

CanYou Project

이 정 희

작품 설명

- 대학교에서는 스스로 시간표를 만든다.
- 학교마다 졸업요건이 있다.
- 졸업을 위해 학점 관리와 요건 체크를 해야함

➡ 학점 관리를 도와주고,

졸업 요건 충족 여부를 쉽게 확인 가능하게 해주는 ‘CanYou’

- 성신여대 기준으로 개발

작품 설명(Scenario)

1. 회원가입 후 로그인 (이메일로 비밀번호 찾기 가능)
2. 졸업 요건 페이지에서 졸업 요건 등록
 - 과목요건(교양, 전공), 과목 유형 요건(세부요건)
영역 요건, 총학점 요건
3. 강의 등록 페이지에서 수강한 강의 등록
4. 결과보기 페이지에서 졸업 요건 충족 여부 확인
 - 각 요건은 pass → 초록색, fail → 빨간색 표시
 - 모든 요건 pass → Congratulation 이미지 표시

Technology

Language - C#(IDE : VisualStudio)

Framework

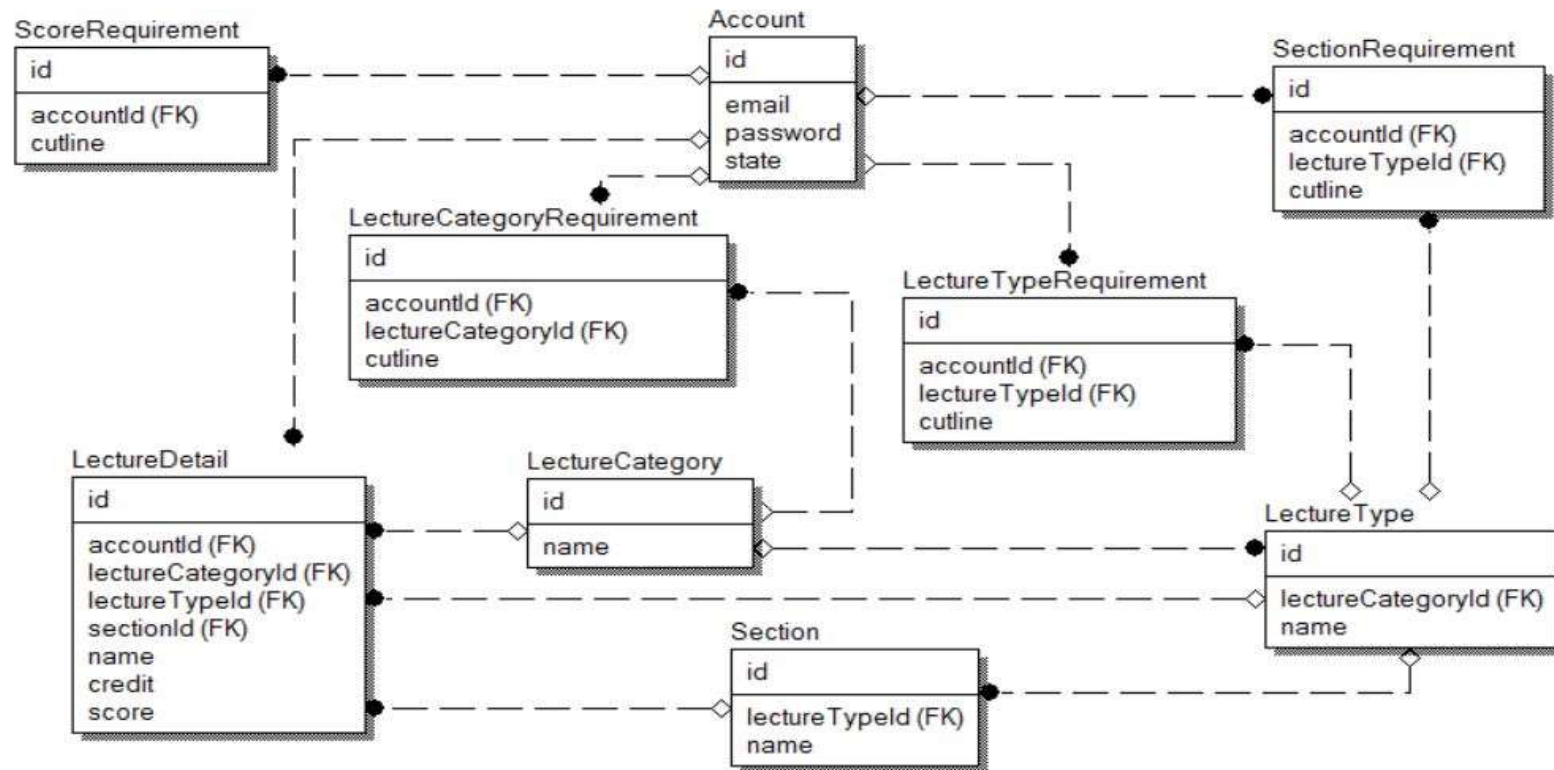
- ASP.NET MVC Framework
- SPRING.NET Framework

