



오늘뭐해?

📅 기간	@2024/03/14 → 2024/07/11
📎 대표 이미지	
☰ 종류	배포 백엔드 프론트엔드
💬 종목	2학년 1학기 캡스톤
🔗 Git	https://github.com/gbsw-ERROR-404
🏆 수상	장려상

▼ 📅 개발 목적

학생들이 평소에는 학사 일정을 확인하기 위해 학교 홈페이지나 학교 종이 앱을 사용하여 힘들게 확인하였습니다. 시간표를 보기 위해서는 학교 홈페이지를 방문해야 하고, 급식 메뉴는 다른 앱에서 확인해야 했습니다. 따라서, 이러한 불편함을 해소하고자 "한 번에 모든 정보를 볼 수 있는 통합 사이트를 만들자"는 생각을 하게 되었습니다. 학사 일정 뿐만 아니라 생일과 같은 반 별 일정을 쉽게 등록하고 공유하여 반의 단합을 돕고, 반 별 추억을 간편하게 일기 형식으로 기록할 수 있습니다. 이를 위해 사용자가 일정을 등록하고 공유할 수 있는 기능을 구현하고, 반 별 일기를 작성하고 저장할 수 있는 환경을 제공할 것입니다. 사용자들은 각 반의 다양한 활동에 대해 효과적으로 소통하고, 소중한 순간들을 기록하고 공유할 수 있게 됩니다.

▼ 📅 기술 스택

Frontend : React

Backend : Node.js + express

Database : MySQL

Google/Github Login : Firebase

Cloud : Naver Cloud

Deploy : CloudType, Raspberry Pi - Ubuntu 22.04 LTS

▼ 구현 기능

- 학사 일정, 시간표, 급식 확인
- 반 별 일정 추가, 수정, 삭제
- 캘린더에 다이어리 기능으로 추억 남기기

 오늘 뭐해?

로그인

어서오세요!
저희 사이트에서 여러분들만의 캘린더를 만들어보세요!

캘린더로 이동



캘린더

학사 일정을 알려주고 원하는 날짜
에 문구를 남기고 사진을 넣을 수
있습니다.



시간표

우리 반 시간표를 보여주고 그 날
의 시간표를 왼쪽에 보여줍니다.



급식표

오늘의 급식을 보여주고 식단과 칼
로리를 보여줍니다.

 오늘 뭐해?

로그인

Login

이메일

비밀번호

회원가입

로그인




오늘 뭐해?
캘린더
급식표
시간표


어서오세요!
저희 사이트에서 여러분들만의 캘린더를 만들어보세요!

캘린더로 이동





캘린더

학사 일정을 알려주고 원하는 날짜에 문구를 남기고 사진을 넣을 수 있습니다.



시간표

우리 반 시간표를 보여주고 그 날의 시간표를 왼쪽에 보여줍니다.



급식표

오늘의 급식을 보여주고 식단과 칼로리를 보여줍니다.


