '' || selectedEnterMinute == '') {
    alert('입차 시간을 입력해주세요!');
  }

  } else if (selectedExitHour == '' || selectedExitMinute == '') {
    alert('출차 시간을 입력해주세요!');
  }

  } else if (isExitTimeBeforeEnterTime(
    selectedEnterHour,
    selectedEnterMinute,
    selectedExitHour,
    selectedExitMinute
  )) {
    alert('입차 시간이 출차 시간보다 늦습니다. 다시 입력해주세요.')
  }

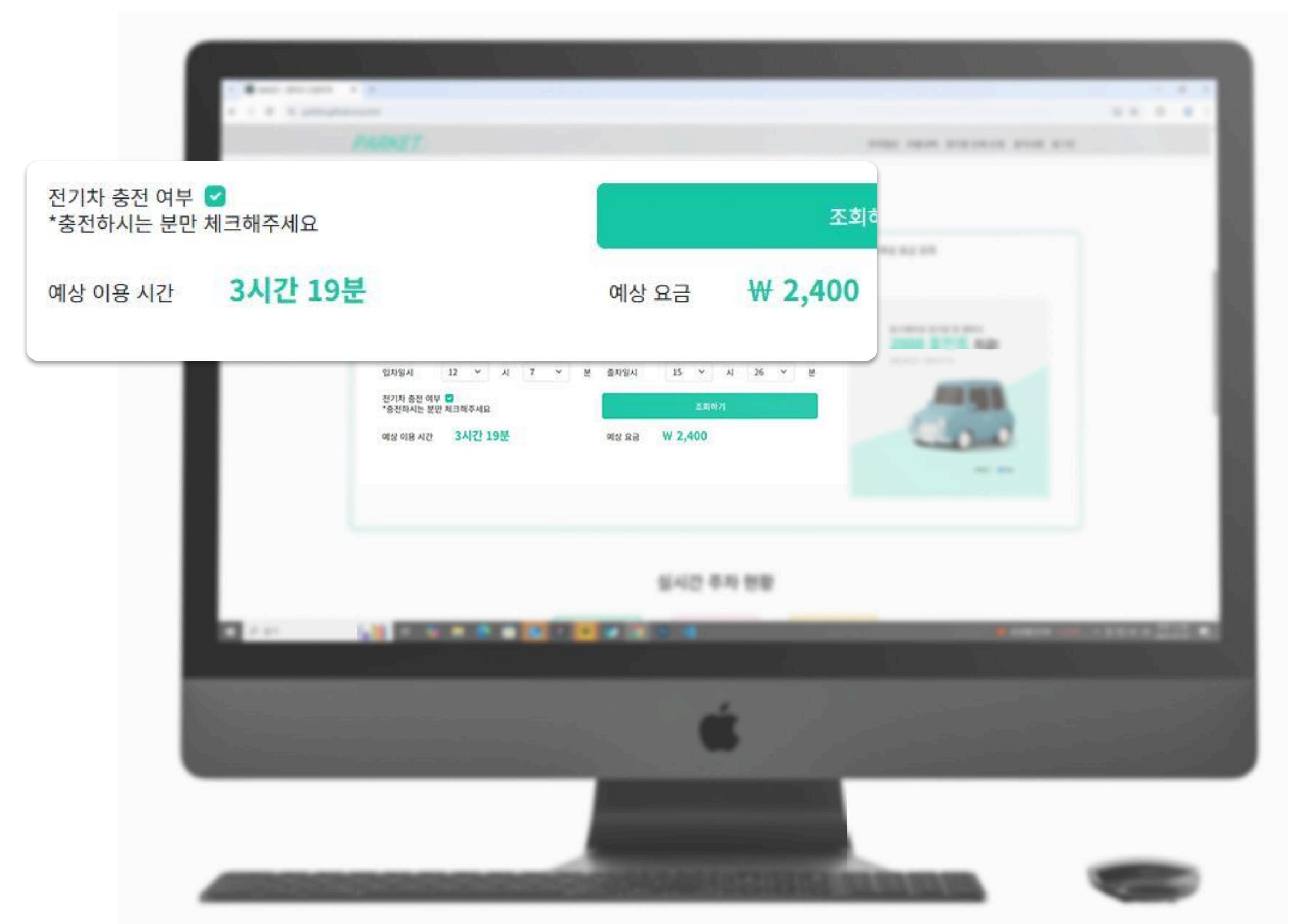
  } else {
    toShowResult = true;
    let hourgap = (Number(selectedExitHour) - Number(selectedEnterHour)) * 60;
    let minutegap = Number(selectedExitMinute) - Number(selectedEnterMinute);
    resultMinute = hourgap + minutegap;
  }
}
```

main_Key Features

*주차 요금 비용 산출 예시

in Component

```
{  
  resultCost != '' ?  
    <div id="paymentResult">  
      <p className='inputShortLeft'>  
        이용 시간 <span> {setResultTimeForm(resultTime)}</span>  
      </p>  
  
      <p className='inputShortRight'>  
        예상 요금 <span> {resultCost}</span>  
      </p>  
    </div> : null  
}
```



in Function

```
// 결과 값이 있다면 표시  
if (toShowResult) {  
  setResultTime(resultMinute);  
  setResultCost(getResultCost(resultMinute, usingCharger));  
} else {  
  setResultTime('')  
  setResultCost('');  
}
```

main_Auto Swiper

