



2019 Global Azure Bootcamp Seoul

# Azure를 이용한 간단한 블로그 만들기의 여정

김대희

# 발표자 소개

---

- 비상교육 ( Junior 개발자 / 2017.12 ~ )
- 서경대학교 컴퓨터공학과
- Back-End Developer
- Spring, PHP, etc..

**그런데 Azure로  
블로그를 왜 ?**

**그 이야기를 하고자 합니다.**

# 목차

---

1. 왜 블로그를 만들었는가?
2. 블로그를 만들다
3. 블로그 리뉴얼 도전
4. 결론

# 왜 블로그를 만들었는가?

---

## 1. 2017년 대학교 4학년

- 웹을 공부함
- 웹을 하는데 나를 표현할 사이트가 없음

## 2. No Desktop

- 24시간동안 접속할 수 있는 사이트 필요

## 3. Jobs...

# 왜 블로그를 만들었는가?

---

## 4.함께 만들어 보고 싶었음

# 일단 블로그를 만들자



# 블로그를 만들다

---

호스팅?

클라우드?

## 고려사항

Spring?

.NET?

# 블로그를 만들다

---

## 블로그를 만들었을 때의 고려사항 (1)

### 1. 호스팅과 클라우드

- 24시간 돌릴 서버를 찾고 있었음
- 당시에 C모 호스팅, G모 호스팅 등이 존재하였음
- 클라우드가 당시 엄청나게 급부상하고 있었음

### 2. 개발 도구의 선택

- 당시의 지식으로는 스프링, ASP.NET MVC 등으로 웹 개발을 파악
- 당시에 아는 IDE는 Eclipse, Visual Studio 정도

# 블로그를 만든다.

---

## 블로그를 만들었을 때의 고려사항 (2)

### 3. 함께 하고 싶은 언어

- 친구들과 함께 하고 싶은 언어를 정함
- 당시에 많이 쓰는 언어 외에 재밌는 걸 해보고 싶었음

### 4. 부담없이 재미있게

- 장기간보다는 단기간
- 무겁지 않고 가볍게 만들기

# 블로그를 만들다

---

그래서 선택은?

# 블로그를 만들다

---

## ASP.NET MVC5 + Azure

# 블로그를 만들다

---

## 왜 Azure?

# 블로그를 만들다

---



# Visual Studio

# 블로그를 만들다

---



## ASP.NET MVC5



# 블로그를 만들다

---



# Azure

# 블로그를 만들다

---

**왜 Azure?  
재밋고 쉬워 보여서.**

# 블로그를 만들다

## App Services

기본 디렉터리

+ 추가    ≡ 열 편집    ↺ 새로 고침    |    🏷 태그 지정    ▶ 시작    ↺ 다시 시작    ■ 중지    🗑 삭제

구독: 종량제



이름을 기준으로 필터링...

모든 리소스 그룹

모든 위치

모든

2개 항목

<input type="checkbox"/> 이름 ↑↓	상태	앱 유형	APP SERVICE 계획
<input type="checkbox"/>  DaeHeeKim	Running	웹 앱	PortfolioWeb2017092104...
<input type="checkbox"/>  Teamimpact	Running	웹 앱	PortfolioWeb2017092104...

# 블로그를 만들다

---

## 게시

---


Azure 또는 다른 호스트에 앱을 게시하세요. [자세한 정보](#)

 DaeHeeKim - Web Deploy ▼

게시(U)

[새 프로필...](#)

동작 ▼

사이트 URL	<a href="https://daeheekim.azurew...">https://daeheekim.azurew...</a> 	<a href="#">App Service 설정 편집</a>
리소스 그룹	Impact	<a href="#">클라우드 탐색기에서 관리</a>
구성	Release	<a href="#">미리 보기</a>
문제 해결 정보	<a href="#">가이드 참조</a>	<a href="#">구성</a>

## 지속적인 업데이트

---

지속적인 업데이트를 사용하여 Azure에 응용 프로그램을 자동으로 게시합니다.

[구성](#)

