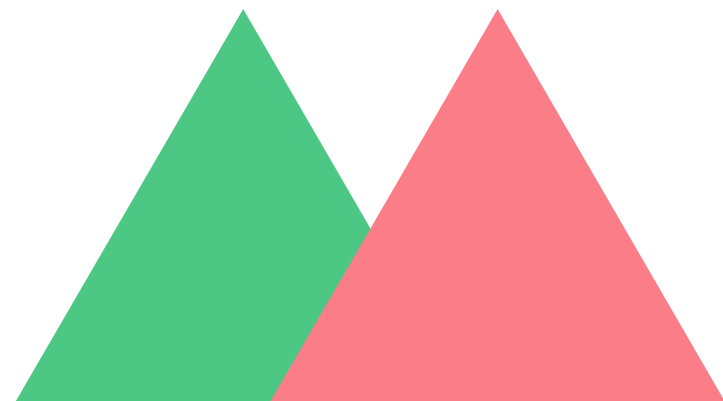


# 나만의 야식 다이어리

2014.2.7.31 우승민



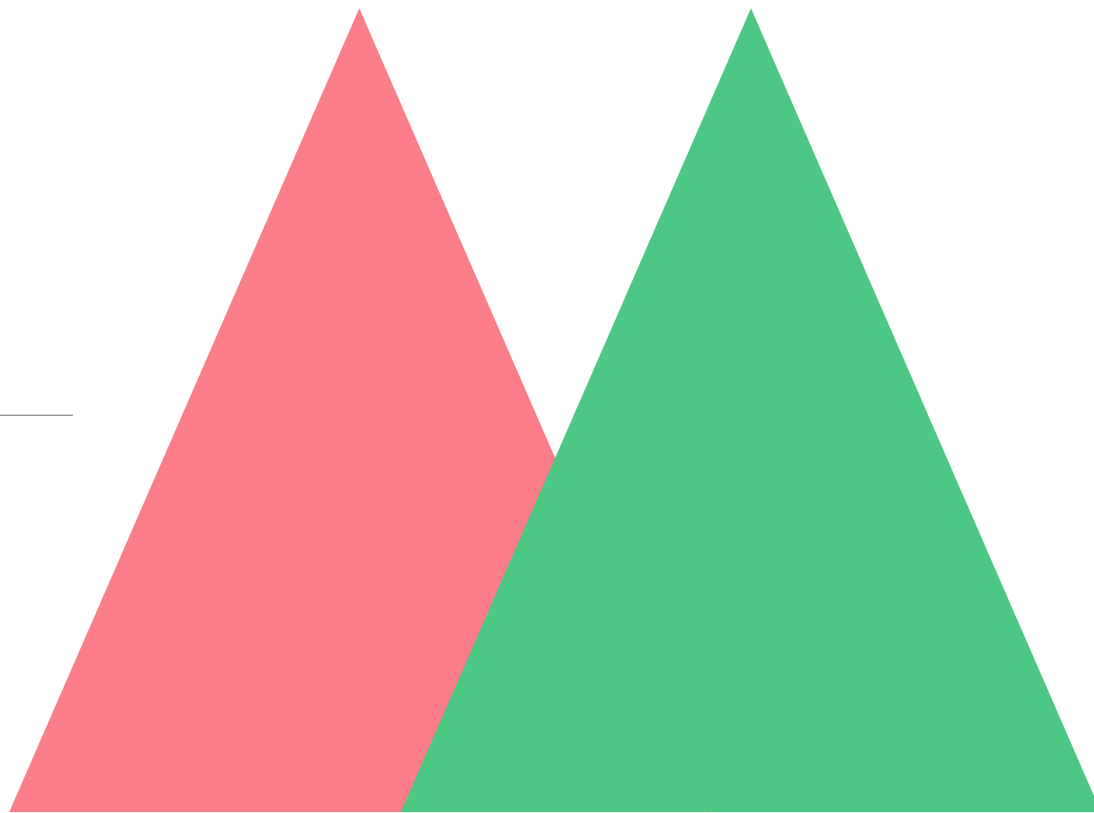
# Contents

- 1 **사이트맵 구성도**
- 2 **DB 구성도**
- 3 **모듈 구성도(기능 설명)**
- 4 **모듈별 소스 코드**
- 5 **시연**