```
Lpdata.map((data, i) => {
  const { car, seat } = changeParking(data);

  return (
    <SwiperSlide className="LpSlide" key={i} style={{ border: `2px solid ${derColor}` }}>
      <div className="LpSlideBox">
        <div className="LpSlideFont">
          <h3>{data.title}</h3>
          <div className="parking_spot">{data.parking_spot}</div>
          <p>{data.address}</p>

          <div className="congestion">
            <img src={data.congestion.src} alt={data.congestion.name} />
            <p>{data.congestion_font}</p>
          </div>
        </div>

        <h3 style={{ color: fontColor }}>
          {car}<span>대 가능</span> / {seat}<span>면</span>
        </h3>

        <div className="electricBigBox">
          <div className="electric2">
            <img src={process.env.PUBLIC_URL + "/img/전기차.png"} alt="전기차" />
            <p>전기차 충전소</p>
          </div>

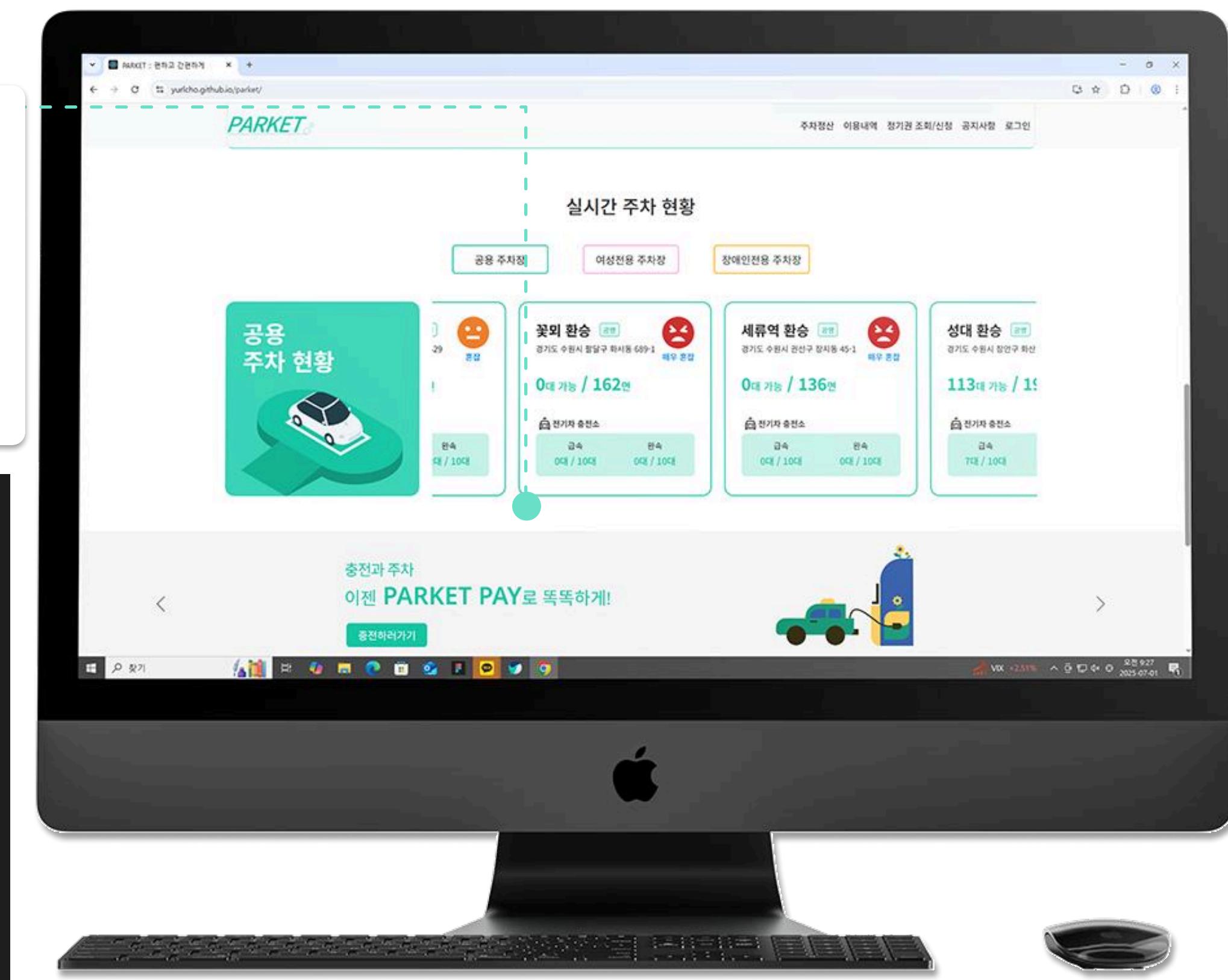
          <div className="electricBox" style={{ backgroundColor: bgColor2 }}>
            <div className="electricfont">
              <p>금속</p>
              <h5 style={{ color: fontColor }}>{data.electric1}대 / 10대</h5>
            </div>