오늘 뭐해?
캘린더
급식표
시간표
마이페이지
김도영님


4월의 일정들

학사일정

4.1 ~ 5
지방기능경기대회

4.2 ~ 4
중영어듣기

4.9 ~ 12
고영어듣기

개인일정

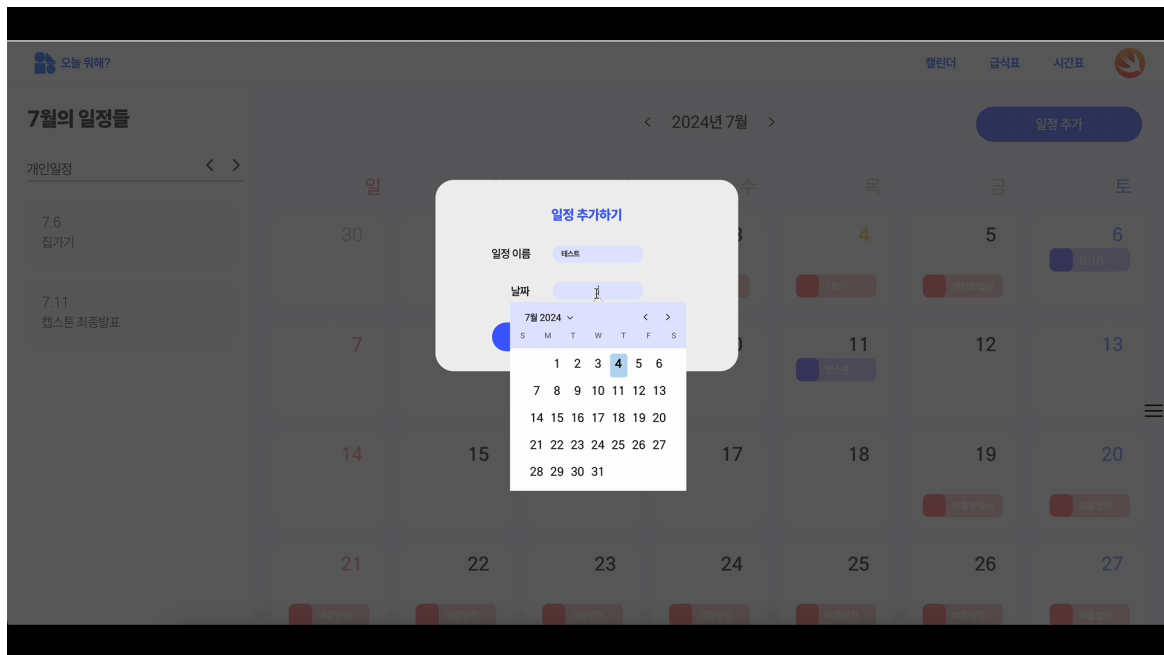
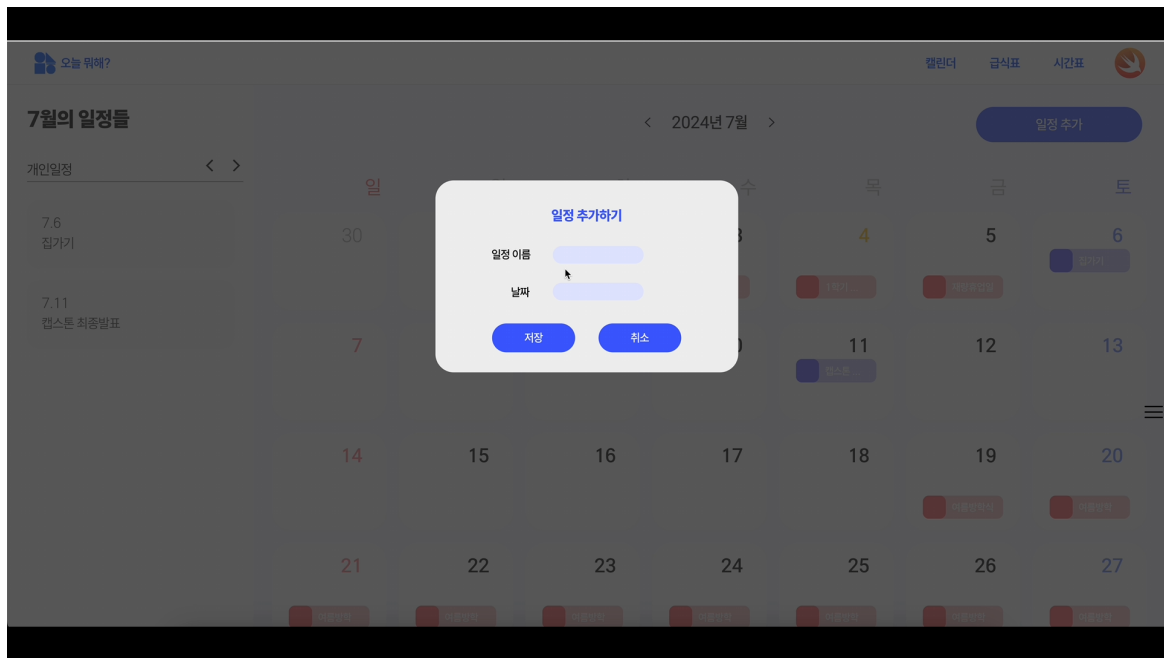
4.2 ~ 4
캡스톤 개발 발표