# 블로그를 만들다

---

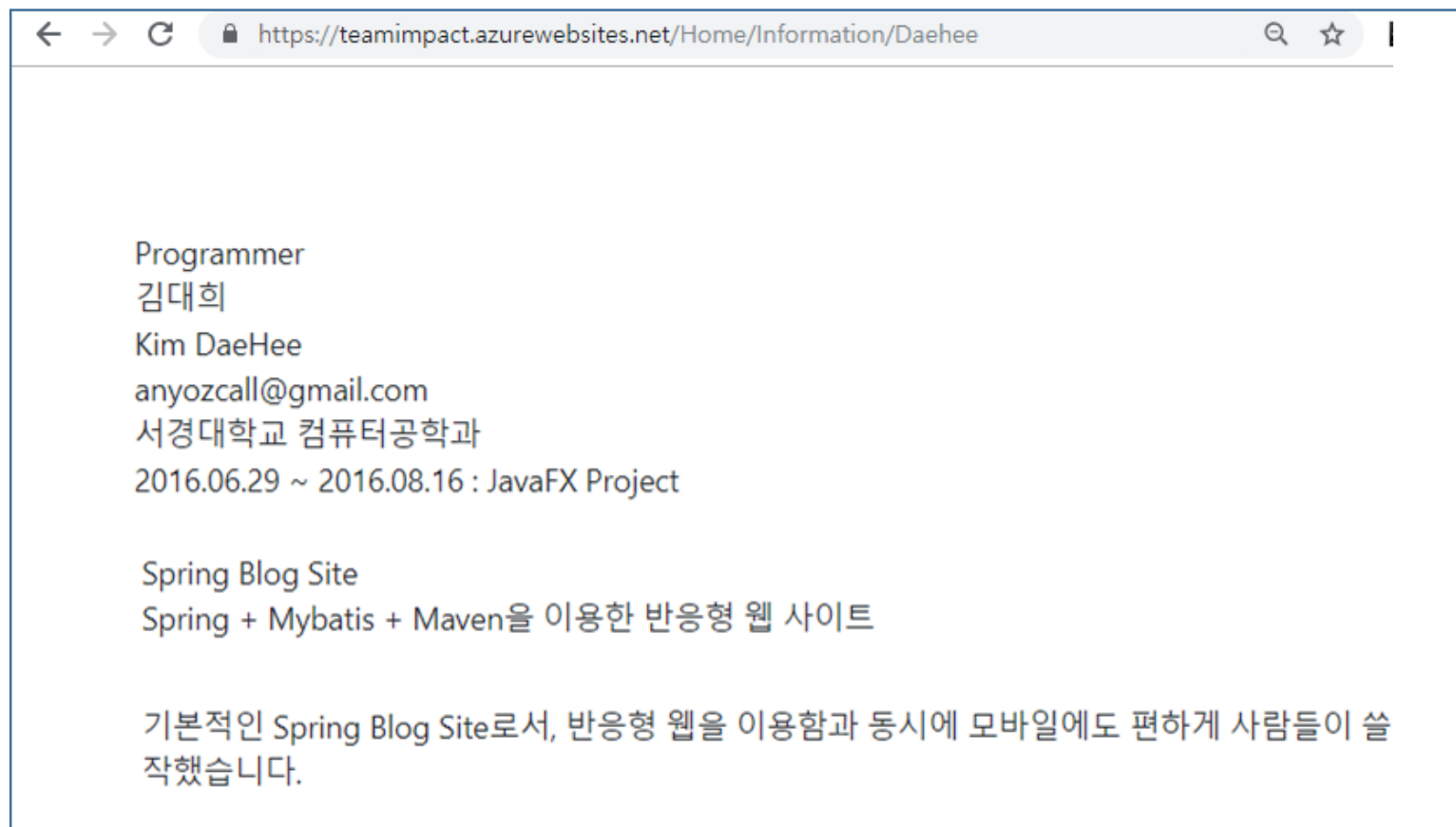
## 첫 동작

# 블로그를 만들다

---

일단 텍스트를 띄우자

# 블로그를 만들다



# 블로그를 만든다.

---

## 텍스트를 띄운 이후..

### 1. 프로필은 어떻게 가져올 것인가?

- DB 연동 / XML, JSON 파싱
- 당시에는 DB를 잘 못 다루어서 파일로 가져오는 것이 편했음

### 2. 해당 File은 어떻게 불러올 것인가?

- XML과 같은 구조인 Class 생성
- XML이 있는 경로 지정 후 해당 Class에 Mapping
- 해당 Mapping 된 경로를 뿌림





# 블로그를 만든다.

---

## 뭔가 부족하다...

# 블로그를 만든다.

---

## 수정사항

### 1. 글로만 있는 블로그를 어떻게 수정할 것인가?

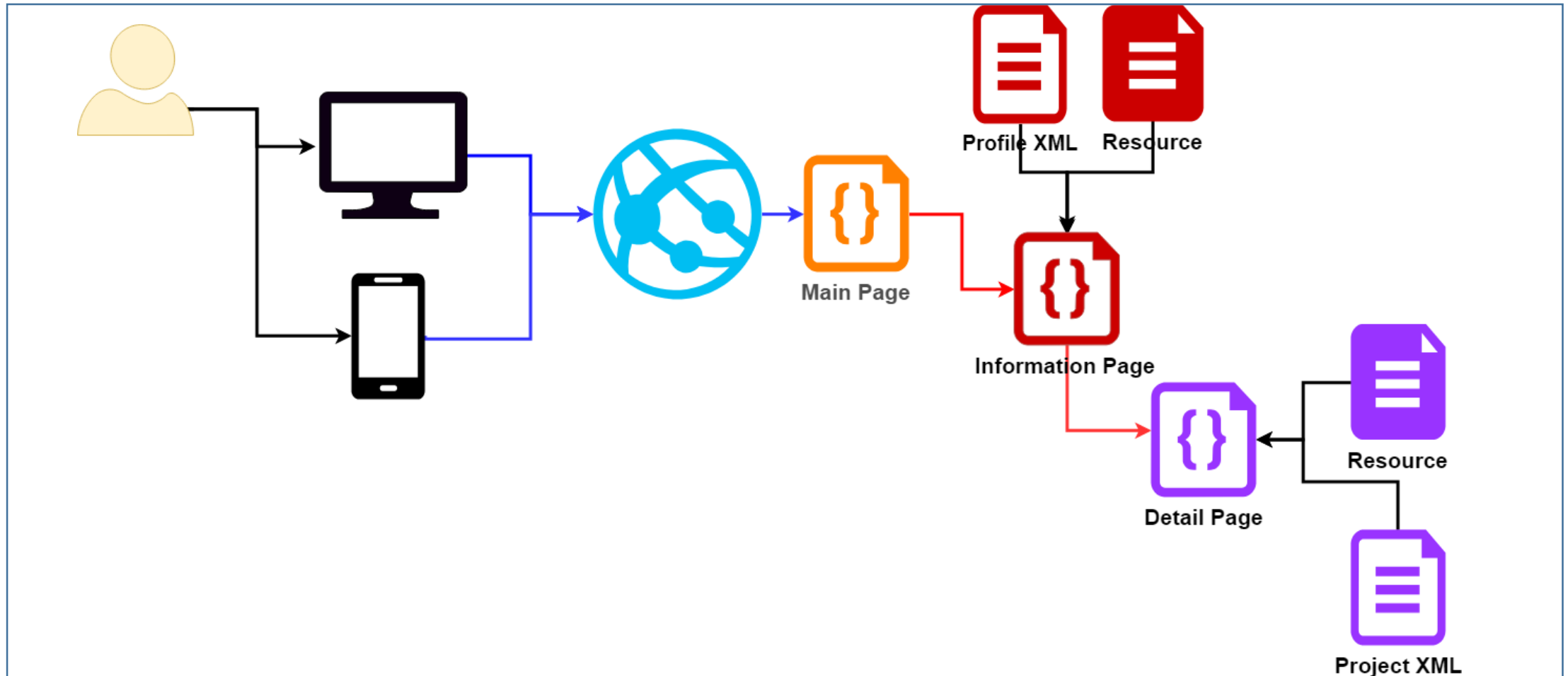
- 프로필 사진을 추가하자
- 사진 저장 후 View에서 파일의 경로를 통해 사진을 읽어오자

### 2. 프로젝트가 추가되면 어떻게 관리할 것인가?

- Project를 읽어올 수 있는 Class를 만들자
- 해당 Project Name을 통해 실제 View에서 볼 수 있게 하자

# 블로그를 만든다.


## 전체 구조도



# 블로그를 만든다.

포트폴리오 - 김대희

← → ↺ 🔒 https://teamimpact.azurewebsites.net/Home/Information/Daehee 🔍 ☆ JB NEW V YouTube Android ABP 1



서경대학교 컴퓨터공학과 2018.02 졸업  
2016.06.29 ~ 2016.08.16 : JavaFX Project  
2017.07.01 ~ 2017.08.15 : Spring Blog Site  
2017.06.28 ~ 2017.09.06 : DA설계 공모전


개발자는 항상 책임감을 가지고 재미를 추구하면서, 사용자가 편리하게 이용하고 행복해  
질 수 있는 서비스를 제공해야 합니다. 이를 위해서는 대화와 협업은 필수적입니다. 이러한  
요건을 갖추기 위해 끊임없이 변화하고 있으며, 계속 노력하는 개발자입니다.