Front - javascript, jQuery

Database - MSSQL

CSS - PURE.CSS

CanYou DB ERD



CanYou 구성(Model)

- Item Class
data 교환을 위한 Class(data mapping)
property & getter, setter 메소드로 이루어짐
- Dao Class(Interface)
데이터베이스의 데이터에 접근하는 Class(객체)
DB 쿼리를 실행하는 메소드 제공
- Mapper Class
데이터베이스에서 SELECT한 각 필드를
Item 객체의 속성과 매핑하는 Class

Model Code(DAO)

참조 0개

```
public class AccountDao : AdoDaoSupport, IAccountDao
```

```
{
```

참조 2개

```
public int Insert(AccountItem item)
```

```
{
```

```
    string query = "Insert into account(email,password,state) VALUES(@Email,@Password,@State)";
```

```
    IDbParameters param = CreateDbParameters();
```

```
    param.AddWithValue("Password", item.Password).DbType = DbType.String;
```

```
    param.AddWithValue("Email", item.Email).DbType = DbType.String;
```

```
    param.AddWithValue("State", item.State).DbType = DbType.String;
```

```
    return AdoTemplate.ExecuteNonQuery(CommandType.Text, query, param);
```

```
}
```

참조 2개

```
public int Update(AccountItem item)
```

```
{
```

```
    string query = "UPDATE account SET password = @Password, email = @Email, state = @State where id = @Id";
```

```
    IDbParameters param = CreateDbParameters();
```

```
    param.AddWithValue("Id", item.Id).DbType = DbType.Int32;
```

```
    param.AddWithValue("Password", item.Password).DbType = DbType.String;
```

```
    param.AddWithValue("Email", item.Email).DbType = DbType.String;
```

```
    param.AddWithValue("State", item.State).DbType = DbType.String;
```

```
    return AdoTemplate.ExecuteNonQuery(CommandType.Text, query, param);
```

```
}
```

참조 5개

```
public AccountItem FindByEmail(string email)
```

```
{
```

```
    string query = "select * from account where email=@Email";
```

```
    IDbParameters param = CreateDbParameters();
```

```
    param.AddWithValue("Email", email).DbType = DbType.String;
```

```
    IList<AccountItem> list = AdoTemplate.QueryWithRowMapper<AccountItem>(CommandType.Text, query, new AccountMapper(), param);
```

```
    if (list.Count == 0) return null;
```

```
    return list[0];
```

```
}
```

Model Code(Item, Mapper)

Item

```
namespace CanYou.Models.Info.Account
{
    참조 19개
    public class AccountItem
    {
        참조 28개
        public int Id { get; set; }
        참조 8개
        public string Password { get; set; }
        참조 6개
        public string Email { get; set; }
        참조 6개
        public string State { get; set; }
    }
}
```