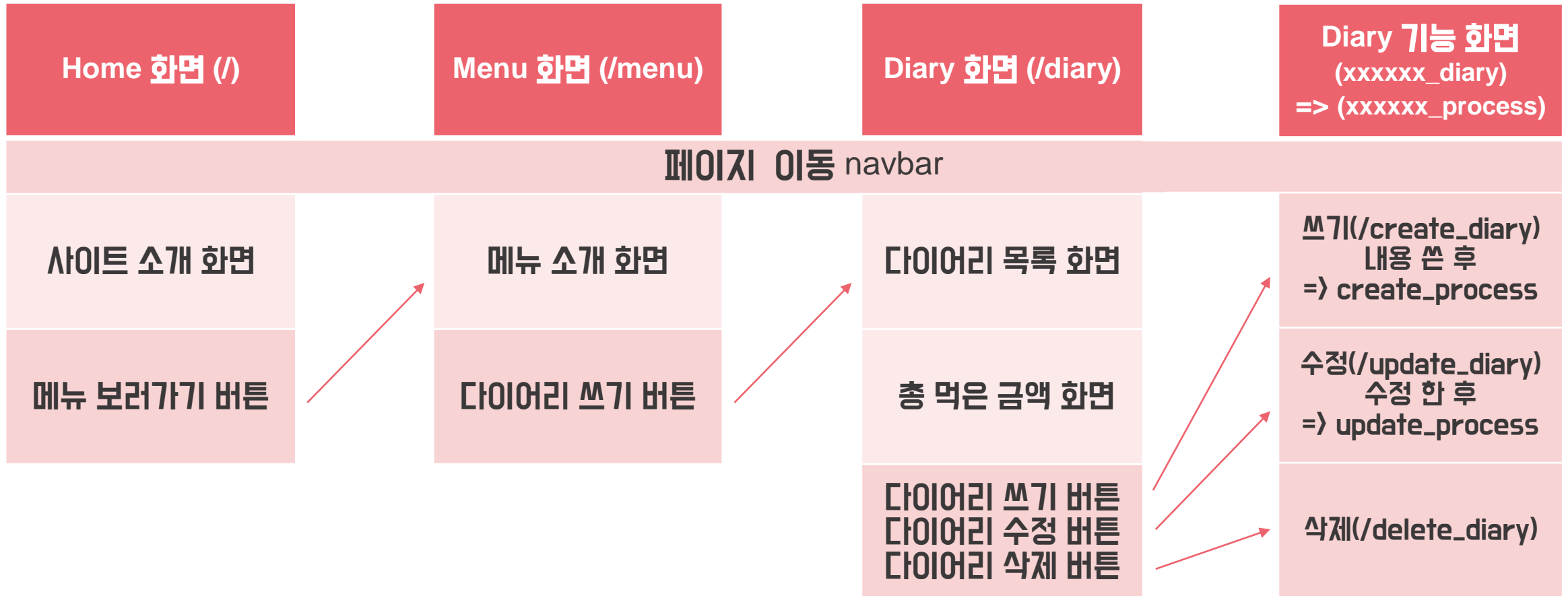
# 001

---

## 사이트맵 구성도



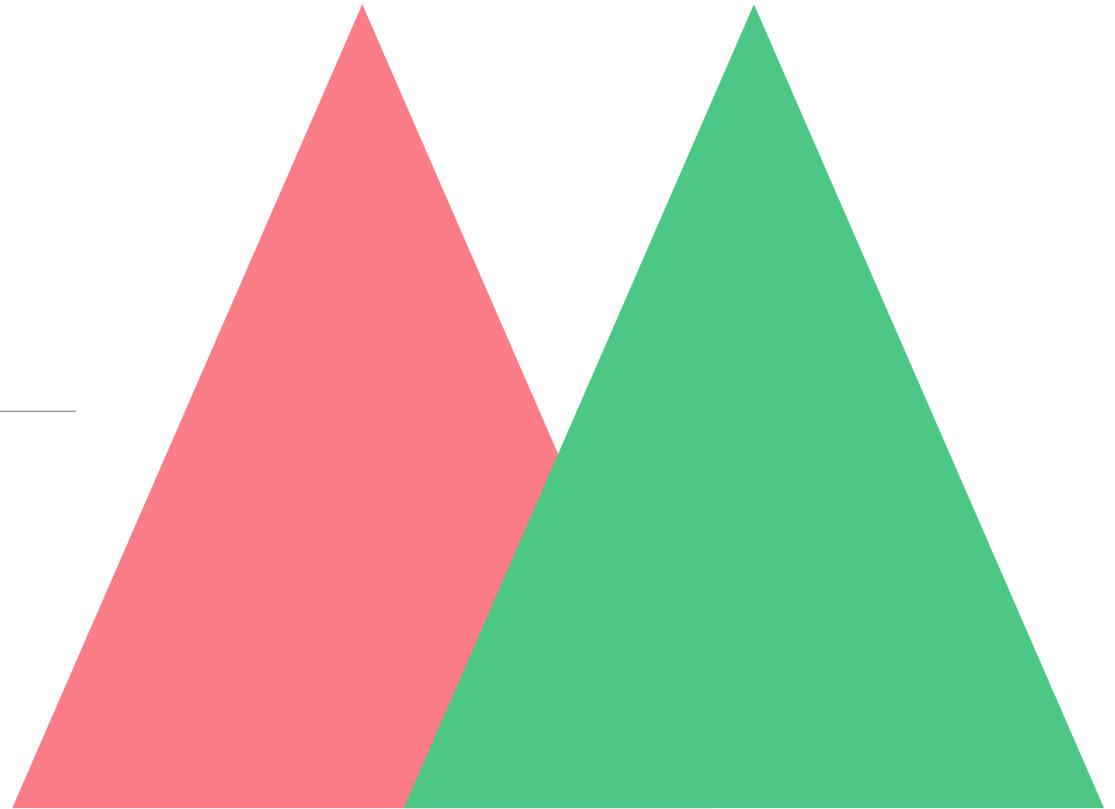
# 〈 사이트맵 구성도 〉



# 002

---

## DB구성도



## 2

## 〈 DB 구성도 〉

```

C:\WINDOWS\system32\cmd.exe - mysql.exe -h localhost -u...
mysql> desc category;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| cId   | int(11)       | NO   | PRI | NULL    | auto_increment |
| cName | varchar(20)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+

```

왼쪽에서 보실 수 있듯이 테이블은 총 3개로 구성했습니다.  
category에는 야식 카테고리의 Id, Name이 담기며,  
primary Key를 Id값으로 주었습니다.

```

C:\WINDOWS\system32\cmd.exe - mysql.exe -h localhost -u root -p
mysql> show tables;
+-----+
| Tables_in_diary |
+-----+
| category         |
| diary            |
| menu             |
+-----+

```

```

C:\WINDOWS\system32\cmd.exe - mysql.exe -h localhost -u root -p
mysql> desc menu;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| mId        | int(11)       | NO   | PRI | NULL    | auto_increment |
| cId        | int(11)       | NO   | MUL | NULL    |                |
| mName      | varchar(30)   | NO   |     | NULL    |                |
| mPrice     | int(20)       | NO   |     | NULL    |                |
| mDescription | text         | YES  |     | NULL    |                |
| mPicture   | text         | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.10 sec)

```

menu에는 야식 메뉴의 Id, 카테고리Id, Name, Price,  
Description, Picture이 담기며, primary Key는 ID값.  
cId는 foreign key로 category의 cId와 연결됩니다.

```

C:\WINDOWS\system32\cmd.exe - mysql.exe -h localhost -u root ...
mysql> desc diary;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| dId        | int(11)       | NO   | PRI | NULL    | auto_increment |
| dTime      | datetime      | NO   |     | NULL    |                |
| dTitle     | varchar(30)   | NO   |     | NULL    |                |
| dDesc      | text         | YES  |     | NULL    |                |
| mId        | int(11)       | NO   | MUL | NULL    |                |
| mMany      | int(3)        | NO   |     | NULL    |                |
| oPrice     | int(20)       | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+

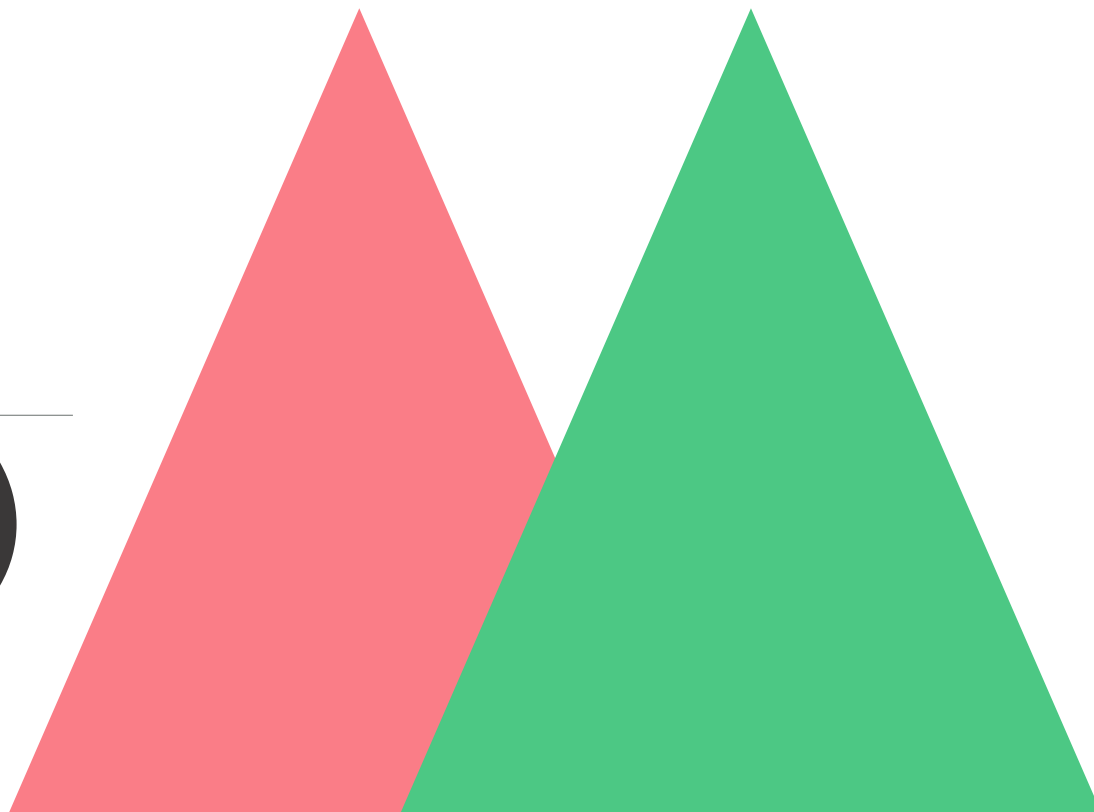
```