PROJECT

Spring Blog Site

Space War

DA공모전대회



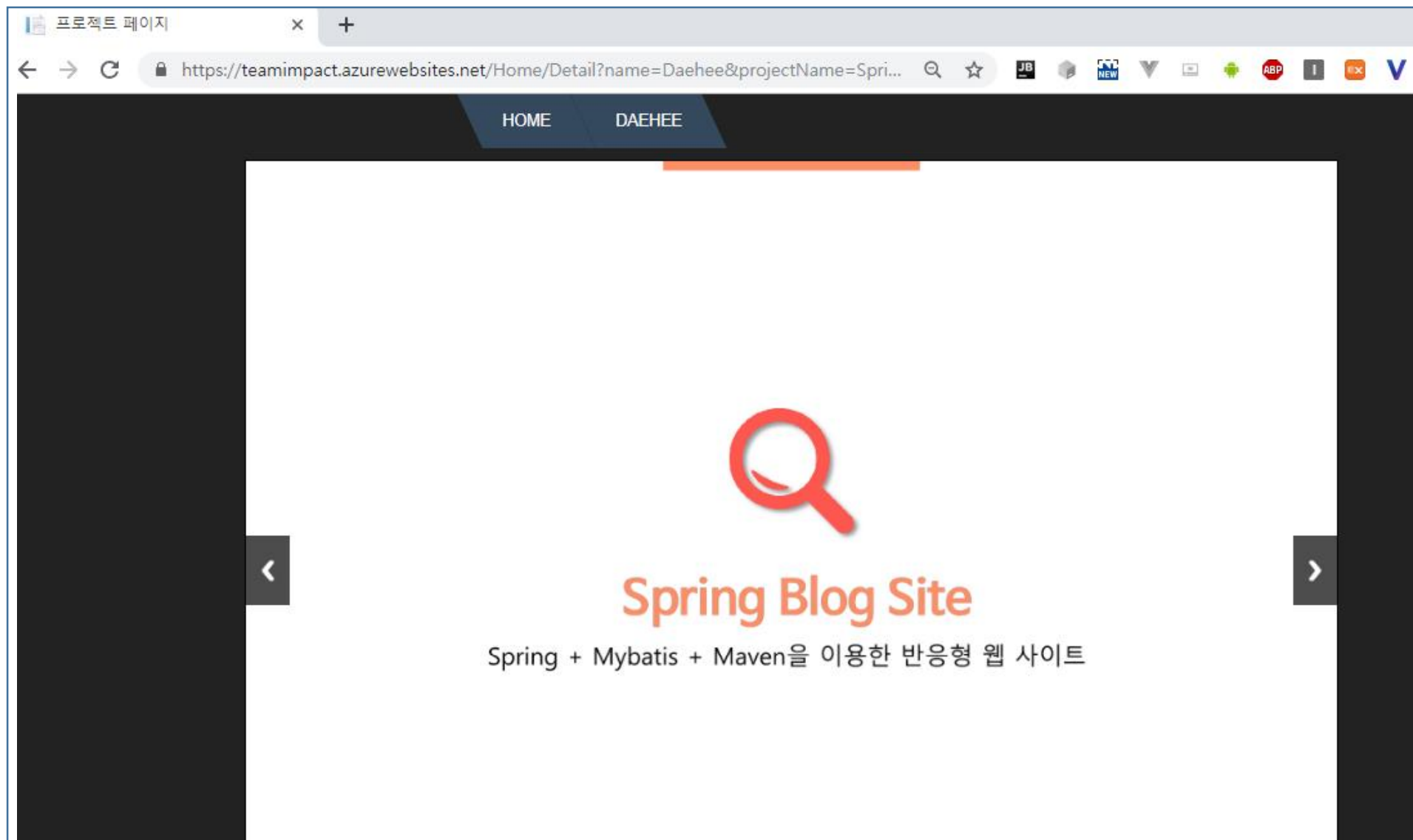
Spring Blog Site

Spring + Mybatis + Maven을 이용한 반응형 웹 사이트

기본적인 Spring Blog Site로서, 반응형 웹을 이용함과 동시에 모바일에도 편하게 사람들이 쓸 수 있는 웹사이트를 제작했습니다.

VIEW DETAIL

# 블로그를 만든다.



# 블로그를 만든다.

```
<Information>
<Name>
  <kr>김대희</kr>
  <en>Kim DaeHee</en>
</Name>
<Job>Programmer</Job>
<University>
  <Name>서경대학교</Name>
  <Major>컴퓨터공학과</Major>
  <GraduationDate>2018.02 졸업</GraduationDate>
</University>
<Achievements>
  <Achievement>2016.06.29 ~ 2016.08.16 : JavaFX Project </Achievement>
  <Achievement>2017.07.01 ~ 2017.08.15 : Spring Blog Site </Achievement>
  <Achievement>2017.06.28 ~ 2017.09.06 : DA설계공모전</Achievement>
  <Achievement>주력 언어 : Spring Framework, Classic ASP, PHP, ASP.NET, Node.js</Achievement>
  <Achievement>DataBase : MariaDB, Mysql, MSSQL, ERWIN, DA#, eXERD</Achievement>
  <!-- 필요한 만큼 추가 -->
</Achievements>
<!-- 필요한 만큼 추가 -->
<Email>anyozcall@gmail.com</Email>
<Comment>
  개발자는 항상 책임감을 가지고 재미를 추구하면서, 사용자가 편리하게 이용하고 행복해 질
  이를 위해서는 대화와 협업은 필수적입니다. 이러한 요건을 갖추기 위해 끊임없이 변화하고
</Comment>
```

```
namespace PortfolioWeb.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult Information(string id)
        {
            string path = Server.MapPath("~/App_Data/{id}/metadata.xml");

            if (!System.IO.File.Exists(path))
            {
                return View(new ProfileInfo(null, null));
            }

            ProfileInfo profile = new ProfileInfo(id, path);
            try
            {
                // Github, LinkedIn Image 추가
                ViewBag.ID = id;
                ViewBag.ProfileImage = ImageUtil.ToBase64String(Server.MapPath("~/App_Data/{id}/profile.jpg"));
            }
            catch (Exception)
            {
                ViewBag.ProfileImage = string.Empty;
            }

            return View(profile);
        }
    }
}
```

# 블로그를 만든다.

```
public class ProjectInfo
{
    public string DirName { get; private set; }
    public string Name { get; private set; }
    public string Topic { get; private set; }
    public string Comment { get; private set; }
    public string Thumbnail { get; private set; }

    public ProjectInfo(string dirName, string name, string topic,
    {
        this.DirName = dirName;
        this.Name = name;
        this.Topic = topic;
        this.Comment = comment;

        FileInfo fileInfo = new FileInfo(metadataPath);

        // ~/Resources/User
        DirectoryInfo dirInfo = fileInfo.Directory;

        // ~/Resources/User/ProjectDir/0.jpg
        var coverImagePath = Path.Combine(dirName, "tn.jpg");

        try
        {
            // ProjectDir/tn.jpg
            this.Thumbnail = coverImagePath;
        }
        catch (Exception)
        {
            this.Thumbnail = string.Empty;
        }
    }
}
```

```
public ActionResult Information(string id)
{
    string path = Server.MapPath($"~/App_Data/{id}/metadata.xml");

    if (!System.IO.File.Exists(path))
    {
        return View(new ProfileInfo(null, null));
    }

    ProfileInfo profile = new ProfileInfo(id, path);
    try
    {
        // Github, LinkedIn Image 추가
        ViewBag.ID = id;
        ViewBag.ProfileImage = ImageUtil.ToBase64String(Server.MapPath($"~/App_Data/{id}/profile.jpg"));
    }
    catch (Exception)
    {
        ViewBag.ProfileImage = string.Empty;
    }


    return View(profile);
}
```



# 블로그를 만든다.

포트폴리오 - 김대희

https://teamimpact.azurewebsites.net/Home/Information/Daehee



서경대학교 컴퓨터공학과 2018.02 졸업

2016.06.29 ~ 2016.08.16 : JavaFX Project

2017.07.01 ~ 2017.08.15 : Spring Blog Site

2017.06.28 ~ 2017.09.06 : DA설계 공모전


개발자는 항상 책임감을 가지고 재미를 추구하면서, 사용자가 편리하게 이용하고 행복해 질 수 있는 서비스를 제공해야 합니다. 이를 위해서는 대화와 협업은 필수적입니다. 이러한 요건을 갖추기 위해 끊임없이 변화하고 있으며, 계속 노력하는 개발자입니다.