Mapper

```
namespace CanYou.Models.Info.Account
{
    참조 1개
    public class AccountMapper : IRowMapper<AccountItem>
    {
        참조 8개
        public AccountItem MapRow(System.Data.IDataReader reader, int rowNum)
        {
            AccountItem account = new AccountItem();
            account.Id = reader.GetInt32(reader.GetOrdinal("id"));
            account.Email = reader.GetString(reader.GetOrdinal("email"));
            account.Password = reader.GetString(reader.GetOrdinal("password"));
            account.State = reader.GetString(reader.GetOrdinal("state"));
            return account;
        }
    }
}
```


CanYou 구성(View) & Code

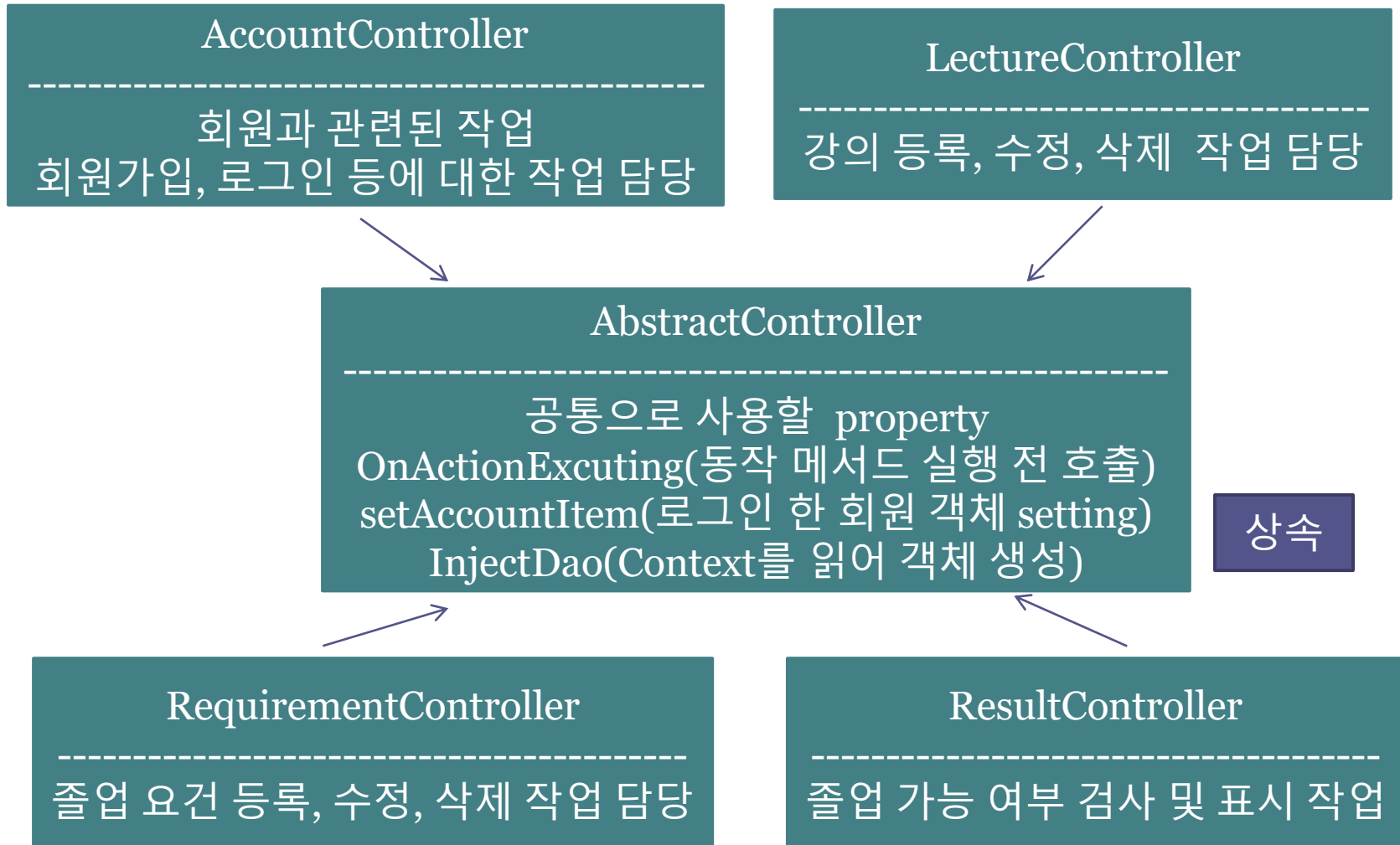
Ajax(AjaxForm) - form의 데이터를 로컬 서버에 전송

jQuery validation plugin - form을 submit할 때 유효성 검사 수행

```
$("#btnLogin").click(function () {  
    if (!$("#frmLogin").valid()) return;  
  
    $("#frmLogin").ajaxForm({  
        success: function (data, textStatus, xhr, $form) {  
            if (data.result == 'fail') {  
                alert(data.message);  
            } else {  
                $(location).attr('href', "/Lecture/List");  
            }  