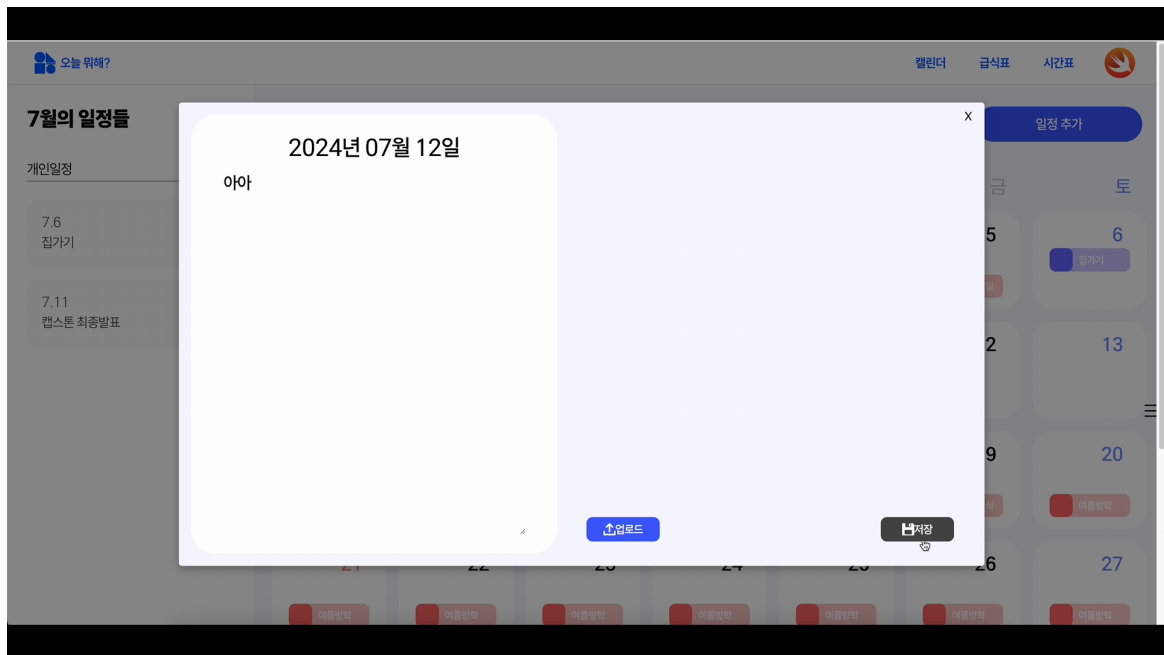
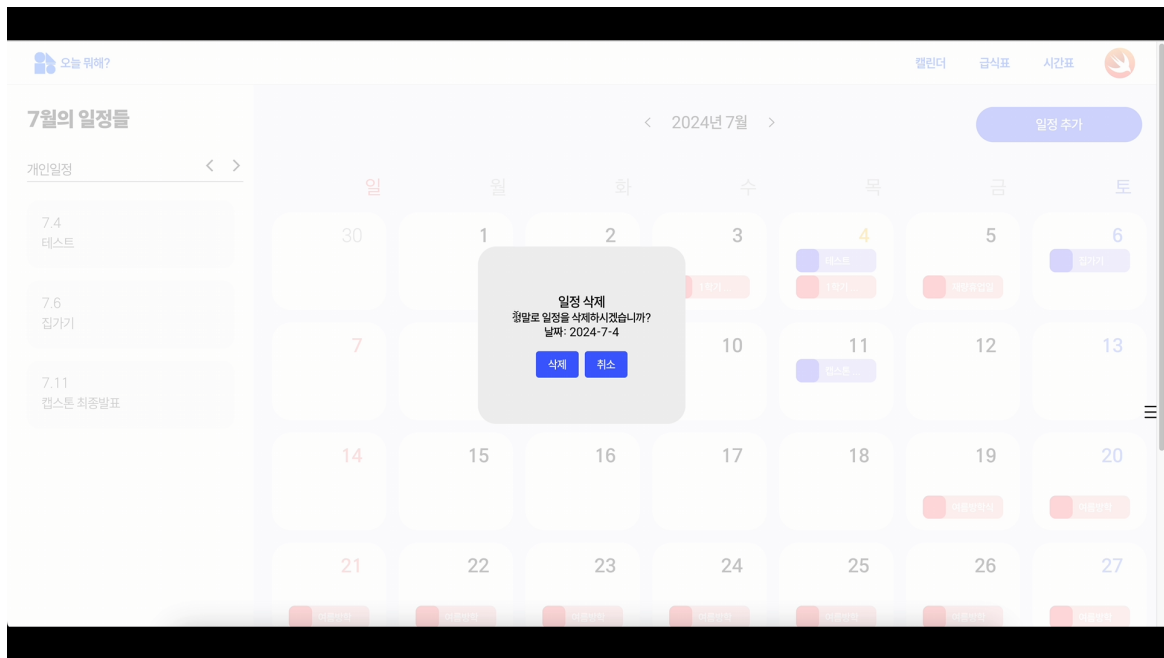
4.2
승환이 생일

< 2024년 4월 >

일정 추가

일	월	화	수	목	금	토
31	1 지방기능...	2 승환이 생일 캡스톤 개...	3 지방기능... 중영어듣기	4 지방기능... 캡스톤 개...	5 지방기능...	6
7	8	9 고영어듣기	10 국회 의원 고영어듣기	11 고영어듣기	12 고영어듣기	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