diary에는 다이어리 Id, Time, Title, Desc, 메뉴 Id, 수량,  
다이어리에 쓴 Price가 담기며, primary Key는 Id값.  
mId는 foreign key로 menu의 mId와 연결됩니다.  
또한 Price는 menu 테이블과 조인, 연산하여 저장합니다.

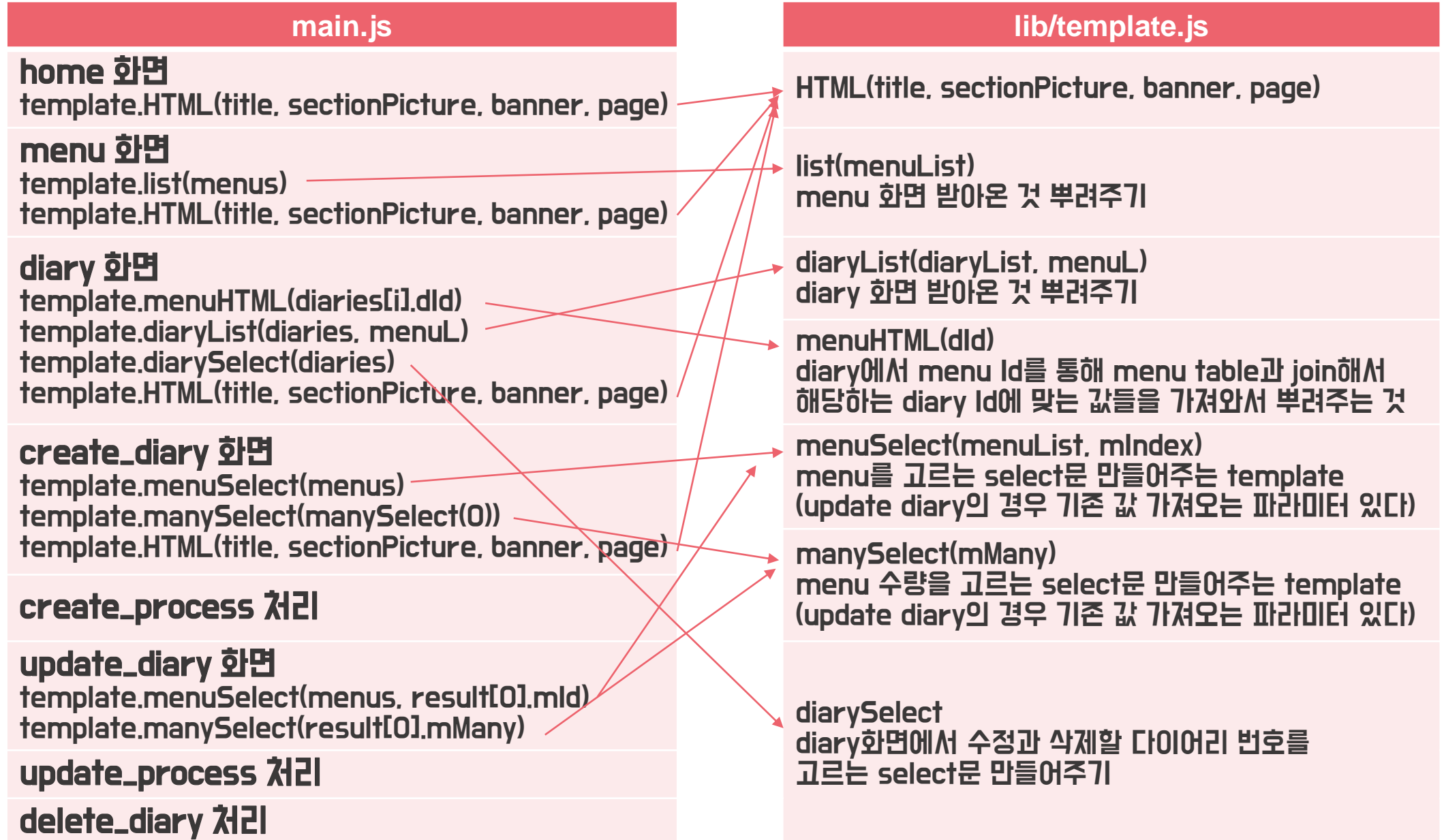
# 003

---

## 모듈 구성도(기능 설명)



# 〈 모듈 구성도 〉

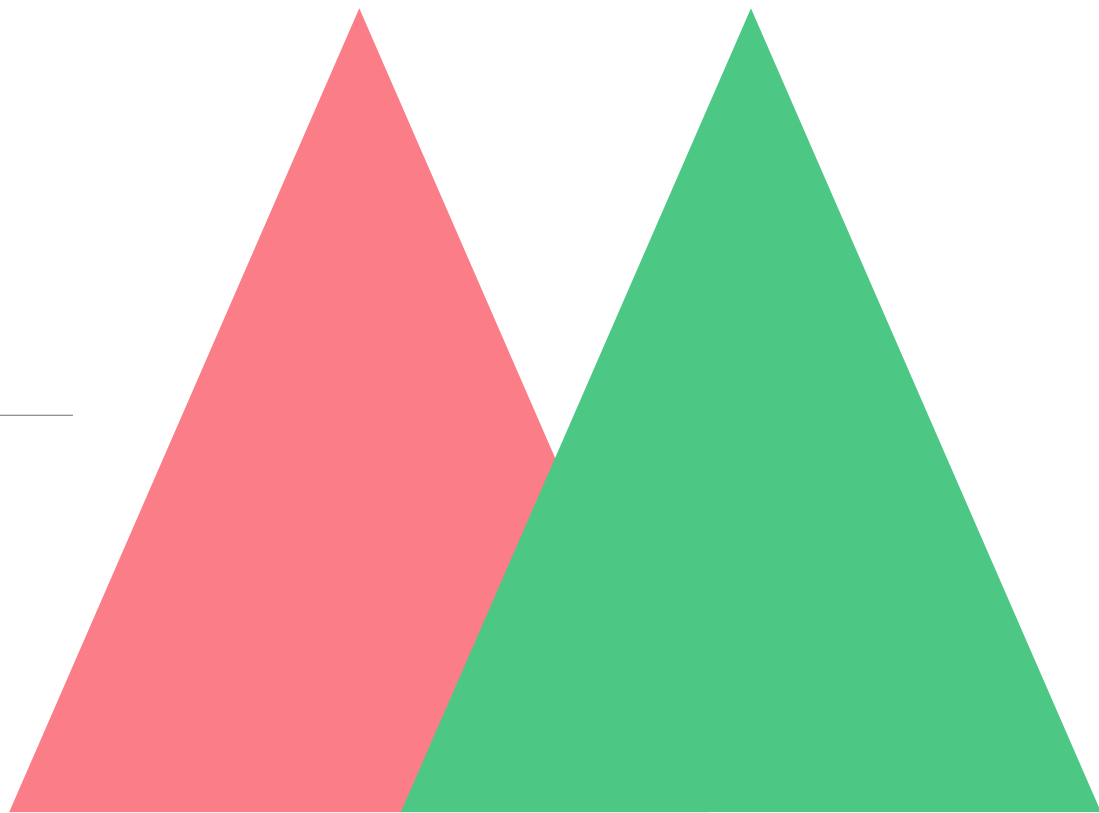




# 004

---

## 모듈별 소스 코드



## home 화면

```

var app = http.createServer(function(request,response){
  var _url = request.url;
  var pathname = url.parse(_url, true).pathname;
  if(pathname === '/') { // 초기 홈화면 보여주기
    var title = '무승민의 야식 다이어리';
    var sectionPicture = `<section class="banner-area text-center">`
    var banner = `<p class="tit">무승민의<br><span class="prime-color">야식 다이어리~</span></p>`
    var page = `
    <section class="welcome-area section-padding2">
      <div class="container-fluid">
        <div class="row">
          <div class="col-md-6 align-self-center">
            <div class="welcome-img">
              
            </div>
          </div>
          <div class="col-md-6 align-self-center">
            <div class="welcome-text mt-5 mt-md-0">
              <h4><span class="style-change">배달의민족</span></h4>
              <p>총 주문금액 보기에서<br>영감을 받았습니다.</p><br>
              <a href="/menu" class="template-btn mt-3 mm">메뉴 보러가기</a>
            </div>
          </div>
        </div>
      </div>
    </section>
    `;
    var html = template.HTML(title, sectionPicture, banner, page);
    response.writeHead(200);
    response.end(html);
  }
});

```

## menu 화면

```

} else if(pathname === '/menu'){           // 메뉴화면 보여주기(db에 저장된 데이터 바탕으로 가져오기)
  var title = '우승민의 메뉴';
  var sectionPicture = `<section class="banner-area banner-area2 menu-bg text-center">`
  var banner = `<h1><i>Menu</i></h1><p class="pt-2"><i>전부 다 맛있어 보이잖아!</i></p>`
  db.query('select * from menu', function(error, menus){
    var list = template.list(menus);
    var page = `
    <section class="food-area section-padding">
      <div class="container">
        <div class="row">
          |   ${list}
          <div class="col-md-4 col-sm-6">
            <br><br><br>
            <div class="single-food mt-5">
              <a href="/diary" class="template-btn mt-3 mm">다이어리 쓰기 !</a>
            </div>
          </div>
        </div>
      </div>
    </div>
    </section>
    `;
    var html = template.HTML(title, sectionPicture, banner, page);
    response.writeHead(200);
    response.end(html);
  });

```

## diary 화면

```

} else if(pathname === '/diary'){ // 다이어리 목록에 대한 화면 가져오기(쓰기, 수정, 삭제 버튼 지원)
    var title = '우승민의 다이어리';
    var sectionPicture = '<section class="banner-area banner-area2 blog-page text-center">';
    var banner = '<h1><i>Diary</i></h1><p>얼만큼 먹었는지 보고, 기록해요!</a>';
    db.query('select * from diary', function(error, diaries){