        },  
        dataType: 'json'  
    });  
    $("#frmLogin").submit();  
    return false;  
});  
// dataType - 응답받을 dataType if Json -> success  
// [Authorize] 로그인 된 상태(SetAuthCookie)여야 한다.  
// 로그인이 안 되어있을 경우 web.config 의 loginpageurl 에 설정한 로그인 url 로 자동 리다이렉션
```

```
$("#frmLogin").validate({  
    rules: {  
        password: {  
            required: true,  
            minlength: 8  
        },  
        email: {  
            required: true,  
            email: true  
        },  
    },  
    messages: {  
        password: {  
            required: "암호를 입력해 주세요.",  
            minlength: "암호는 8자 이상이어야 합니다."  
        },  
        email: "형식에 맞는 이메일을 입력해 주세요."  
    },  
});
```

CanYou 구성(Controller)



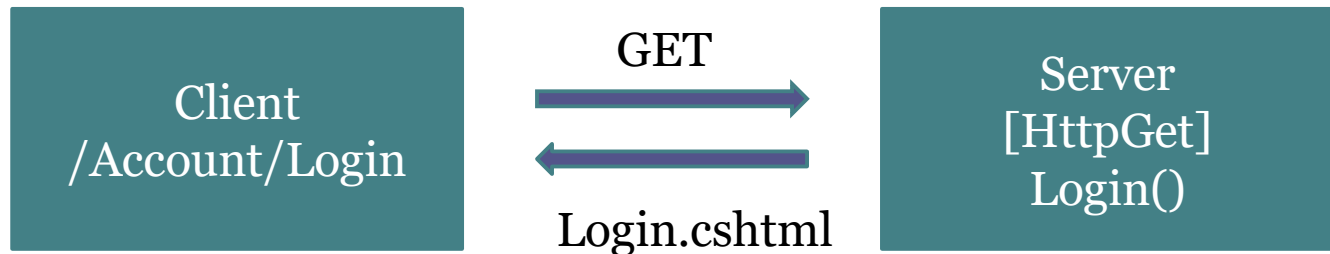
Controller Code & 동작

ex) Client : /Account/Login URL(로그인페이지)로 HTTP GET 요청

Code

```
참조 0개
public class AccountController : AbstractController
{
    [HttpGet]
    참조 0개
    public ActionResult Login()
    {
        // 이미 로그인 - list page로 redirect
        if (accountItem != null) return Redirect("/Lecture/List");
        // Login.cshtml return
        return View();
    }
}
```

