

인터넷 프로그래밍 프로젝트

명지대 택시 합승 구하기!

여럿이 타서 싸게 학교 가자

합승 후 정산은 확실하게 해주세요!
무임승차할 경우 법적 책임은 본인에게 있습니다.

게시글 작성하기 >

검색 필터 설정 >

출발지

목적지

날짜

출발 시간

현재 인원

윤현섭, 오윤택, 김대현, 문승관

목차

A table of contents

1 배경

2 주요 기능



1 배경

돈을 절약하자!

2시쯤 서울에서 학교가실분 없나요..?

택시나 카풀 구합니다

13:20 | 익명

1시 30분

기흥 5출 택시 타실분?

🔵 1 | 12:53 | 익명

12:45 기흥 택시팟 구해요

5출에서 타용 댓 남겨주세요

🔵 2 | 12:36 | 익명

12:35 기흥역 5출

택시타실 분 있나요

🔵 13 | 12:06 | 익명

12시 45분 기흥 5출 택시

같이 타실분 없나요?

🔵 8 | 11:56 | 익명

11시45분 기흥 5출 택시

꼭지주세요

11:18 | 익명

11시 45 기흥 택시

1공가는 택시 같이 타실분? 5출에서 탈 예정입니다

🔵 1 | 11:17 | 익명

Why?

1. 셔틀버스의 공급 부재로 많은 학생들이 택시를 이용
2. 합승을 구하는 과정이 번거롭다
3. 명지대 학생들이 간단하게 사용할 수 있는 플랫폼 필요

2 게시물 작성

명지대 택시 합승

합승에 필요한 정보를 입력 받아
서버에 전송

유효하지 데이터가 입력될 경우
에러 메시지 표시

서버가 입력받은 데이터를 **post**를 통해
데이터베이스에 기록

```
const mySchema = new mongoose.Schema({
  ride_month: String, //출발 월
  ride_day: String, //출발 일
  ride_hour: String, //출발 시
  ride_minute: String, //출발 분
  count: Number, //현재인원
  maxCount: Number, //최대탑승인원
  link: String, //채팅링크
  create: String, //게시글 생성일
  arrival: String, //출발지
  departure: String, //목적지
  password: String, //글 비밀번호
  studentNum: [] //학번
})
```

명지대 택시 합승 구하기!

여럿이 타서 싸게 학교 가자

합승 후 정산은 확실하게 해주세요!

무임승차할 경우 법적 책임은 본인에게 있습니다.

게시글 작성하기 >

월	-- 월 선택 --	▼	일	-- 일 선택 --	▼
출발 시간	-- 시간 선택 --	▼	분	-- 분 선택 --	▼
출발지	-- 출발지를 선택해주세요 --	▼	목적지	-- 목적지를 선택해주세요 --	▼
탑승 인원	-- 본인 포함 최대 탑승 인원 --				
오픈채팅 링크					
비밀번호			학번		

게시글 작성하기

```
app.post('/article', (req, res) => {
  const newArticle = postArticle(req.body)
  newArticle.save()
    .then(res.json({msg: "게시글 작성 완료"}))
    .catch((err)=>{res.json({msg: "에러가 발생했습니다"})})
});
```


2 게시물 필터 & 정렬

명지대 택시 합승

get을 통해 게시물 정보를 가져온다

검색필터를 통해 원하는 조건의
게시글만 볼 수 있는 **필터 기능**

게시글들을 탑승시간 순으로
정렬해서 표시

검색 필터 설정 >

출발지 선택	<input type="text" value="출발지 선택"/>	목적지 선택	<input type="text" value="목적지 선택"/>
월	<input type="text" value="월 선택"/>	일	<input type="text" value="일 선택"/>
시	<input type="text" value="시 선택"/>	분	<input type="text" value="분 선택"/>

출발지	목적지	날짜		출발 시간		현재 인원		
기흥역	명지대학교	6월	1일	9시	10분	1/4	신청	삭제
기흥역	명지대학교	6월	1일	10시	30분	2/4	신청	삭제
명지대학교	기흥역	6월	1일	14시	20분	1/4	신청	삭제
명지대학교	기흥역	6월	1일	17시	10분	1/4	신청	삭제
기흥역	명지대학교	6월	2일	8시	30분	2/4	신청	삭제