PROJECT

Spring Blog Site

Space War

DA공모전대회



Spring Blog Site

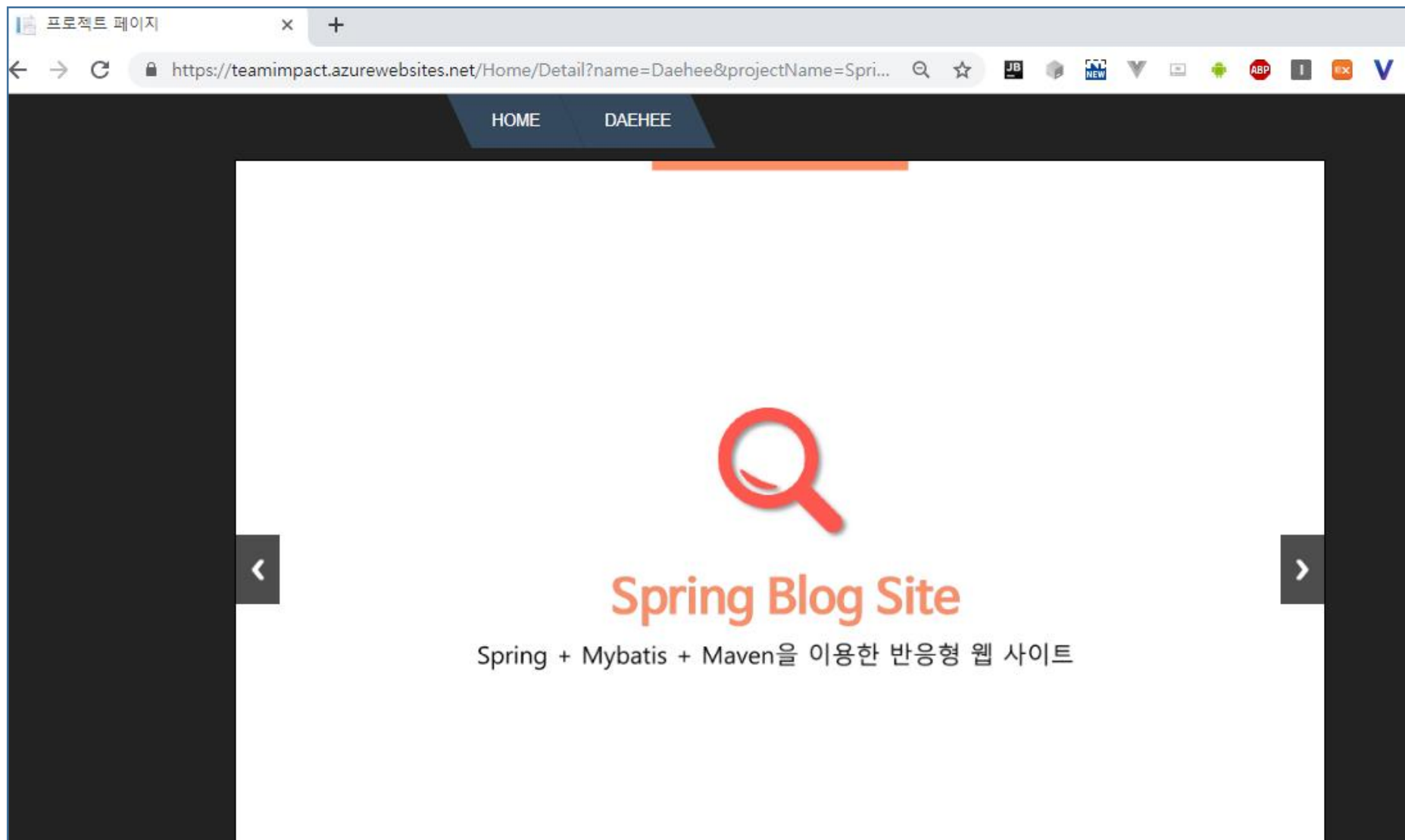
Spring + Mybatis + Maven을 이용한 반응형 웹 사이트

기본적인 Spring Blog Site로서, 반응형 웹을 이용함과 동시에 모바일에도 편하게 사람들이 쓸 수 있는 웹사이트를 제작했습니다.

VIEW DETAIL

# 블로그를 만든다.

---



# 블로그를 만든다.

---

# 원하는 결과물이 나온다 !

# 블로그를 만든다.

---

## 블로그 첫 동작으로 느낀 점

### 1. 결과물이 보임

- XML으로 이루어진 것들이 페이지에서 완성된 정보로 나옴

### 2. 함께 간단한 블로그를 만드는 과정 자체가 재밌었음

- 함께 만들면서 어떻게 협업을 할 지 알 수 있었음

### 3. 서로의 코딩 스타일을 알고 리뷰하는 과정을 배울 수 있었음

- A는 A 방법대로 코딩을 하고 , B는 B 방법대로 코딩을 하고, 합치는 과정은 A,B도 아닌 C 방식으로 하고..

# 블로그를 만든다.

---

## 그렇게

# 블로그를 만든다.

---

## 첫 블로그 개발 끝

# 2017 ~ 2018

---

## 1. 2017.7 함께 제작 완료

- XML Parsing
- ASP.NET MVC

## 2. 2017 ~ 2018

- XML 부분만 수정

2년 후



# 발표 제의가 왔다.

# 단순한 생각

# 단순한 생각

---

## 1. 기존의 XML 파싱 방법만 하면 되겠지

- XML Parsing
- DB 연결 없음

## 2. ASP.NET MVC 5유지

- 여전히 ASP.NET MVC 5로 유지
- .NET Core 모름

# 그러나

**너무나도 달라진..**

# .NET의 변화

---

## 1. ASP.NET MVC -> .NET Core ( Linux, Mac ..)

- ASP.NET MVC 대신 .NET Core가 새롭게 쓰이고 있음
- .NET Core는 멀티 플랫폼 지원
- 노트북이 멀티 부팅이었기 때문에 쓰기 편했음

# .NET의 변화

---

## 2. Open Source

- Close Source로 생각 했었음
- .NET Core는 Open Source...
- 어떻게 구성되는지 볼 수 있었음

# .NET의 변화

---

## 3. VS Code

- VS 외에도 VS Code로 작업가능..
- Windows 외에 다른 OS로 작업할 때 VSCode를 쓰기가 편했음



# 변화를 따라가보자!

# 블로그 리뉴얼 도전

무엇을 바꾸고 싶은가?

# 블로그 리뉴얼 도전

---

## 블로그 리뉴얼 시 바꾸고 싶던 것

### 1. .NET Core로 변경

- 새로운 변화인 .NET Core로 작업을 해보자

### 2. Profile 수정 및 추가

- 개발자인 만큼 GitHub, Linkedin 등 나를 표현할 수 있는 사이트 추가

### 3. 세부 내용 출력 시간 개선

- 느린 속도를 빠르게 하고 싶었음

### 4. DATABASE 연동

- 2019년 초까지 DATABASE 연동 없이 XML로만 운영
- 이번에는 데이터베이스를 써보자

# 블로그 리뉴얼 도전

---

**고쳐보자 !**

라고 했으나 지옥의 시작

## .NET Core로 변경

# .NET Core로 변경

---

## . ASP.MVC5 -> .NET Core

### 1. .NET Core의 Controller 호출 방법은 어떻게 되어있을까?

- 기존의 MVC 5는 ActionResult를 통해 URL을 호출
- Net Core 프로젝트 생성시 IActionResult로 되어있었음
- Action Result -> IActionResult로 바꾸어 보자

### 2. 해당 리소스와 자원들은 어떻게 옮기지?

- .NET Core는 wwwroot 에 정적 파일을 모두 넣음
- App\_Data / Resource => wwwroot로 옮기자



# .NET Core로 변경

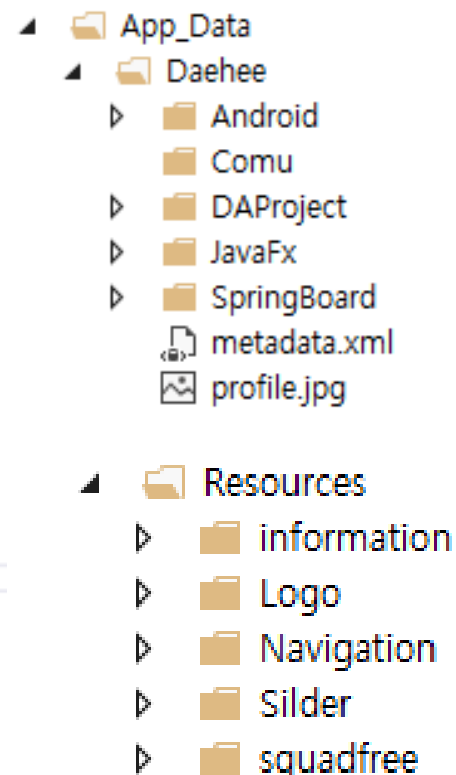
## As-Is

```
public ActionResult Information(string id)
{
    string path = Server.MapPath("~/App_Data/{id}/metadata.xml");

    if (!System.IO.File.Exists(path))
    {
        return View(new ProfileInfo(null, null));
    }

    ProfileInfo profile = new ProfileInfo(id, path);
    try
    {
        // Github, LinkedIn Image 추가
        ViewBag.ID = id;
        ViewBag.ProfileImage = ImageUtil.ToBase64String(Server.MapPath("~/App_Data/{id}/profile.jpg"));
    }
    catch (Exception)
    {
        ViewBag.ProfileImage = string.Empty;
    }

    return View(profile);
}
```