        db.query('select sum(oPrice) as sumP from diary', function(error, sum){
            var i = 0;
            var menuL = [];
            while(i < diaries.length){
                template.menuHTML(diaries[i].dId).then(function(result){
                    menuL.push(result);
                });
                i++;
            }
            setTimeout(function(){
                var list = template.diaryList(diaries, menuL);
                var page = `
                <section class="blog_area section-padding">
                    <div class="container">
                        <div class="row">
                            <div class="progress-table-wrap">
                                <div class="progress-table">
                                    <div class="table-head">
                                        <div class="serial">번호</div>
                                        <div class="visit">제목</div>
                                        <div class="visit">설명</div>
                                        <div class="country" style="width:430px;">먹은 메뉴</div>
                                        <div class="visit">먹은 수량</div>
                                        <div class="visit">먹은 금액</div>
                                        <div class="country">먹은 날짜</div>
                                    </div>
                                    ${list}
                                </div>
                            </div>
                        <div class="row">
                            <div class="col-lg-12">
                                <div class="section-top2 text-center">
                                    <p>그동안 먹은 총 금액은 ${sum[0].sumP} 원!</p>
                                    <form method="post" name="form">
                                        <input type="submit" class="genric-btn primary circle" value="다이어리 쓰기" onclick="javascript: form.action='/create_diary';"/>
                                        ${template.diarySelect(diaries)}
                                        <input type="submit" class="genric-btn primary circle" value="다이어리 수정" onclick="javascript: form.action='/update_diary';"/>
                                        <input type="submit" class="genric-btn primary circle" value="다이어리 삭제" onclick="javascript: form.action='/delete_diary';"/>
                                    </form>
                                </div>
                            </div>
                        </div>
                    </div>
                </section>
                `;
                var html = template.HTML(title, sectionPicture, banner, page);
                response.writeHead(200);
                response.end(html);
            }, 500);
        });
    });
}