            <div className="electricfont">
              <p>원속</p>
              <h5 style={{ color: fontColor }}>{data.electric2}대 / 10대</h5>
            </div>
          </div>
        </div>
      </div>
    </SwiperSlide>
  );
}

let LpData = [
  {
    id: 0,
    title: '화서역 환승',
    parking_spot: '공영',
    address: '경기도 수원시 팔달구 화서동 644-1',
    parking_car: 174,
    parking_seat: 290,
    woman_car: 20,
    woman_seat: 20,
    abbreviation_car: 20,
    abbreviation_seat: 20,
    electric1: 6,
    electric2: 10,
    congestion: { src: process.env.PUBLIC_URL + "/img/여유.png", name: "밀집도" },
    congestion_font: '여유',
    //이용내역
    num: 1,
    date: '2025-00-00',
    amount: 10000,
    orderNum: 2025000020250001,
    pay: '신용카드'
  },
]
```

실시간 주차 현황 데이터를 JavaScript 객체 배열로 구성하고, 이를 React에서 동적으로 슬라이드 컴포넌트(Swiper 등)에 매핑하여 각 주차장의 현재 상태 (예: 차량 수, 빈자리, 위치 등)를 시각적으로 확인할 수 있도록 구현했습니다.

```
let LpData = [
  {
    id: 0,
    title: '화서역 환승',
    parking_spot: '공영',
    address: '경기도 수원시 팔달구 화서동 644-1',
    parking_car: 174,
    parking_seat: 290,
    woman_car: 20,
    woman_seat: 20,
    abbreviation_car: 20,
    abbreviation_seat: 20,
    electric1: 6,
    electric2: 10,
    congestion: { src: process.env.PUBLIC_URL + "/img/여유.png", name: "밀집도" },
    congestion_font: '여유',
    //이용내역
    num: 1,
    date: '2025-00-00',
    amount: 10000,
    orderNum: 2025000020250001,
    pay: '신용카드'
  },
]
```



main_onClick Event



```
const LiveParking = () => {
  const [lpData, setLPData] = useState(lpData);
  const [selectedType, setSelectedType] = useState('공용');

  const changeType = () => {
    if (selectedType === '공용') {
      return {
        image: process.env.PUBLIC_URL + '/img/공용.png',
        bgColor: '#3905B8',
        textColor: '#14C6A4',
        fontColor: '#14C6A4',
        bgColor2: '#C4F2E8',
      };
    }

    if (selectedType === '여성전용') {
      return {
        image: process.env.PUBLIC_URL + '/img/여성전용.png',
        bgColor: '#FFA6DF',
        textColor: '#FFA6DF',
        fontColor: '#FA70C9',
        bgColor2: '#FFE6F6',
      };
    }

    if (selectedType === '장애인전용') {
      return {
        image: process.env.PUBLIC_URL + '/img/장애인전용.png',
        bgColor: '#FFCD58',
        textColor: '#FB301',
        fontColor: '#FFB301',
        bgColor2: '#FFEDC4',
      };
    }
  };

  const { image, bgColor, textColor, fontColor, bgColor2 } = changeType();
}
```

```
const changeParking = (data) => {
  if (selectedType === '공용') {
    return {
      car: data.parking_car,
      seat: data.parking_seat
    };
  }

  if (selectedType === '여성전용') {
    return {
      car: data.woman_car,
      seat: data.woman_seat
    };
  }

  if (selectedType === '장애인전용') {
    return {
      car: data.abbreviation_car,
      seat: data.abbreviation_seat
    };
  }
}
```

버튼 클릭 이벤트를 통해 현재 선택된 주차장 유형(공용, 여성, 장애인)을 상태값으로 저장하고, 해당 상태에 따라 아래 3가지가 동적으로 변경됩니다:

- 주차장 데이터: 선택된 유형에 맞는 데이터만 필터링해서 렌더링
- 스타일: 각 주차장 유형에 맞는 배경 색상, 텍스트 색상 적용
- 이미지: 유형별로 지정된 대표 이미지를 동적으로 교체

React 상태 관리(useState 등)를 통해 UI와 데이터를 연동하여 구현했습니다

sub_DetailPage

```
<Table bordered className='usageDetailsTb'>
  <tbody>
    <tr>
      <th>총 결제 금액</th>
      <td colspan=4 style={{ borderRight: 'transparent' }}><strong>{detailItem?.amount?.toLocaleString() ?? '0'}원</strong></td>
    </tr>

    <tr>
      <th>총 할인 금액</th>
      <td>0원</td>
      <th>P포인트 총 사용액</th>
      <td style={{ borderRight: 'transparent' }}><span>0</span>원</td>
    </tr>

    <tr>
      <th>실 결제 금액</th>
      <td>{detailItem?.amount?.toLocaleString() ?? '0'}원</td>
      <th>P포인트 총 적립액</th>
      <td style={{ borderRight: 'transparent' }}><span>Math.floor(detailItem.amount * 0.1).toLocaleString() ?? '0'</span>원</td>
    </tr>

    <tr>
      <th>결제하신 금액</th>
      <td colspan=4 style={{ borderRight: 'transparent' }}><strong>{detailItem?.pay ?><strong>{detailItem?.amount?.toLocaleString() ?? '0'}원</strong></td>
    </tr>
  </tbody>
```



```

  {
    usageData
      .slice()
      .sort((a, b) => b.num - a.num)
      .map((item) =>

        <tr key={item.num}>
          <td style={{ color: '#333' }}>
            {item.num}
          </td>