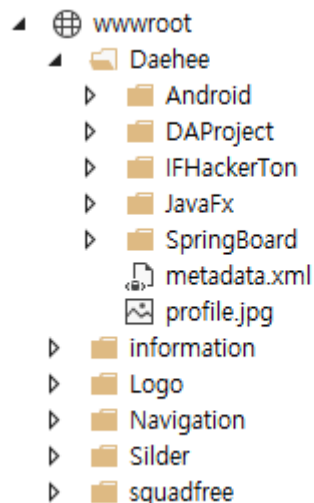
# .NET Core로 변경

## To-be

```
// 블로그 리스트 페이지
// URL : /Blog/Index
public IActionResult Index(int? page)
{
    // Page가 없거나 첫 페이지보다 작다면 Page 번호를 0으로 설정
    if(page == null || page < 0)
    {
        page = 0;
    }
    // 시작 글, 끝 글 설정
    startRange = (int)page * range;

    // 기준은 10페이지로 하나 글의 숫자가 10개보다 적을 경우 남은 글 숫자만큼만 출력
    // EX) 26개인데 20번째 글부터 보여준다면 7개만 보여줌.
    blogCount = _BlogContext.Blog.ToList<Blog>().Count - startRange;


    // 마지막 페이지 고정
    // 마지막 페이지보다 더 뒤로 가는 Index로 갈 경우 마지막 글이 있는 Index Page로 고정
    if(blogCount < 0)
    {
        page -= 1; // 마지막 Index로
        startRange = (int)page * range; // 한 페이지 보여줄 글 수
        blogCount = _BlogContext.Blog.ToList<Blog>().Count - startRange; // 시작 Index 지정
    }
}
```



# .NET Core로 변경

Kim Dae Hee Blog

https://daeheekim.azurewebsites.net/Home/Information/Daehee



## Programmer

김대희

Kim DaeHee

Email: [anyozcall@gmail.com](mailto:anyozcall@gmail.com)

---

서경대학교 컴퓨터공학과 2018.02 졸업  
2016.06.29 ~ 2016.08.16 : JavaFX Project  
2017.07.01 ~ 2017.08.15 : Spring Blog Site  
2017.06.28 ~ 2017.09.06 : DA설계 공모전  
2019.01.23 ~ 2019.02.01 : D-TECH 공모전  
주력 언어 : Spring Framework, Classic ASP, PHP, ASP.NET, Node.js  
DataBase : MariaDB, Mysql, MSSQL, ERWIN, DA#, eXERD

---

경력

2017.12.18 ~ : 비상교육 IT전략과

---

수상

2017.09.01 ~ 2017.10.31 - 서울시 앱 공모전 2017 : 예선당선작  
2018.08.24 ~ 2018.08.25 - 웹컴 투 메이커톤 2회 : 우수상  
2018.11.10 ~ 2018.11.11 - 2018 사회혁신 이프 해커톤 : 소셜임팩트 상 ( 2등 )

---

개발자는 항상 책임감을 가지고 재미를 추구하면서, 사용자가 편리하게 이용하고 행복해  
질 수 있는 서비스를 제공해야 합니다. 이를 위해서는 대화와 협업은 필수적입니다. 이러한  
요건을 갖추기 위해 끊임없이 변화하고 있으며, 계속 노력하는 개발자입니다.


---

PROJECT

Top

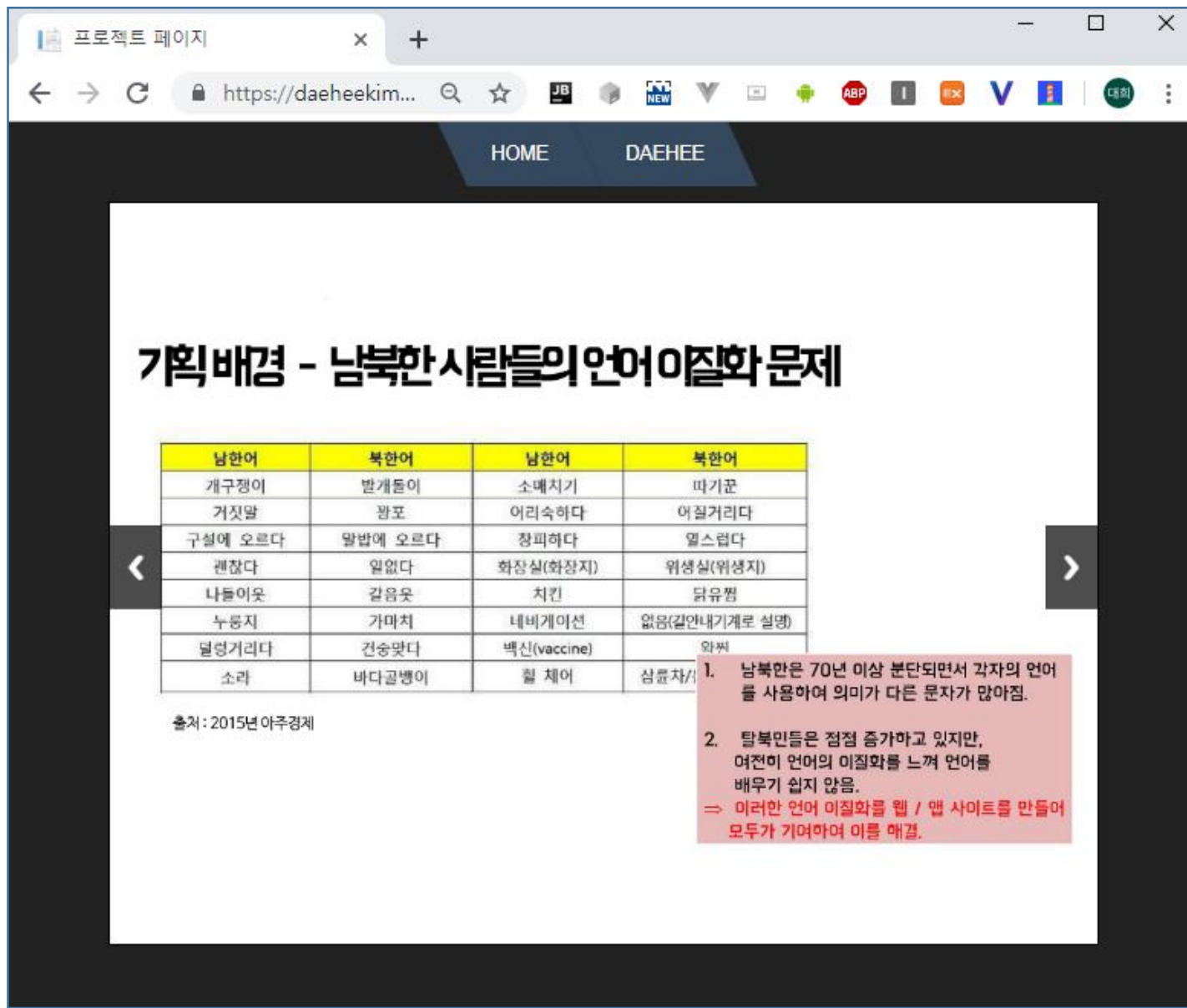
Home

Spring Blog Site



Spring Blog Site

# .NET Core로 변경



# 블로그 리뉴얼 도전

---

**잘 나온다 !**

# 블로그 리뉴얼 도전

## 기존의 블로그 vs 리뉴얼 된 블로그

	기존의 블로그 (ASP.NET MVC 5)	리뉴얼 블로그 (.NET Core 2.X )
URL 호출 Controller	ActionResult	ActionResult
작업환경	Windows	Windows, Ubuntu
공통 Layout	X	_ViewStart + _Layout (.cshtml)
리소스관리	Resource/App_DATA	Wwwroot
공통 모델	X	_ViewImports (.cshtml)

# 블로그 리뉴얼 도전

---

## 이관 성공

# 블로그 리뉴얼 도전

---

**끝!**



# 블로그 리뉴얼 도전

---

이면 좋겠지만...