```

# 〈 모듈별 소스 코드 -main.js 〉

create\_diary화면

create\_process

```

} else if(pathname === '/create_diary'){ // 다이어리 생성에 대한 화면 만들어주기
db.query('select * from menu', function(error, menus){
  var title = '우승민의 다이어리';
  var sectionPicture = '<section class="banner-area banner-area2 blog-page text-center">'
  var banner = '<h1><i>Diary</i></h1><p>얼만큼 먹었는지 보고, 기록해요!</a>'
  var page = `
<section class="blog_area section-padding">
  <div class="container">
    <div class="row">
      <div class="comment-form">
        <form action="/create_process" method="post">
          <div class="form-group form-inline">
            <div class="form-group col-lg-6 col-md-6 menu">
              <p>먹은 메뉴를 선택하세요${template.menuSelect(menus)}</p>
            </div>
            <div class="form-group col-lg-6 col-md-6 many">
              <p>먹은 수량을 선택하세요${template.manySelect(0)}</p>
            </div>
          </div>
          <div class="form-group">
            <p><input type="text" class="form-control" id="title" name="title" placeholder="제목" onfocus="this.placeholder = ''" onblur="this.placeholder = '제목'" required=""></p>
          </div>
          <div class="form-group">
            <p><textarea class="form-control mb-10" rows="3" id="description" name="description" placeholder="설명" onfocus="this.placeholder = ''" onblur="this.placeholder = '설명'"></p>
          </div>
          <input type="submit" class="but" value="다이어리 쓰기">
        </form>
      </div>
    </div>
  </div>
</section>
`;
  var html = template.HTML(title, sectionPicture, banner, page);
  response.writeHead(200);
  response.end(html);
});
} else if(pathname === '/create_process'){ // 다이어리 생성에 대한 post방식 form 처리
  var body = '';
  request.on('data', function(data){
    body = body + data;
  });
  request.on('end', function(){
    var post = qs.parse(body);
    db.query('select mPrice from menu where mId=?',[post.menuSelect], function(error, menu){
      var price = parseInt(menu[0].mPrice * post.manySelect);
      db.query('insert into diary (dTime, dTitle, dDesc, mId, mMany, oPrice) values(now(), ?, ?, ?, ?, ?)',
        [post.title, post.description, post.menuSelect, post.manySelect, price]), function(error, result){
        if(error){
          throw error;
        }
      }
      response.writeHead(302, {Location: '/diary'});
      response.end();
    });
  });
});

```

# 〈 모듈별 소스 코드 -main.js 〉

## update\_diary화면

```

} else if(pathname === '/update_diary'){ // 다이어리 업데이트에 대한 화면 만들어주기(선택한 다이어리의 기존값 가져옴)
db.query('select * from menu', function(error, menus){
  var title = '우승민의 다이어리';
  var sectionPicture = `<section class="banner-area banner-area2 blog-page text-center">`
  var banner = `<h1><i>Diary</i></h1><p>얼만큼 먹었는지 보고, 기록해요!</a>`
  var body = '';
  request.on('data', function(data){
    body = body + data;
  });
  request.on('end', function(){
    var post = qs.parse(body);
    db.query('select menu.mId, diary.mMany, diary.dTitle, diary.dDesc
    from menu join diary on menu.mId = diary.mId where diary.dId=?',
    [post.diarySelect], function(error, result){
      console.log(result);
      var page = `
<section class="blog_area section-padding">
  <div class="container">
    <div class="row">
      <div class="comment-form">
        <form action="/update_process" method="post">
          <input type="hidden" name="id" value="${post.diarySelect}">
          <div class="form-group form-inline">
            <div class="form-group col-lg-6 col-md-6 menu">
              <p>먹은 메뉴를 선택하세요${template.menuSelect(menus, result[0].mId)}</p>
            </div>
            <div class="form-group col-lg-6 col-md-6 many">
              <p>먹은 수량을 선택하세요${template.manySelect(result[0].mMany)}</p>
            </div>
          </div>
          <div class="form-group">
            <p><input type="text" value="${result[0].dTitle}" class="form-control" id="title" name="title" placeholder="제목" onfocus="this.placeholder = ''" onblur="this.placeholder = '제목'" required=""></p>
          </div>
          <div class="form-group">
            <p><textarea class="form-control mb-10" rows="3" id="description" name="description" placeholder="설명" onfocus="this.placeholder = ''" onblur="this.placeholder = '설명'">${result[0].dDesc}</textarea></p>
          </div>
          <input type="submit" class="but" value="다이어리 수정">
        </form>
      </div>
    </div>
  </div>
</section>
`;
  var html = template.HTML(title, sectionPicture, banner, page);
  response.writeHead(200);
  response.end(html);
});
});
}

```

update\_process

delete\_diary

```

    } else if(pathname === '/update_process'){ // 다이어리 업데이트에 대한 post방식 form 처리
        var body = '';
        request.on('data', function(data){
            body = body + data;
        });
        request.on('end', function(){
            var post = qs.parse(body);
            db.query('select mPrice from menu where mId=?',[post.menuSelect], function(error, menu){
                var price = parseInt(menu[0].mPrice * post.manySelect);
                db.query('update diary set dId=?, dTime=now(), dTitle=?, dDesc=?, mId=?, mMany=?, oPrice=? where dId=?',
                    [post.id, post.title, post.description, post.menuSelect, post.manySelect, price, post.id], function(error, result){
                        if(error){
                            throw error;
                        }
                        response.writeHead(302, {Location: `/diary`});
                        response.end();
                    });
            });
        });
    } else if(pathname === '/delete_diary'){ // 선택한 다이어리에 대한 삭제 기능
        var body = '';
        request.on('data', function(data){
            body = body + data;
        });
        request.on('end', function(){
            var post = qs.parse(body);
            db.query('delete from diary where dId=?', [post.diarySelect], function(error, result){
                if(error){
                    throw error;
                }
                response.writeHead(302, {Location: '/diary'});
                response.end();
            });
        });
    } else {
        response.writeHead(404);
        response.end('Not found');
    }
});
app.listen(3000);

```

## 〈 모듈별 소스 코드 -template.js 〉

```
<div class="main-menu main-menu2">
    <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/menu">Menu</a></li>
        <li><a href="/diary">Diary</a></li>
    </ul>
</div>
</div>
</div>
</header>

<sectionPicture>
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <div class="banner">
                    <!-- 템플릿 베너까지 끝 -->
                </div>
            </div>
        </div>
    </sectionPicture>
</div>

<!-- Javascript -->
<script src="http://kookmingps.cafe24.com/diary/assets/js/vendor/jquery-2.2.4.min.js"></script>
<script src="http://kookmingps.cafe24.com/diary/assets/js/vendor/bootstrap-4.1.3.min.js"></script>
<script src="http://kookmingps.cafe24.com/diary/assets/js/vendor/wow.min.js"></script>
<script src="http://kookmingps.cafe24.com/diary/assets/js/vendor/owl-carousel.min.js"></script>
<script src="http://kookmingps.cafe24.com/diary/assets/js/vendor/jquery.datetimepicker.full.min.js"></script>
<script src="http://kookmingps.cafe24.com/diary/assets/js/vendor/jquery.nice-select.min.js"></script>
<script src="http://kookmingps.cafe24.com/diary/assets/js/main.js"></script>
</body>
</html>
;
```