          <td onClick={() => { navigate('/usage/details/' + item.num) }}>
            {item.title} 주차장
          </td>

          <td style={{ color: '#333' }}>
            {item.date}
          </td>

          <td>
            {item.amount?.toLocaleString() ?? '0'}원
          </td>
        </tr>
      )
  }
</tbody>
```

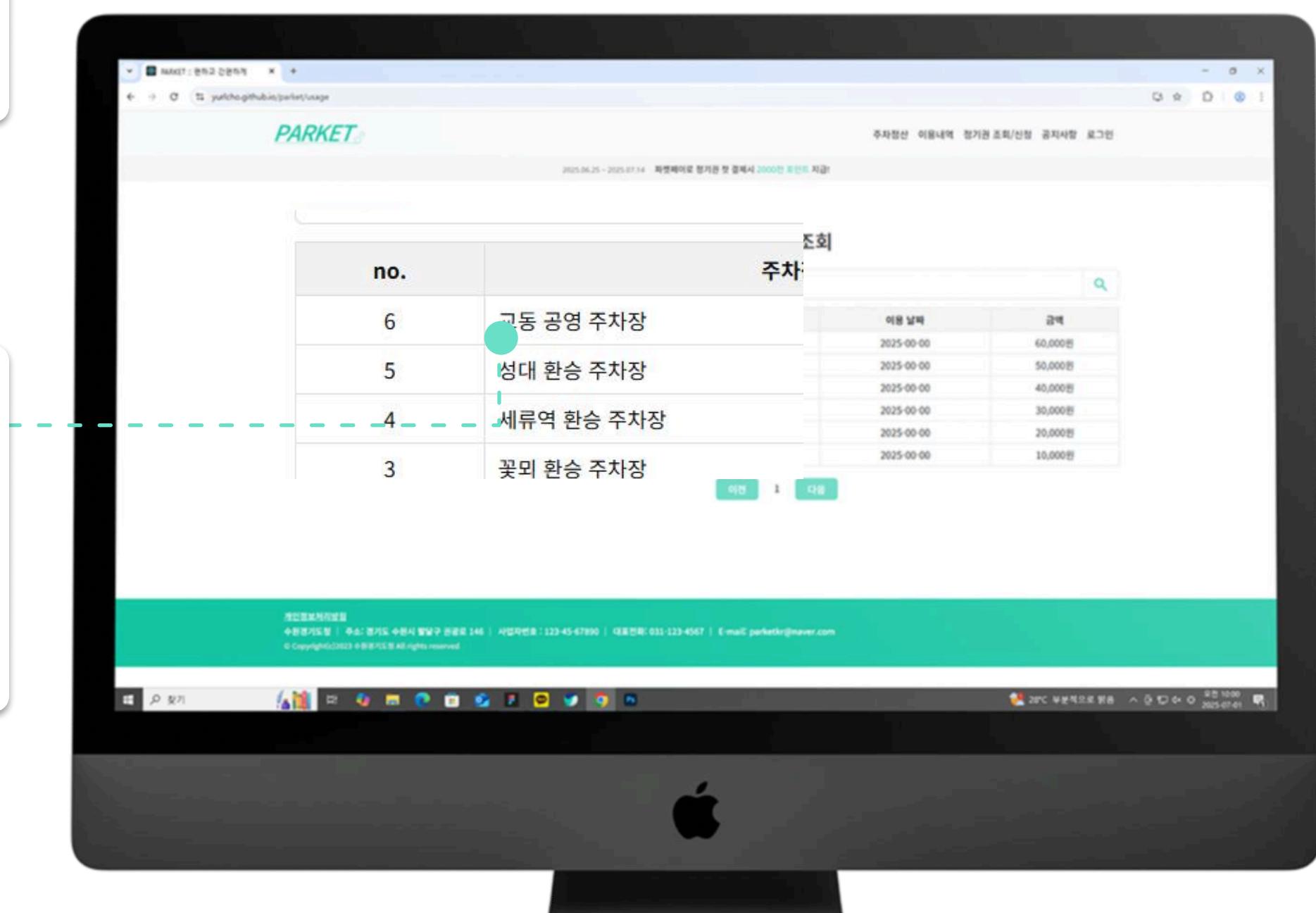
*상세내역 조회

이용내역 조회에서 각 주차장을 클릭하면 해당 항목의 상세 정보를 볼 수 있도록 구현했습니다. 이용내역 조회에서 항목을 선택하면 URL 파라미터를 통해 해당 데이터가 상세 페이지로 전달되고, 이에 맞는 상세 정보가 자동으로 렌더링됩니다.

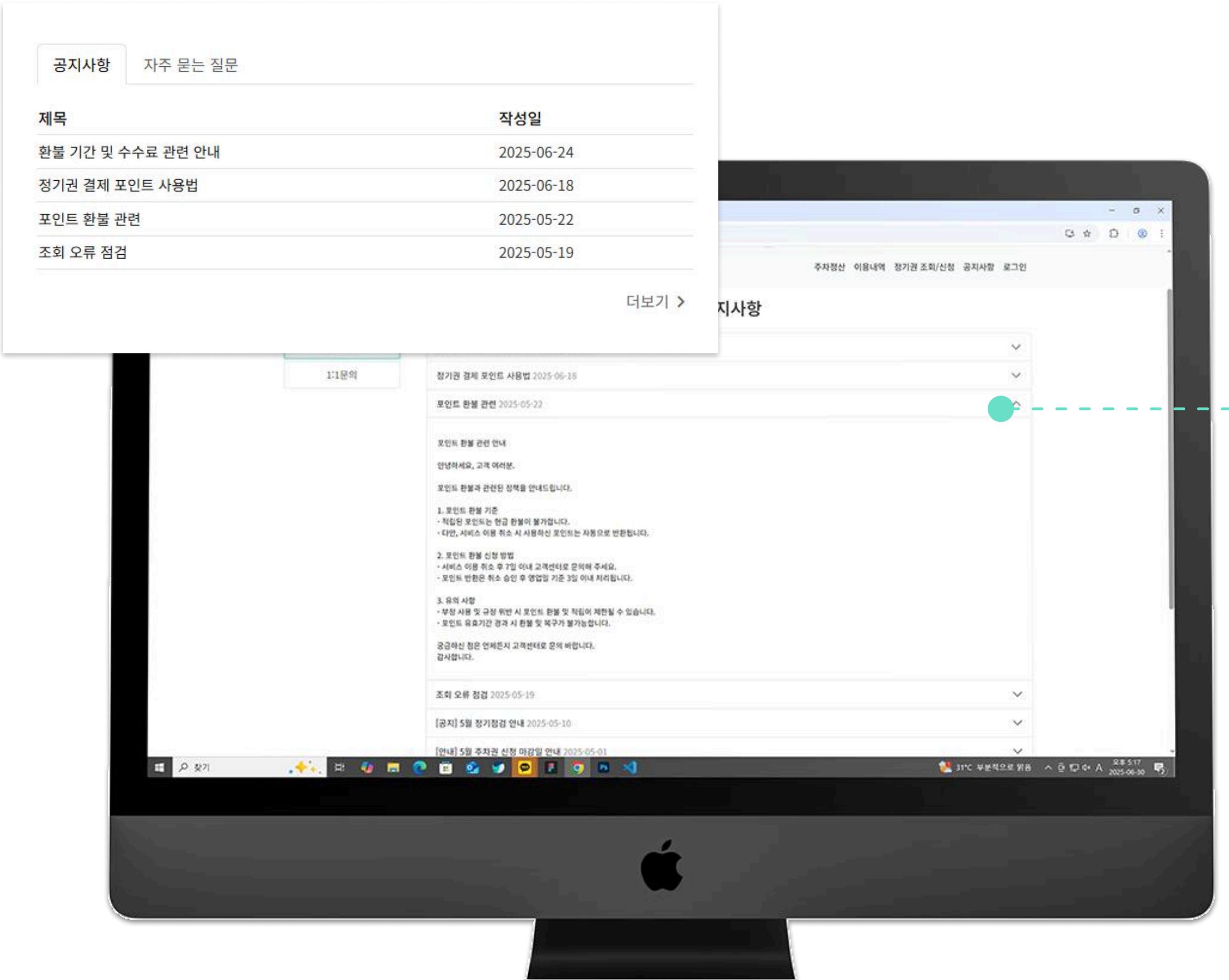
*이용내역 조회

Lp_data.js의 주차장 데이터를 기반으로, 항목의 번호, 제목, 이용날짜, 금액을 리스트 형태로 출력하는 이용내역 조회 페이지를 구현했습니다. 클릭 시 상세 페이지로 이동할 수 있도록 동적 라우팅도 연동했습니다.

The screenshot shows a summary table for usage details. It includes columns for total payment amount (60,000원), total discount (0원), total P-point usage (0원), and total P-point accumulation (6,000원). Below this is a detailed view for a specific transaction, showing a breakdown of payment amounts by parking lot (고동 공영 주차장, 성대 환승 주차장, 세류역 환승 주차장, 꽃뫼 환승 주차장) and their respective payment dates and amounts.



sub_onClick findIndex



메인 페이지의 공지사항에 있는 제목을 클릭하면 공지사항의 서브 페이지로 넘어갈 수 있도록 onClick 을 설정했습니다.
서브페이지로 넘어갔을 때 클릭한 공지사항의 상세 내용이 보이는 걸 원해 클릭한 공지사항의 아코디언이 열려 있도록 findIndex를 활용해 id 값이 같은 걸 찾도록 했습니다.