아직 고칠 것이 많다

## Profile 수정 및 추가

# Profile 수정 및 추가

---

## 추가시키고 싶던 것

### 1. 작업했던 Github 등의 사이트를 보여주고 싶었음

- 기존에는 프로필 + 프로젝트 + 글만 존재했음
- 홈페이지를 통해 나와 관련된 사이트들을 보여주고 싶었음

### 2. 글씨가 아닌 아이콘으로 보여주었으면 좋겠다고 생각

- Github 글씨를 눌러서 이동하는 것 대신 아이콘을 눌러서 이동하고 싶었음


# Profile 수정 및 추가

```
// 해당되는 경로와, 이름을 통해 원하는 XML 경로의 파일을 가져와 ProfileInfo Object에 Mapping
ProfileInfo profile = new ProfileInfo(id, path);
try
{
    // wwwroot + 경로 가져오기 ( static으로도 할 수 있으나 그냥 뿌림 )
    String ProfileImageRoute = "/" + id + "/" + "profile.jpg";
    String GithubImageRoute = "/Logo/github.png";
    String LinkedInImageRoute = "/Logo/linkedin.png";

    // Github, LinkedIn Image 추가
    ViewBag.ID = id;
    ViewBag.ProfileImage = ProfileImageRoute;
    ViewBag.GithubImage = GithubImageRoute;
    ViewBag.LinkedinImage = LinkedInImageRoute;
}
catch (Exception)
{
    ViewBag.ProfileImage = string.Empty;
    ViewBag.GithubImage = string.Empty;
    ViewBag.LinkedinImage = string.Empty;
}
```

```
<!-- Github , LinkedIn Image 형태로 변경 -->
<div class="row">
    <div class="col-md-2">
        <a href="@Model.GithubLink">
            <!-- 절대 경로에서 상대 경로로 되도록 변경 -->
            @{ string githubImageUrl = Context.Request.Scheme + "://" + Context
            
        <a href="@Model.LinkedinLink">
            <!-- 절대 경로에서 상대 경로로 되도록 변경 -->
            @{ string linkedinImageUrl = Context.Request.Scheme + "://" + Context
            <img class="img-fluid rounded" src="@((string.IsNullOrEmpty(ViewBag.I
            </a>
        </div>
    </div>
```

# Profile 수정 및 추가



CULTURE POSTMO COMPLEX DERNISM

Email: [anyozcall@gmail.com](mailto:anyozcall@gmail.com)

---

서경대학교 컴퓨터공학과 2018.02 졸업  
2016.06.29 ~ 2016.08.16 : JavaFX Project  
2017.07.01 ~ 2017.08.15 : Spring Blog Site  
2017.06.28 ~ 2017.09.06 : DA설계 공모전  
2019.01.23 ~ 2019.02.01 : D-TECH 공모전  
주력 언어 : Spring Framework, Classic ASP, PHP, ASP.NET , Node.js  
DataBase : MariaDB , Mysql, MSSQL , ERWIN , DA# , eXERD

---

**경력**  
2017.12.18 ~ : 비상교육 IT전략과



---

**수상**  
2017.09.01 ~ 2017.10.31 - 서울시 앱 공모전 2017 : 예선당선작  
2018.08.24 ~ 2018.08.25 - 웰컴 투 메이커톤 2회 : 우수상  
2018.11.10 ~ 2018.11.11 - 2018 사회혁신 이프 해커톤 : 소셜임팩트 상 (2등)


---

개발자는 항상 책임감을 가지고 재미를 추구하면서, 사용자가 편리하게 이용하고 행복해질 수 있는 서비스를 제공해야 합니다. 이를 위해서는 대화와 협업은 필수적입니다. 이러한 요건을 갖추기 위해 끊임없이 변화하고 있으며, 계속 노력하는 개발자입니다.

---




# Profile 수정 및 추가



**DaeHee Kim**  
Back-End Developer  
대한민국

프로필 항목 등

열심히 하고자 하



free

**Kim Dae Hee**  
DaeHeeKim93

항상 꿈꾸며 발전하는 개발자가 되고,  
그 기술로 사회가 행복해지길 바랍니다.  
또한 꾸준히 자신을 성장시키기  
위해 노력을 진행중인 개발자입니다.

Visang Education  
Seoul, Korea  
anyozcall@gmail.com  
https://daeheekim.azurewebsitesit...

Edit

Overview Repositories 6 Projects 0 Stars 32 Followers 55 Following 32

Popular repositories

**DaeHeeKim-Review**

개발자의 면접 준비를 위해 개념 및 복습을 위한 이론 및 코드입니다.

Java ★ 230 46

**PortFolio**

포트폴리오 목록 및 정리

Java ★ 2

**Study**

공부한거 정리해서 아무거나 올림.

Dockerfile ★ 1

**deploy-test-project**

Forked from gnu-gnu/deploy-test-project

demo project for lecture of deployment

Java

125 contributions in the last year

Contribution settings

2019

2018

2017

2016

Learn how we count contributions.

Less More

# Profile 수정 및 추가

---

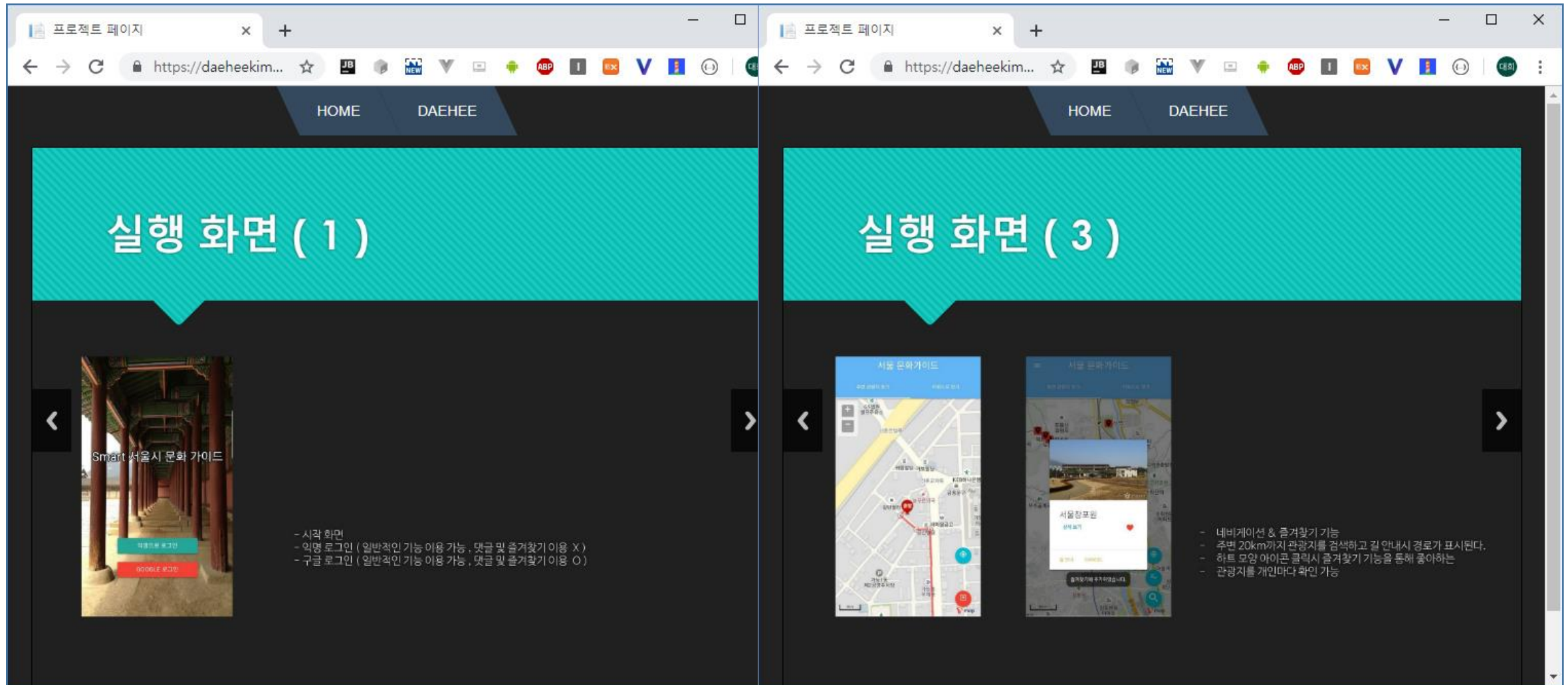
## 기존의 블로그 vs 리뉴얼 된 블로그

	기존의 블로그 (ASP.NET MVC 5)	리뉴얼 블로그 (.NET Core 2.X )
사용 Class	Profile	Profile Github Linkedin