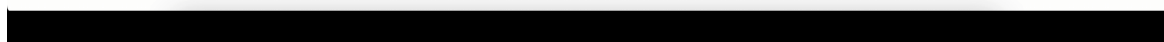




< 2024년 7월 4일 목요일 >

아침	점심	저녁
<p>쌀밥</p> <p>훈제오리김치볶음 (5.9.13)</p> <p>감자쇠고기국 (5.6.16)</p> <p>문어맛 방울오징어 조림 (1.5.6.13.17)</p> <p>배추김치 (9)</p> <p>단백질바 (2.5)</p> <p>식물성요구르트 (2)</p>	<p>*낙지콩나물비빔밥 (5.6.12.13.18)</p> <p>오이미역냉국(소) (13)</p> <p>참나물들깨무침 (5.6)</p> <p>무말랭이무침 (5.6.13)</p> <p>깎풍기 (1.5.6.13.15.18)</p> <p>배추김치 (9)</p> <p>친환경멜론(현)</p>	

알레르기 정보 : 1.난류 2.우유 3.메밀 4.땅콩 5.대두 6.밀 7.고등어 8.계 9.새우 10.돼지고기 11.복숭아 12.토마토 13.아황산류 14.호두 15.닭고기16.쇠고기 17.오징어 18.조개류(굴,전복,홍합 포함) 19.젓



오늘 뭐해?

캘린더
금식표
시간표

시간	월	화	수	목	금
1교시	자율활동	* SQL활용	한국사	음악 연주	자료 구조
2교시	* 인터페이스 구현	* SQL활용	영어 I	* 애플리케이션 테스트 수행	자료 구조
3교시	수학 I	* 인터페이스 구현	진로활동	영어 I	자율활동
4교시	영어 I	한국사	운동과 건강	운동과 건강	영어 I
점심시간	점심시간	점심시간	점심시간	점심시간	점심시간
5교시	* 애플리케이션 테스트 수행	* 인터페이스 구현	* SQL활용	자료 구조	동아리활동
6교시	* 애플리케이션 테스트 수행	* 인터페이스 구현	* SQL활용	자료 구조	
7교시	한국사	실용 국어		수학 I	

▼ 버그 & 어려웠던 점

로컬에 이미지가 저장되는 형식의 코드였는데 이 코드를 Naver Cloud로 저장하도록 하고 불러오는 것이 어려웠다.

```
// 프로필 이미지 업로드 하는 코드
app.post('/uploadimg', setUpload('profileimg'), (req, res) => {
```

```

try {
  if (!req.file) {
    console.log('No file uploaded.');
```

d.');

```
    return res.status(400).send('No file uploade
```

d.');

```
  }

  console.log('File uploaded:', req.file);

  const email = req.headers.email; // 헤더에서 이메일 가
  쳐오기
  if (!email) {
    console.error('No email provided in the reques
```

t.');

```
    return res.status(400).send('No email provide
```

d.');

```
  }

  const filePath2 = req.file.location;
  const fileName = path.basename(filePath2); // 파일
  이름만 추출

  const selectQuery = 'SELECT email FROM student WHE
  RE email = ?';
  db5.query(selectQuery, [email], (err, results) =>
  { // db5 = 학생정보 저장 테이블
    if (err) {
      console.error('Database error:', err);
      return res.status(500).send('Database erro
```

r.');

```
    }

    if (results.length > 0) {
      // 이메일이 존재하면 photoURL 업데이트
      const updateQuery = 'UPDATE student SET ph
      otoURL = ? WHERE email = ?';
      db5.query(updateQuery, [fileName, email],
      (updateErr, updateResults) => {

```

```

        if (updateErr) {
            console.error('Database error:', u
pdateErr);
            return res.status(500).send('Datab
ase error.');
```

```

        }
        res.json({ message: 'Photo URL updated
successfully.', filePath: fileName });
    });
    } else {
        // 이메일이 존재하지 않으면 에러 반환
        res.status(404).send('Email not found.');
```

```

    }
    });
} catch (error) {
    console.error(error.message);
    res.status(400).json({ message: error.message });
}
});

```

같은 팀 프론트를 제작했던 학생이 css를 지정하면서 모든 버튼들이 이 css 코드와 똑같아 지는 버그가 발생했었지만 고쳤다.

```

@font-face {
    font-family: 'Freepresentation-6SemiBold';
    src: url('https://cdn.jsdelivr.net/gh/projectnoonnu/2404@
font-weight: 600;
    font-style: normal;
}

@font-face {
    font-family: 'Freepresentation-5Medium';
    src: url('https://cdn.jsdelivr.net/gh/projectnoonnu/2404@
font-weight: 500;
    font-style: normal;
}