```
const Dtnotice = () => [
  const [page, setPage] = useState(1);
  const [lastPage, setLastPage] = useState(1);
  let [notice] = useState(noticeData);

  const location = useLocation();
  const openId = location.state?.openId || null;

  const openIndex = notice.findIndex(n => n.id === openId);

  useEffect(() => {
    if (notice.length) {
      setLastPage(Math.ceil(notice.length / 7));
    }
    if (openId !== null && openIndex !== -1) {
      const targetPage = Math.floor(openIndex / 7) + 1;
      setPage(targetPage);
    }
  }, [notice, openId]);

  const startIndex = (page - 1) * 7;
  const endIndex = page * 7;
  const currentPageNotices = notice.slice(startIndex, endIndex);

  const defaultOpenKey =
    openIndex >= startIndex && openIndex < endIndex
    ? (openIndex % 7).toString()
    : null;
}
```

```
/* 아코디언 */
<div className="noticedetail">
  <h2 className='noticename'>공지사항</h2>
  <Accordion defaultActiveKey={defaultOpenKey}>
    {notice.slice((page - 1) * 7, page * 7).map((item, i) => (
      <Accordion.Item eventKey={i.toString()} key={item.id}>
        <Accordion.Header><p>{item.title} <span style={{color: '#999'}}>{item.date}</span></p></Accordion.Header>
        <Accordion.Body>
          <pre>{item.content}</pre>
        </Accordion.Body>
      </Accordion.Item>
    ))}
  </Accordion>

```

sub_Search

정기권 신청

신청조회

토스페이로 정기권 첫 결제시
2000 포인트 지금!

2025.06.25 ~ 2025.07.14

정기권 조회

search...

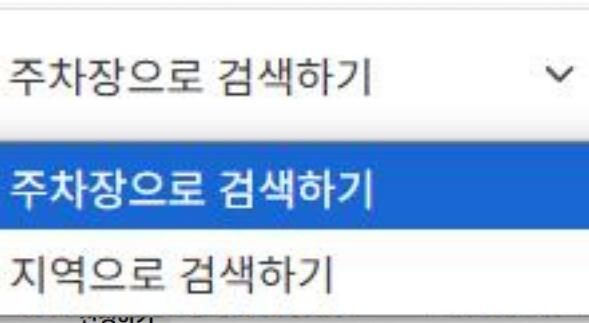
선택	지역	주차장명	종류	사용기간	상태	잔여수량	접수기간	교환하기
<input type="radio"/>	경기도 수원시 팔달구 화서동 644-1	화서역 환승	공영	2025-07-30 ~ 08-30	접수중	5	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	경기도 수원시 장안구 율전동 529	성대 환승(구)	공영	2025-07-30 ~ 08-30	마감	0	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	경기도 수원시 팔달구 화서동 689-1	꽃뫼 환승	공영	2025-07-30 ~ 08-30	접수중	16	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	경기도 수원시 권선구 장지동 45-1	세류역 환승	공영	2025-07-30 ~ 08-30	접수중	1	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	경기도 수원시 장안구 화산로 242	성대 환승	공영	2025-07-30 ~ 08-30	마감	0	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	경기도 수원시 교동 139	교동 공영	공영	2025-07-30 ~ 08-30	접수중	15	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	대전 중구 중앙로138번길 30	우리들 공원 주차장	민영	2025-07-30 ~ 08-30	접수중	5	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	대전 동구 중앙로200번길 70	중앙시장 공영 주차빌딩	공영	2025-07-30 ~ 08-30	접수중	4	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	대전 유성구 은천북로33번길 21-21 GT타워	GT주차타워	민영	2025-07-30 ~ 08-30	마감	0	2025-06-28 ~ 07-09	<button>신청하기</button>
<input type="radio"/>	대전 서구 둔산대로 155 대전시립미술관	대전시립미술관	공영	2025-07-30 ~ 08-30	접수중	9	2025-06-28 ~ 07-09	<button>신청하기</button>

이전 1 2 다음

필터별 검색 기능을 추가하여 사용자의 편의성을 고려하였으며,

사용자가 '주차장명' 또는 '지역명' 중 어떤 기준으로 검색할지를 선택하면,
해당 기준에 따라 실시간 필터링이 이루어집니다.

검색어 입력 시 페이지는 1페이지로 초기화되게 구현하였습니다.



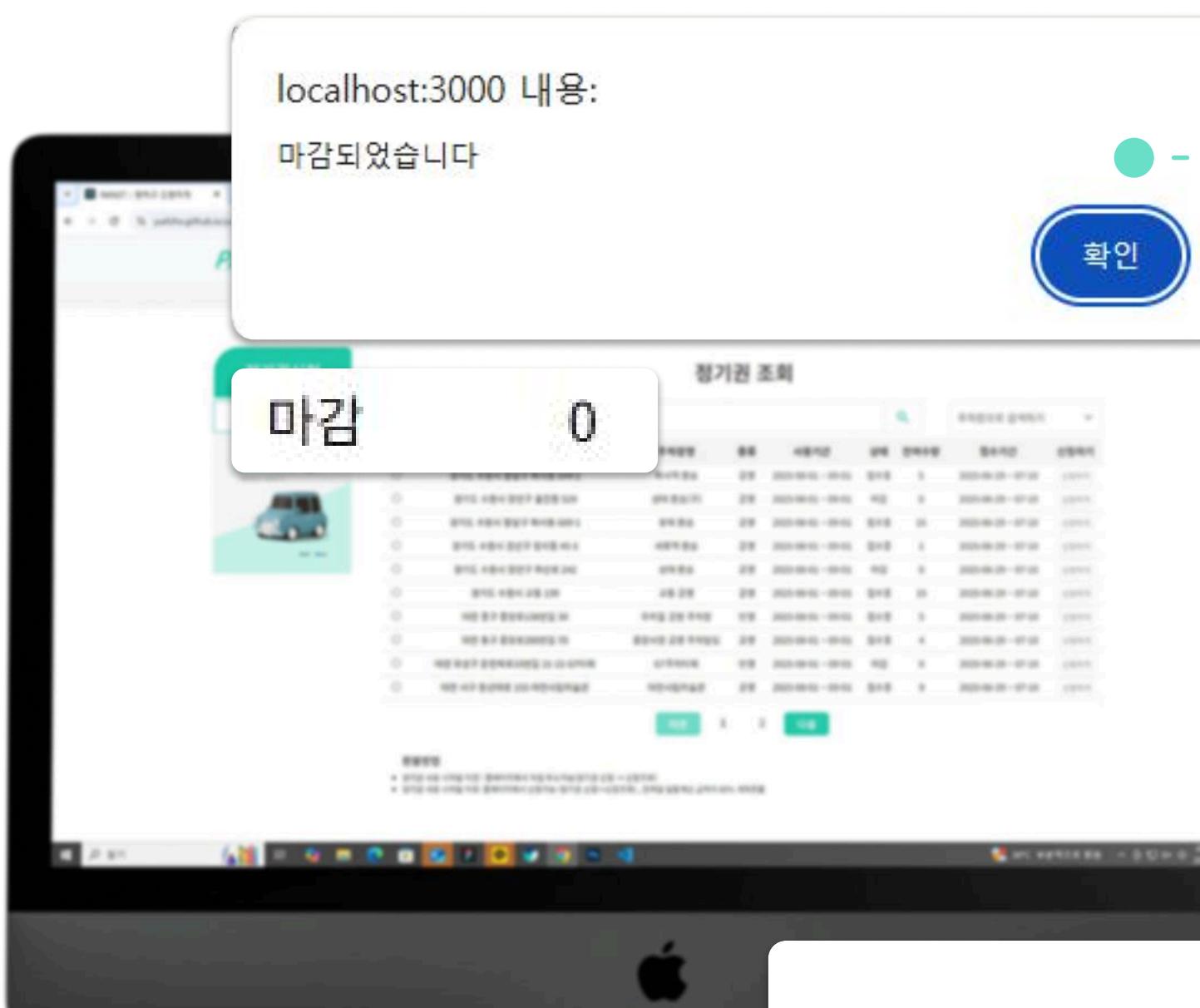
```
const { LpData2 } = props