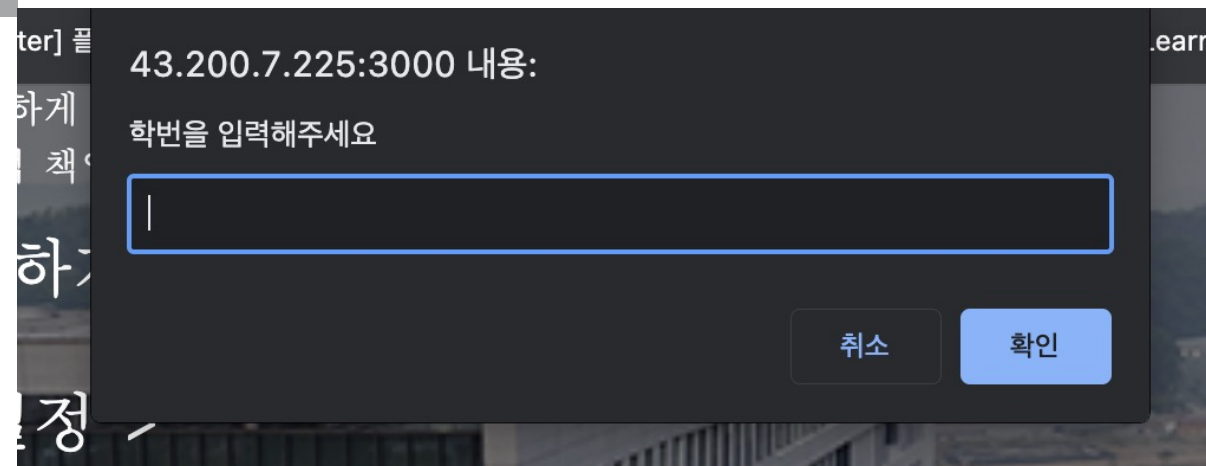
2 합승신청과 삭제

명지대 택시 합승

각 게시물마다 신청, 삭제 버튼을 통해
확인과 **삭제**

합승 신청시 학번을 입력 받아 **중복확인**

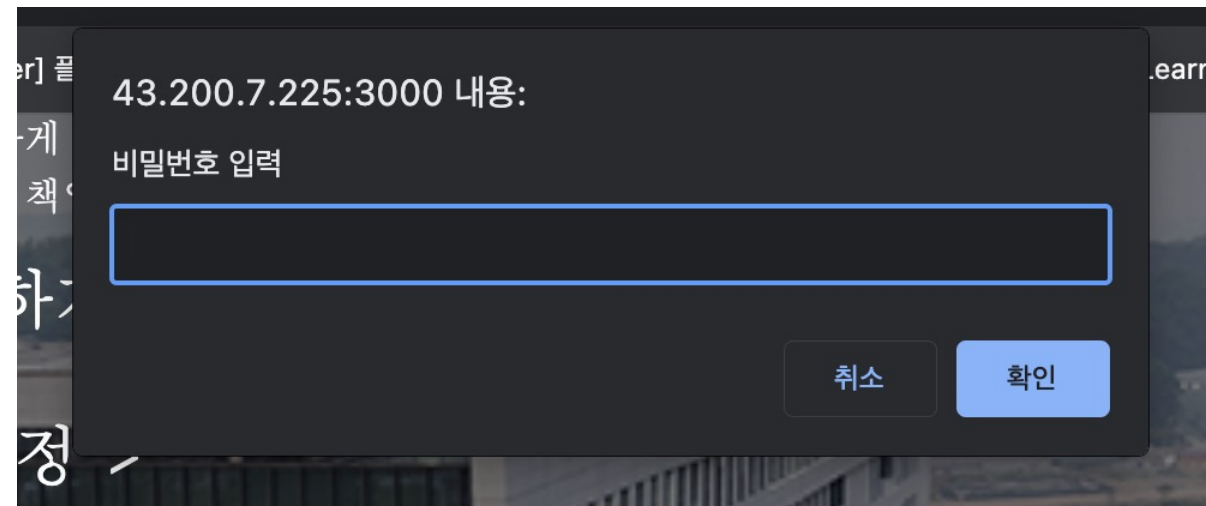
게시글 작성시 등록한 비밀번호를
확인하여 **삭제**



43.200.7.225:3000 내용:

학번을 입력해주세요

취소 확인



43.200.7.225:3000 내용:

비밀번호 입력

취소 확인

2 데이터 관리와 배포

명지대 택시 합승

5분 주기로 출발 시간이 지난
데이터를 확인하여
데이터베이스에서 삭제

aws ec2를 이용하여 온라인 배포

<http://43.200.7.225:3000/>
(현재는 접속이 불가능 합니다)



Amazon EC2

```
schedule.scheduleJob('0 0/5 * * *', async () => {
  const now = new Date();
  const now_year = now.toLocaleDateString('en-US', {
    year: 'numeric',});
  const now_month = now.toLocaleDateString('en-US', {
    month: '2-digit',});
  const now_day = now.toLocaleDateString('en-US', {
    day: '2-digit',});
  const now_hour = now.toLocaleDateString('en-US', {
    hour: '2-digit',});
  const now_minute = now.toLocaleDateString('en-US', {
    minute: '2-digit',});
  const article = await Article.find({})

  console.log(now);
  console.log(now_hour);
  console.log(now_minute);
  for(let i=0; i<article.length; i++){
    const id = article[i]['id']
    if(
      Number(article[i]['create'].substring(0, 4)) < Number(now_year) &&
      Number(article[i]['ride_month'].substring(0, article[i]['ride_month'].length - 1)) < Number(now_month) &&
      Number(article[i]['ride_day'].substring(0, article[i]['ride_day'].length - 1)) < Number(now_day) &&
      Number(article[i]['ride_hour'].substring(0, article[i]['ride_hour'].length - 1)) < Number(now_hour) &&
      Number(article[i]['ride_minute'].substring(0, article[i]['ride_minute'].length - 1)) < Number(now_minute)
    ){
      Article.deleteOne({'_id': id})
        .then(console.log("삭제완료"))
        .catch((err)=> console.log("오류가 발생했습니다"))
    }
  }
})
```