# 〈 모듈별 소스 코드 -template.js 〉

```

}, list: function(menuList){ // menu 화면 받아온 것 뿌려주기
  var list = '';
  var i = 0;
  while(i < menuList.length){
    list = list + `
      <div class="col-md-4 col-sm-6">
        <div class="single-food mt-5">
          <div class="food-img">
            
          </div>
          <div class="food-content">
            <div class="d-flex justify-content-between">
              <h5>${menuList[i].mName}</h5>
              <span class="style-change">₩${menuList[i].mPrice}</span>
            </div>
            <p class="pt-3">${menuList[i].mDescription}</p>
          </div>
        </div>
      </div>
    `;
    i = i + 1;
  };
  return list;
}, diaryList: function(diaryList, menuL) // diary 화면 받아온 것 뿌려주기
  var list = ``;
  var i = 0;
  while(i < diaryList.length){
    list = list + `
      <div class="table-row">
        <div class="serial">${diaryList[i].dId}</div>
        <div class="visit">${diaryList[i].dTitle}</div>
        <div class="visit">${diaryList[i].dDesc}</div>
        <div class="visit">${menuL[i]}</div>
        <div class="visit">${diaryList[i].mMany}</div>
        <div class="visit">${diaryList[i].oPrice}</div>
        <div class="country">${diaryList[i].dTime.toLocaleString()}</div>
      </div>
    `;
    i++;
  };
  return list;
};

```

```

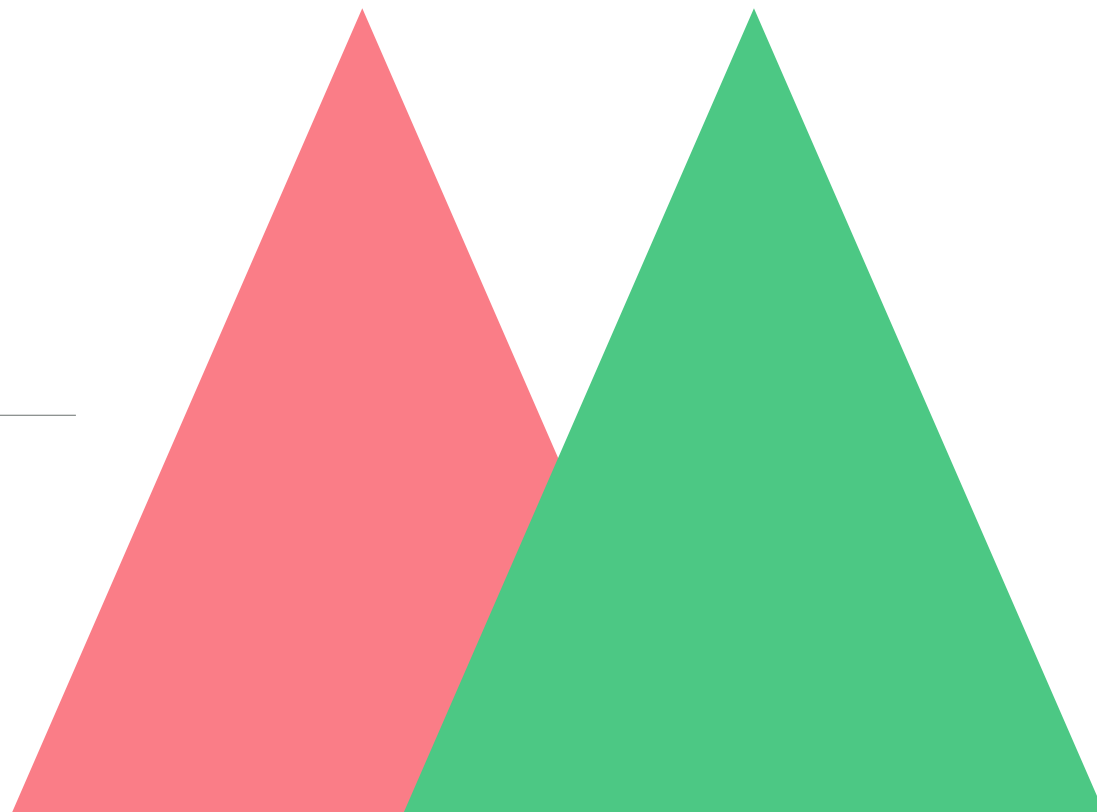
}, menuHTML: function(dId){ // diary에서 menu Id를 통해 menu table과 join해서 해당하는 diary Id에 맞는 값을 가져와서 뿌려주는 것
  return new Promise(function(resolve, reject){
    db.query('select mName, mPicture from menu join diary on menu.mId = diary.mId where diary.dId=?',[dId], function(error, menu){
      var menuName = menu[0].mName;
      var menuPicture = menu[0].mPicture;
      var html = `<div class="country" style="width:430px;">${menuName}</div>`;
      resolve(html);
    });
  });
}, menuSelect: function(menuList, mIndex){ // menu를 고르는 select문 만들어주는 template(update diary의 경우 기존 값 가져오기 위해 파라미터 있다)
  var tag = '';
  var i = 0;
  while(i < menuList.length){
    if(mIndex-1 == i){
      tag += `<option value="${menuList[i].mId}" selected>${menuList[i].mName}</option>`;
    } else {
      tag += `<option value="${menuList[i].mId}">${menuList[i].mName}</option>`;
    }
    i++;
  }
  return `
    <div class="input-group-icon mt-10">
      <div class="form-select" id="default-select"><select name="menuSelect">
        ${tag}
      </select></div>
    </div>
  `;
}, manySelect: function(mMany){ // menu 수량을 고르는 select문 만들어주는 template(update diary의 경우 기존 값 가져오기 위해 파라미터 있다)
  var tag = '';
  var i = 0;
  while(i < 10){
    if(mMany == i){
      tag += `<option value="${i}" selected>${i}</option>`;
    } else {
      tag += `<option value="${i}">${i}</option>`;
    }
    i++;
  }
  return `
    <div class="input-group-icon mt-10">
      <div class="form-select" id="default-select"><select name="manySelect">
        ${tag}
      </select></div>
    </div>
  `;
}, diarySelect: function(diaryList){ // diary화면에서 수정과 삭제할 다이어리 번호를 고르는 select문 만들어주기
  var tag = '';
  var i = 0;
  while(i < diaryList.length){
    tag += `<option value="${diaryList[i].dId}">${diaryList[i].dId} 번</option>`;
    i++;
  }
  return `
    <select class="default-select2" id="selectDiary" name="diarySelect">
      ${tag}
    </select>
  `;
};

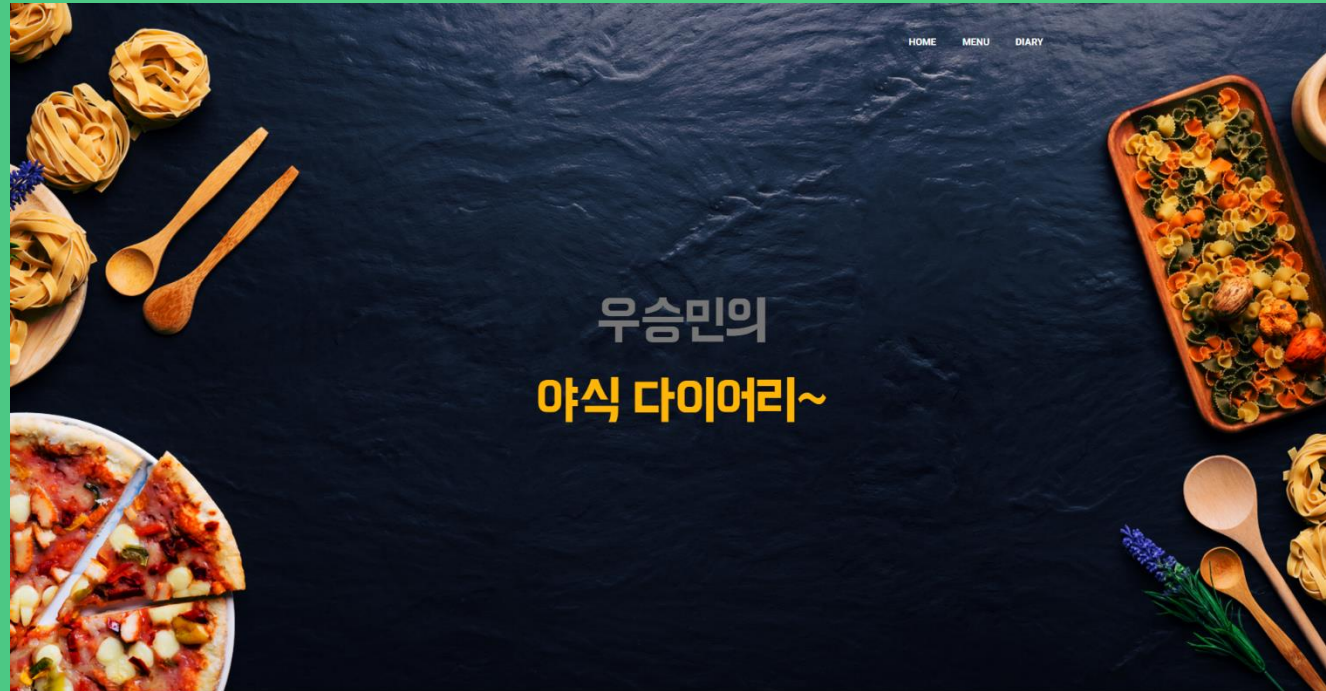
```