const filteredData = (LpData2 || []).filter(product => {
  if (searchText === '') return true;

  if (searchType === 'address') {
    return product.address.toLowerCase().includes(searchText.toLowerCase());
  } else {
    return product.title.toLowerCase().includes(searchText.toLowerCase());
  }
}) //검색어, 검색 조건(title 또는 address)에 따라 데이터 필터링

<Form.Select
  aria-label="검색 유형 선택"
  className="custom-select"
  value={searchType}
  onChange={(e) => {
    setSearchType(e.target.value);
    setCurrentPage(1); // 검색 유형 변경 시 페이지 1로 초기화
  }}
>
  <option value="parking">주차장으로 검색하기</option>
  <option value="address">지역으로 검색하기</option>
</Form.Select>
```

sub_Remaining



잔여 수량에 따라 상태 표기를 다르게 렌더링하고
사용자가 '신청하기' 버튼을 클릭하면 경고 창이 출력되며,
데이터 상태에 따라 UI가 동적으로 반응하도록 구현하였습니다.

```
<tr
  key={index}
  className={`${
    checkedidx === index ? 'tractive' : ''
  } ${product.Remaining === 0 ? 'SoldOut' : ''}
`}>
  /* 마감된 주차장을 흐리게 보이는 CSS클래스 추가 */
  <td>
    <input className='IC'
      type="radio"
      name="parking"
      onChange={(e) => { setCheckedindx(index) }} />
  </td>
  <td>{product.Remaining === 0 ? '마감' : '접수중'}</td>
  /* 잔여수량 0일시에 마감 아니면 접수중으로 표시 */
  <td>{product.Remaining}</td>
  <td>{later} ~ {endlater}</td>
  <td><Button
    className="DBTN"
    style={{ height: '27px', width: '58px', fontSize: '12px' }}
    onClick={(e) => {
      e.currentTarget.blur(); // 포커스 제거
      if (product.Remaining === 0) {
        alert('마감되었습니다'); //마감된 주차장 신청시 경고창
      } else {
        // 신청 처리 로직
        console.log('정상 신청 진행');
      }
    }}>신청하기</Button></td>
```

sub_PageButton

정기권 조회

선택	차량	주차장명	종류	사용기간	상태	잔여주행	점유기간	선행하기
○	경기도 수원시 팔달구 화이동 644-1	화이트 환승	운행	2023-07-30 ~ 08-09	정수증	0	2023-08-28 ~ 07-09	선행하기
○	경기도 수원시 팔달구 팔달로 529	화이트 환승	운행	2023-07-30 ~ 08-09	정수증	0	2023-08-28 ~ 07-09	선행하기
○	경기도 수원시 팔달구 화이동 649-1	화이트 환승	운행	2023-07-30 ~ 08-09	정수증	0	2023-08-28 ~ 07-09	선행하기
○	경기도 수원시 팔달구 화이동 49-1	화이트 환승	운행	2023-07-30 ~ 08-09	정수증	1	2023-08-28 ~ 07-09	선행하기
○	경기도 수원시 팔달구 화산로 242	화이트 환승	운행	2023-07-30 ~ 08-09	정수증	0	2023-08-28 ~ 07-09	선행하기
○	경기도 수원시 팔달구 화정로 139	화정 운행	운행					
○	대한 풍구 운행제308번길 30	화정 운행 주차장	주차장					
○	대한 풍구 운행제308번길 70	화정시민 운행 주차장	주차장					
○	대한 풍구 운행제308번길 20-13 운행제	화정 운행제308번길	운행					
○	대한 풍구 운행제308번길 20-13 운행제	화정 운행제308번길	운행					

이전 1 2 다음

페이지네이션은 기능을 통해 사용자의 가독성을 향상시키고
대용량 데이터를 분할 처리하여 성능 최적화하며,
현재 페이지 변경 시 검색어 및 필터를 유지한 상태에서 탐색 가능하게 구현하였습니다.