동작



Controller Code

ex) Client : /Account/Login URL(로그인페이지)로 HTTP POST요청

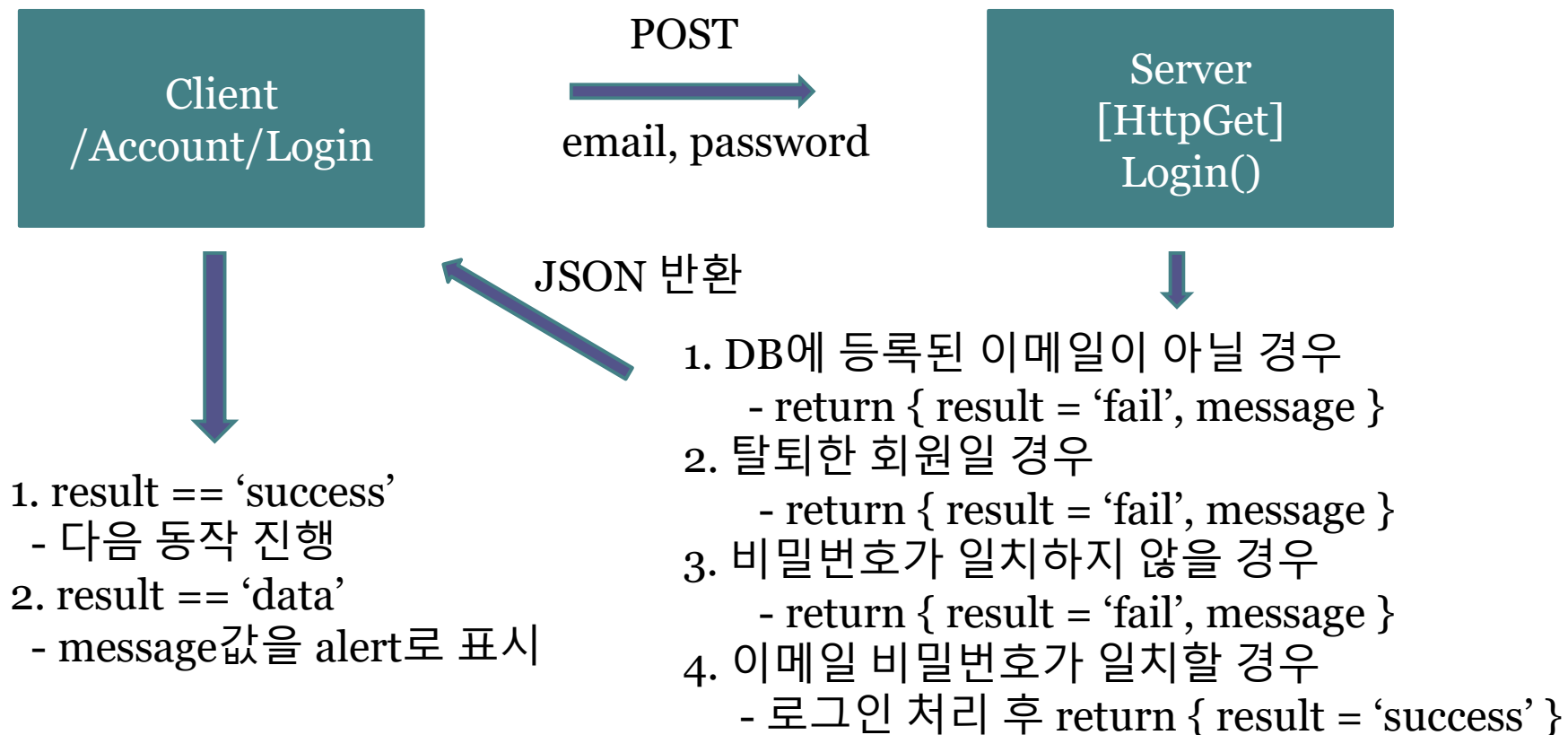
Code

```
// 액션은 특정 URL에 대한 요청이 들어올 때 호출되는 컨트롤러 상의 메서드
// 액션 결과는 컨트롤러 액션이 브라우저 요청에 대해 반환하는 것을 의미
[HttpPost]
참조 0개
public ActionResult Login(string email, string password)
{
    AccountItem item = AccountDao.FindByEmail(email);
    if (item == null) return Json(new { result = "fail", message = "존재하지 않는 이메일입니다." }, JsonRequestBehavior.AllowGet);
    if (item.State == "DEL") return Json(new { result = "fail", message = "탈퇴한 이메일입니다." }, JsonRequestBehavior.AllowGet);
    if (!item.Password.Equals(password)) return Json(new { result = "fail", message = "존재하지 않는 비밀번호입니다." }, JsonRequestBehavior.AllowGet);
    // 쿠키 할당
    FormsAuthentication.SetAuthCookie(item.Email, true);
    Session["account"] = item;
    // JsonResult 액션 결과 반환
    // default JsonRequestBehavior.DenyGet - HttpGet x
    return Json(new { result = "success" }, JsonRequestBehavior.AllowGet);
}
```