# 005

---

## 시연



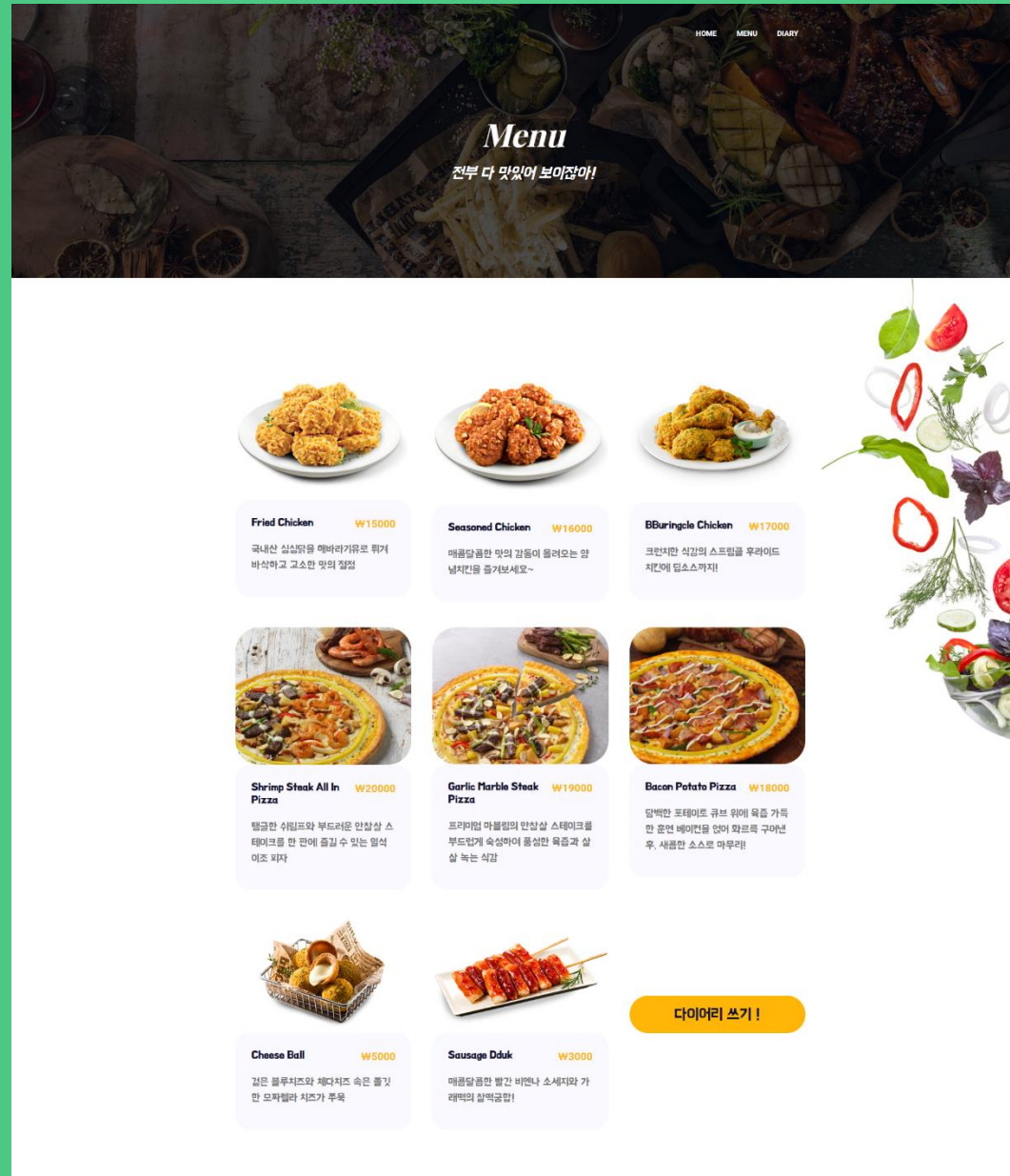


### 배달의민족

총 주문금액 보기에서  
영감을 받았습니다.

메뉴 보러가기

# 〈 시연 - 화면단위 캡처 - menu 〉



## 5 < 시연 - 화면단위 캡처 - diary null일 때 >



번호	제목	설명	먹은 메뉴	먹은 수량	먹은 금액	먹은 날짜
----	----	----	-------	-------	-------	-------

그동안 먹은 총 금액은 null 원!