```
// 페이지네이션 상태
const [currentPage, setCurrentPage] = useState(1);
const itemsPerPage = 10;
// 현재 페이지에 표시할 데이터 범위 계산
const indexOfLastItem = currentPage * itemsPerPage;
const indexOfFirstItem = indexOfLastItem - itemsPerPage;
const currentItems = filteredData.slice(indexOfFirstItem, indexOfLastItem);
// 전체 페이지 수 계산
const totalPages = Math.ceil(filteredData.length / itemsPerPage);
// 페이지 번호 배열 생성
const pageNumbers = [];
for (let i = 1; i <= totalPages; i++) {
  pageNumbers.push(i);
}

/* 페이지네이션 버튼 */
<Button
  className="PBTN"
  onClick={() => setCurrentPage(prev => Math.max(prev - 1, 1))}
  disabled={currentPage === 1}
  style={{ margin: '0 5px', minWidth: '35px' }}>이전</Button>

{pageNumbers.map(number => (
  <span
    key={number}
    className={` PBTN pagination-btr ${number === currentPage ? 'active' : ''}`}
    onClick={() => setCurrentPage(number)}
    style={{ color: number === currentPage ? '#c5c5c5' : '#f3f3f3',
      color: '#666' }}>
    {number}
  </span>))
  <Button
    className="PBTN"
    onClick={() => setCurrentPage(prev => Math.min(prev + 1, pageNumbers.length))}
    disabled={currentPage === pageNumbers.length}
    style={{ margin: '0 5px', minWidth: '35px' }}>다음</Button>
```

Team Project Review

잘한 점

가람:

나름 책임감 있게 잘 했다고 생각합니다.
그리고 내가 React를 잘하고 있는 건지 몰랐었는데 이번 팀 프로젝트를 통해서 잘은 아니지만 어느 정도 할 수 있다는 것을 알게 되었고, 내가 뭐가 부족한지도 알게 되었습니다.

어진:

React를 통해 Data 송수신 및 수정, 표현 등이 가능했고, Frontend 작업과 Backend 작업이 가능했다.

유리:

리액트를 조금 더 자유롭게 활용할 수 있게 되었습니다. 다른 팀원은 코드를 어떤 방식으로 쓰고 scss를 어떻게 적용하는지 코드를 보여 배울 수 있었습니다. 피드백을 유연하게 수용할 수 있는 방법을 조금은 터득한 것 같습니다.

수빈:

이번 프로젝트를 통해 그동안 어려움을 느꼈던 동적 데이터 처리 방식과 리액트의 컴포넌트 기반 구조에 대해 실제로 구현하며 이해할 수 있었습니다.
특히, 기존의 정적 HTML 기반 작업보다 React와 Bootstrap을 활용함으로써 더 빠르고 효율적으로 기능을 구현할 수 있다는 점을 익혔습니다.
또한 팀 프로젝트를 수행하면서, 기획, 역할 분담, 소통, 발표까지의 전 과정을 경험하며 팀워크와 협업의 중요성을 다시 한 번 깨달았습니다.

보완할 점

가람:

자바스크립트로 이벤트 주는 것을 더 해봐야 할 거 같고, React도 계속 해봐야 할 거 같습니다.
그리고 팀 프로젝트 경우 코드는 나만 보는 게 아니라 다른 팀원들도 보기 때문에 팀원들을 생각해서 좀 더 깔끔하게 작성하고 주석으로도 설명을 추가해야겠다고 생각했습니다.

어진:

화면 표시를 위해 처리한 방식은 사실 SQL로 쉽게 할 수 있고 추후 DB 연동까지 되면 사실상 필요 없어지는 code 였다

유리:

다른 팀원들을 생각해 코드를 조금 더 깔끔하게 작성해야 할 것 같다고 생각했습니다. 팀원의 피드백을 훨씬 더 유연하게 적용할 수 있도록 의사소통 능력을 키워야 할 것 같다고 생각했습니다.

수빈:

프로젝트 기간이 길지 않았던 만큼, 스마트폰 등 모바일 환경에 최적화된 반응형 웹사이트로 구현하지 못한 점이 가장 아쉬웠습니다.
또한, 아직 전기차 인프라가 충분히 활성화되지 않은 현실적인 문제로 인해 사용자들이 실질적으로 활용할 수 있는 다양한 기능을 웹사이트에 더 전문적으로 담아내지 못한 점도 아쉬움으로 남습니다.
향후 전기차 시장의 확대에 따라, 예약 시스템, 실시간 충전소 정보, 요금 비교 등 보다 다양한 기능을 추가해보고 싶습니다.

Thank you!

team project | 김가람 남어진 박수빈 최유리

☞ <https://jalgaram.github.io/parket/>