Controller 동작

ex) Client : /Account/Login URL(로그인페이지)로 HTTP Post 요청

동작



CanYou Page Images

localhost:17010/Lecture/Register

강의등록

강의관리

졸업요건관리

과목요건

과목유형요건

영역요건

총학점요건

결과보기

회원관리

LOGOUT

과목구분: 교양

과목유형구분: 공통

영역: 구분없음

학점: 숫자를 입력하세요.

성적: 받은 성적을 입력하세요.

교과목명: 교과목명을 입력하세요.

등록 취소

localhost:17010/Requirement/Type

강의유형요건 List

*과목유형요건이 존재하지 않는다면 입력하지 않으면 됩니다.

	과목구분	과목유형구분	이수학점	수정	삭제
1	전공	핵심	27	수정	삭제
2	교양	공통	5	수정	삭제
3	교양	일반	20	수정	삭제
4	교양	핵심	33	수정	삭제

과목유형등록

CanYou Page Images

← → ↻ localhost:17010/Result/Index

JeongHee sopt Revcontent Integrat

사용법

강의관리

졸업요건관리

과목요건

과목유형요건

영역요건

총학점요건

결과보기

회원관리

LOGOUT

결과 – fail

결과보기

No	요건명	요건	현재 상태	결과	
1	부전공	15	33	18학점 초과 이수하였습니다.	
2	교양	33	39	6학점 초과 이수하였습니다.	
3	교양 공통	5	10	5학점 초과 이수하였습니다.	
4	교양 일반	20	11	9학점 이 부족합니다.	
5	교양 핵심	33	18	15학점 이 부족합니다.	
6	전공	30	51	21학점 초과 이수하였습니다.	
7	전공 핵심	27	45	18학점 초과 이수하였습니다.	
8	핵심 교양 이수 영역 수	3	3	이수 요건을 만족하였습니다.	
9	총학점	129	153	24학점 초과 이수하였습니다.	

CanYou Page Images

← → ↻ localhost:17010/Result/Index

JeongHee sept Revcontent Integra

사용법

강의관리

출업요건관리

과목요건

과목유형요건

영역요건

총학점요건

결과보기

회원관리

LOGOUT

결과 – pass

결과보기

No	요건명	요건	현재 상태	결과	
1	부전공	15	33	18학점 초과 이수하였습니다.	
2	교양	33	39	6학점 초과 이수하였습니다.	
3	교양 공통	5	10	5학점 초과 이수하였습니다.	
4	교양 일반	11	11	이수 요건을 만족하였습니다.	
5	교양 핵심	18	18	이수 요건을 만족하였습니다.	
6	전공	30	51	21학점 초과 이수하였습니다.	
7	전공 핵심	27	45	18학점 초과 이수하였습니다.	
8	핵심 교양 이수 영역 수	3	3	이수 요건을 만족하였습니다.	
9	총학점	129	153	24학점 초과 이수하였습니다.	

Congratulations

배운 것들

- 전반적인 웹 프로그래밍 프로세스 경험
- 웹 동작 방식(HTTP GET, POST 등) 이해
- ASP.NET MVC , jQuery 학습

CanYou Project Codes

<https://github.com/jjoylee/CanYou>