다이어리 쓰기

다이어리 수정

다이어리 삭제



## 5

## 〈 시연 - 화면단위 캡처 - diary 쓰기 〉



먹은 메뉴를 선택하세요 Fried Chicken ^

먹은 수량을 선택하세요 0 v

제목

설명

- Fried Chicken
- Sausage Chicken
- BBQing Chicken
- Skirt Steak All In Pizza
- Garlic Florida Steak Pizza
- Bacon Potato Pizza
- Cheese Ball
- Sausage Dduk

다이어리 쓰기



번호	제목	설명	먹은 메뉴	먹은 수량	먹은 금액	먹은 날짜
25	부림골 7개 먹은날	맛있었다 !!	BBQing Chicken	7	119000	2019-11-24 17:08:07
26	부림골엔 소떡소떡	3개는 거든애!	Sausage Dduk	3	9000	2019-11-24 17:09:27

그동안 먹은 총 금액은 128000 원!

25 번 v

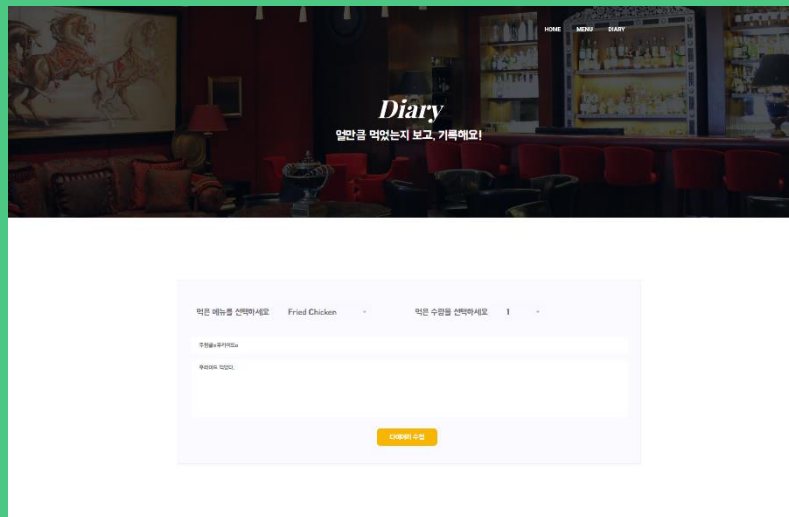
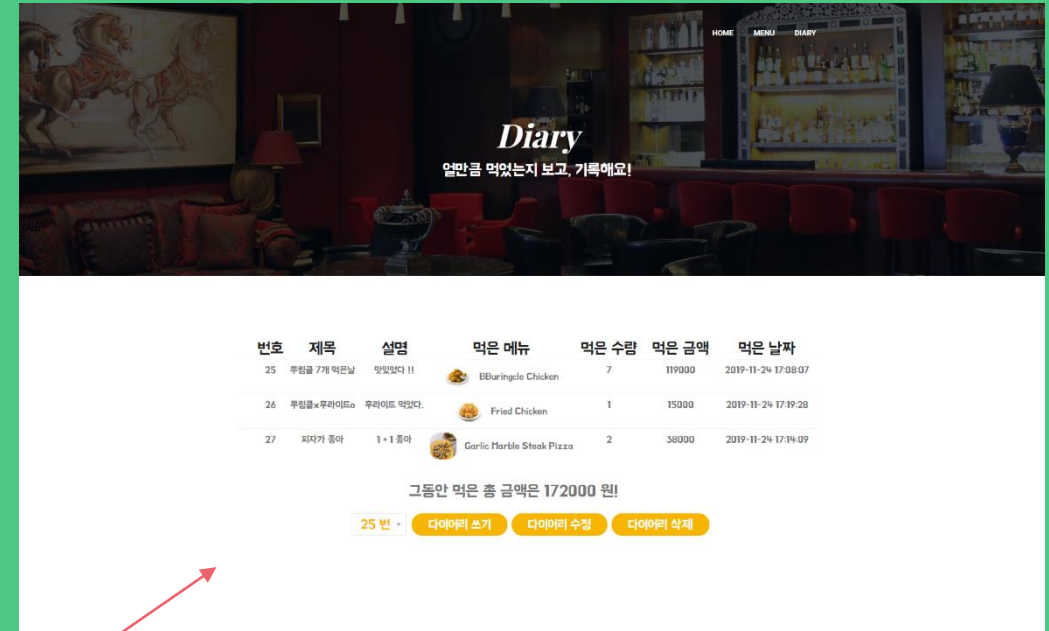
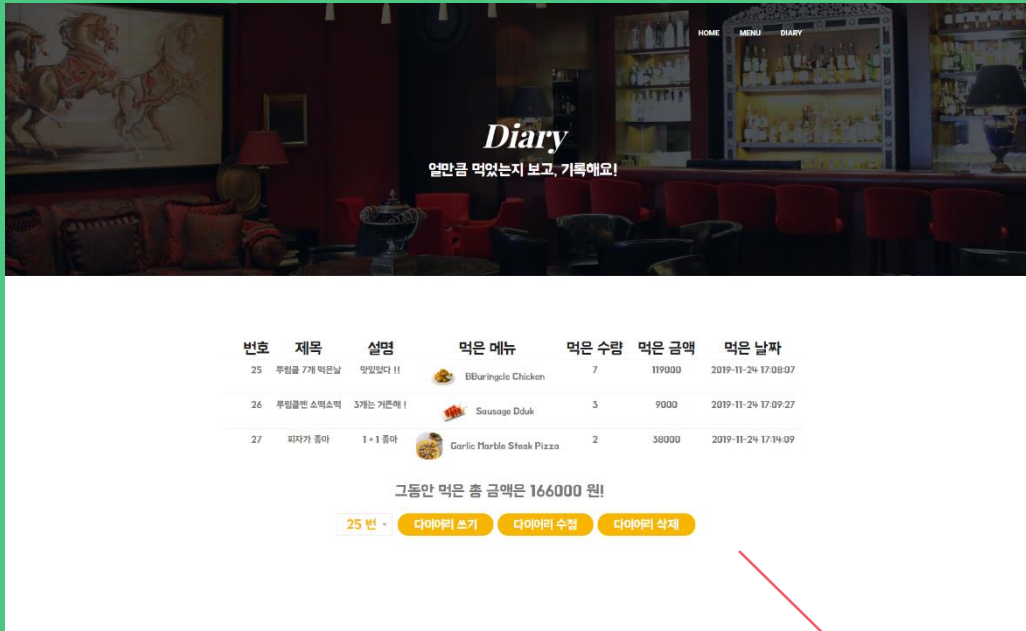
다이어리 쓰기

다이어리 수정

다이어리 삭제

## 5

## 〈 시연 - 화면단위 캡처 - diary 수정 〉



# 〈 시연 - 화면단위 캡처 - diary 삭제 〉





시연 영상은 <https://youtu.be/2W-BqPaD2o8> 에서 보실 수 있습니다.

자세한 코드는

[https://github.com/wsmwin/Late-night-snacks\\_node.js-mysql](https://github.com/wsmwin/Late-night-snacks_node.js-mysql)

여기서 보실 수 있습니다.

---

감 사 합 니 다

---