```



```

.modalOverlay {
  position: fixed;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: rgba(0, 0, 0, 0.7);
  display: flex;
  justify-content: center;
  align-items: center;
  animation: fadeIn 0.3s ease-in-out;
}

.modalContent {
  background-color: #fff;
  padding: 40px;
  border-radius: 12px;
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.3);
  text-align: center;
  width: 400px;
  height: 300px;
  animation: slideIn 0.3s ease-in-out;
  display: flex;
  flex-direction: column; /* 수직으로 배치하기 위해 추가 */
  align-items: center;      /* 수직으로 가운데 정렬 */
}

.fileInputLabel {
  display: inline-block;
  width: 168px;
  height: 34.5px;
  background-color: #007bff;
  color: #fff;
  border-radius: 100px;
  cursor: pointer;
  display: flex;
  justify-content: center;

```

```

    align-items: center;
    font-size: 16px;
    transition: background-color 0.3s ease;
    font-family: 'Freestart-5medium', sans-serif;
}

.fileInputLabel:hover {
    box-shadow: 0px 0px 0px 3px #69AEFF;
    background-color: #ffffff;
    color: black;
    transition: background-color 0.4s;
}

.fileInput {
    display: none;
}

.uploadimage {
    width: 300px;
    height: 242.5px;
    border: solid 4px black;
    border-radius: 50px;
    margin-bottom: 20px;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
}

.previewImage {
    margin-top: 20px;
    margin-bottom: 20px;
    width: 125px;
    height: 125px;
    object-fit: cover;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    display: block;
    margin-left: auto;
}

```

```

    margin-right: auto;
}

.uploadimage-img {
    width: 125px;
    height: 125px;
    flex-direction: column;
}

.uploadtext {
    font-family: 'Freeseentation-6SemiBold', sans-serif;
    font-size: 24px;
}

button {
    padding: 10px 20px;
    margin: 10px;
    border: none;
    border-radius: 8px;
    font-size: 16px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

button:first-of-type {
    background-color: #69AEFF;
    color: #fff;
}

button:first-of-type:hover {
    background-color: #01499b;
}

button:last-of-type {
    background-color: #69AEFF;
    color: #fff;
}

```

```

button:last-of-type:hover {
  background-color: #01499b;
}

@keyframes fadeIn {
  from { opacity: 0; }
  to { opacity: 1; }
}

@keyframes slideIn {
  from { transform: translateY(-20px); }
  to { transform: translateY(0); }
}

```

이 코드에서 12월에서 다음년도 1월로 가야하는데 24년에서 26년으로 가는 버그가 있었지만 고쳤다.

```

<div className={styles["angle-left"]} onClick={() => {
  if(currentMonth === 0) {
    setCurrentMonth(11);
    setCurrentYear(currentYear - 1);
  } else {
    setCurrentMonth(currentMonth - 1);
  }
}}>
  <i className="xi-angle-left xi-x"></i>
</div>
<span>{currentYear}년 {months[currentMonth]}</span>
<div className={styles["angle-right"]} onClick={() => {
  if(currentMonth === 11) {
    setCurrentMonth(0);
    setCurrentYear(currentYear + 1);
  } else {
    setCurrentMonth(currentMonth + 1);
  }
}}>

```

```
<i className="xi-angle-right xi-x"></i>  
</div>
```

▼ 좋았던 점

React를 처음으로 사용하면서 신기했던점이 분명 함수를 지정해주었는데 return 부분이 HTML이라는 것이었다. 그래서 여러가지 기능을 사용 해볼 수 있다는 것이 매우 좋았다. 그리고 Node.js + express라는 백엔드를 이용하면서 코드를 짜는 것이 정해져 있는 것이 아니고 여러 방법으로 짤 수 있다는 것이 좋았던 점이라고 생각한다.

▼ 아쉬웠던 점

계획서를 쓸때 웹을 개발한 후 앱까지 개발할 생각이었지만 처음해보는 프레임워크로 개발을 하다보니 하나부터 열까지 하게 되어 시간이 부족했다는 것이 아쉬웠다.

ㅌ ㅏ ㅓ

ㅌ ㅏ ㅓ

ㅌ ㅏ ㅓ

ㅌ ㅏ ㅓ

ㅌ ㅏ ㅓ

ㅌ ㅏ ㅓ