## 세부 내용 출력 시간 개선

# 세부 내용 출력 시간 개선



# 세부 내용 출력 시간 개선

---

## 기존 구조의 문제점과 고민

### 1. 세부 페이지 실행만 10초 이상이 걸림

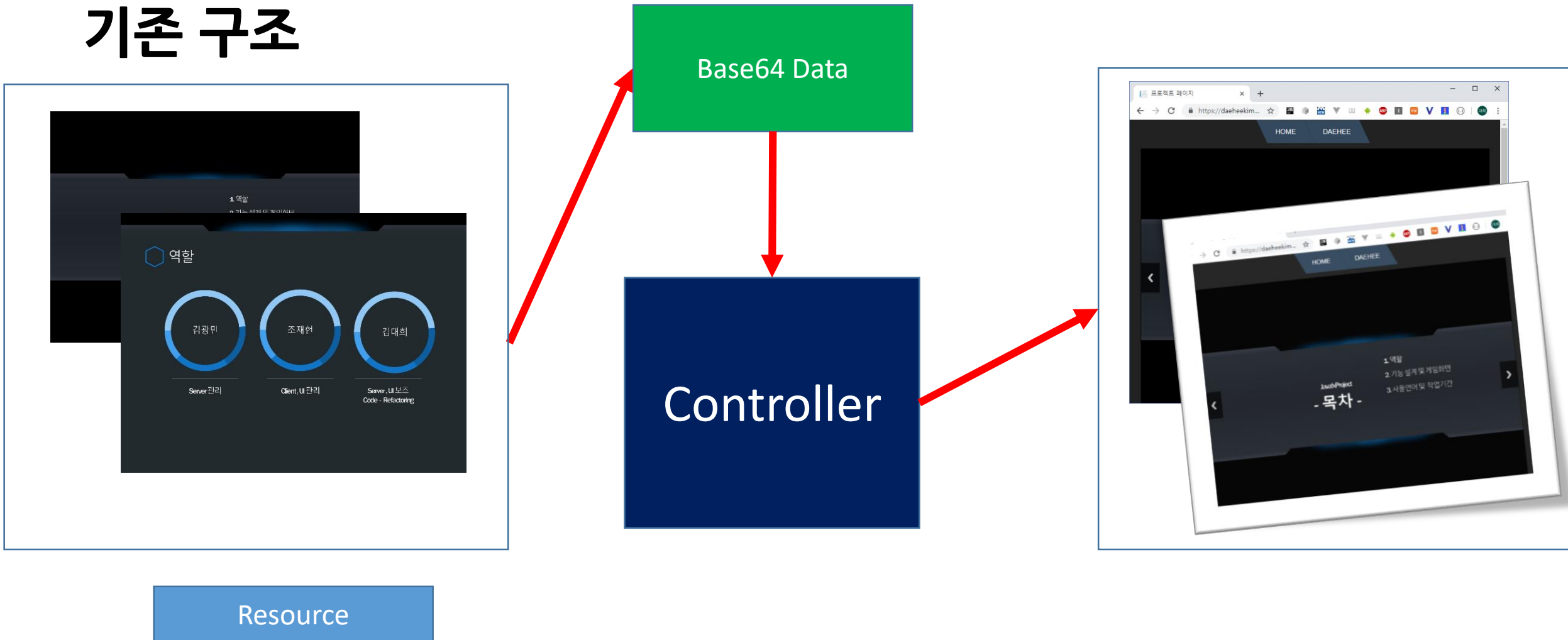
- 슬라이드 양이 많을 수록 한번에 정보를 가져오는 양이 많아져 속도가 느림
- 모든 사진의 대한 정보들이 실행되기 전까지 슬라이드를 움직일 수 없었음

### 2. 왜 속도가 느릴까?

- 대학교 시절에 ASP.NET MVC 리소스 구조에 대해 몰랐음
- **APP\_DATA는 경로를 통해 접근할 수 없다는 것을 모르고 이름만 보고 넣음**
- 당시에 경로로 읽어올 수 없어서 Base64 파일로 급하게 뺐음
- **이 과정 때문에 속도가 느림**


# 세부 내용 출력 시간 개선

## 기존 구조



# 세부 내용 출력 시간 개선

HOME DAEHEE YONGSEOK



## Spring Blog Site

Spring + Mybatis + Maven을 이용한 반응형 웹 사이트

김대희

Elements Console Sources Network

```
<!doctype html>
<html lang="kr" class="-webkit-">
  <head>...</head>
  <body style="background-color:#222">
    <div class="container">
      ::before
      <div class="row">...</div>
      <div class="row">
        ::before
        <div class="col-md-1"></div>
        <div class="col-md-10">
          <div class="callbacks_container">
            <ul class="rslides callbacks callbacks1" id="slider4" box-shadow:"10px 10px 5px #fff">
              <li id="callbacks1_s0" class="slide callbacks1_on" style="float: left; position: relative; opacity: 1; z-index: 2; transition: opacity 500ms ease-in-out 0s;">
                 == $0
                <a class="prev" href="#no-js-slider16"></a>
                <a class="next" href="#no-js-slider1"></a>
              </li>
              <li id="callbacks1_s1" class="slide" style="float: none; position: absolute; opacity: 0; z-index: 1; display: list-item; transition: opacity 500ms ease-in-out 0s;">...</li>
              <li id="callbacks1_s2" class="slide" style="float: none; position: absolute; opacity: 0; z-index: 1; display: list-item; transition: opacity 500ms ease-in-out 0s;">...</li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

html body div div div div #slider4 li#callbacks1\_s0.slide.callbacks1\_on img

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls + ition 0

element style {

# 세부 내용 출력 시간 개선

---

너무 느리다

# 세부 내용 출력 시간 개선

---

**잘못된 것을 고치자**

# 세부 내용 출력 시간 개선

---

## 어떻게 고칠까?

### 1. Base64 형태에서 경로를 통해 이미지를 읽어오자

- Base64 형태가 아닌 경로로 읽어와서 출력해야 함
- APP\_DATA 처럼 DB에 연동하는 부분이 아닌 리소스 폴더에 저장하자

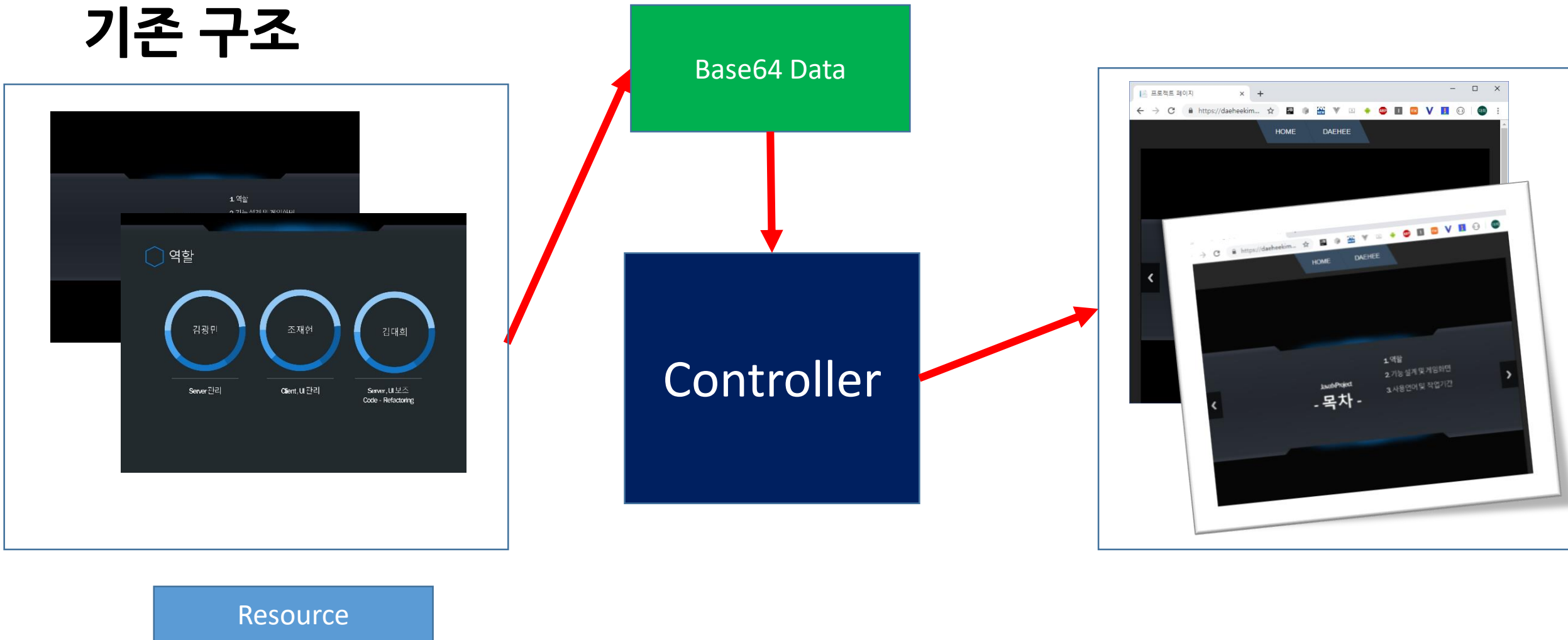
### 2. 리소스를 일원화하자

- 대학생때는 APP\_DATA / Resource에 따로따로 나누고 있었음
- 이번에는 리소스 전체를 wwwroot라는 정적 파일 저장소로 합치자



# 세부 내용 출력 시간 개선

## 기존 구조



# 세부 내용 출력 시간 개선

## As-Is

```
ProfileInfo profile = new ProfileInfo(id, path);
try
{
    // Github, Linkedin Image 추가
    ViewBag.ID = id;
    ViewBag.ProfileImage = ImageUtil.ToBase64String(Server.MapPath($"~/App_Data/{id}/profile.jpg"));
    ViewBag.GithubImage = ImageUtil.ToBase64String(Server.MapPath($"~/Resources/Logo/github.png"));
    ViewBag.LinkedinImage = ImageUtil.ToBase64String(Server.MapPath($"~/Resources/Logo/linkedin.png"));
}
catch (Exception)
{
    ViewBag.ProfileImage = string.Empty;
    ViewBag.GithubImage = string.Empty;
    ViewBag.LinkedinImage = string.Empty;
}

return View(profile);
}

public ActionResult Detail(string name, string projectName)
{
    String route = "~/App_Data/" + name + "/" + projectName;
    String picture = Server.MapPath(route);
    System.IO.DirectoryInfo di = new System.IO.DirectoryInfo(picture);
    int filelength = di.GetFiles().Length - 1;
    ViewBag.Route = picture;
    ViewBag.Picture = filelength;
    return View();
}
}
```

# 세부 내용 출력 시간 개선

## To-be

```
// wwwroot + 경로 가져오기 ( static으로도 할 수 있으나 그냥 버림 )
String ProfileImageRoute = "/" + id + "/" + "profile.jpg";
String GithubImageRoute = "/Logo/github.png";
String LinkedInImageRoute = "/Logo/linkedin.png";

// Github, LinkedIn Image 추가
ViewBag.ID = id;
ViewBag.ProfileImage = ProfileImageRoute;
ViewBag.GithubImage = GithubImageRoute;
ViewBag.LinkedinImage = LinkedInImageRoute;
}
catch (Exception)
{
    ViewBag.ProfileImage = string.Empty;
    ViewBag.GithubImage = string.Empty;
    ViewBag.LinkedinImage = string.Empty;
}

return View(profile);
}

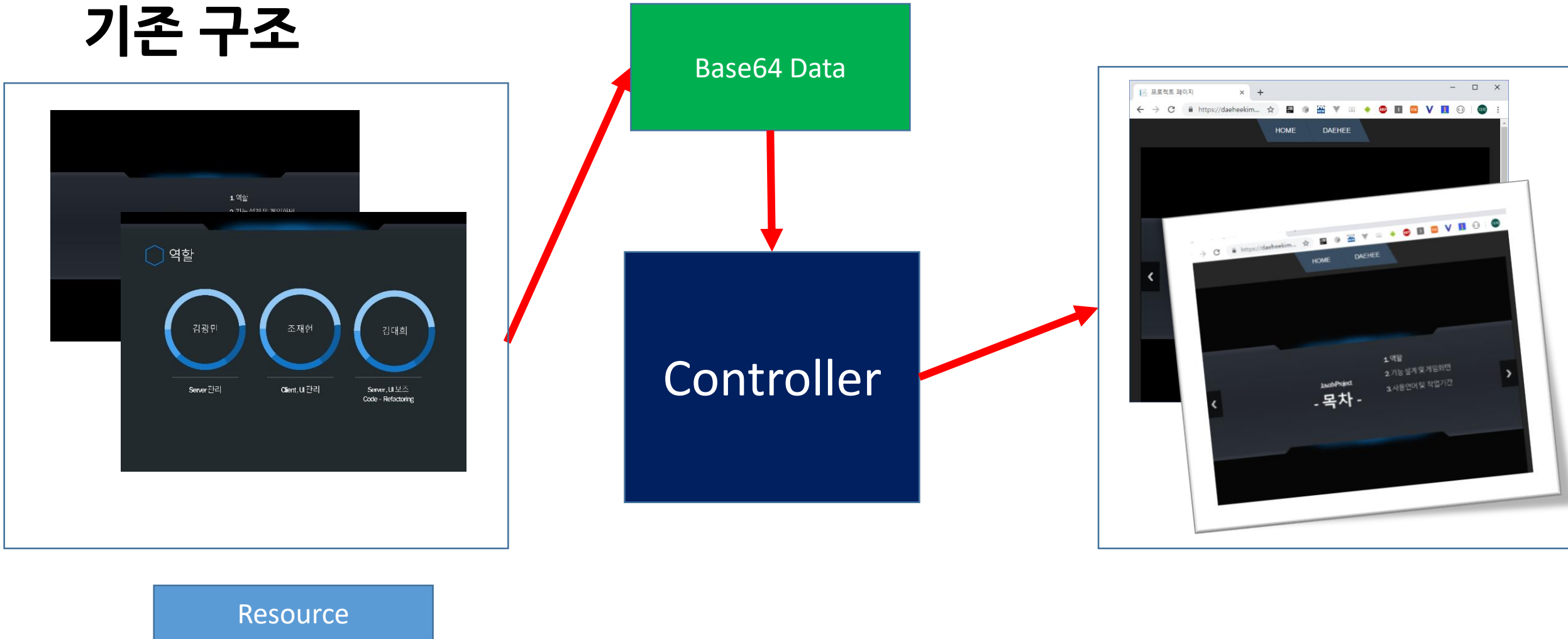
// 세부 : 프로젝트 확인
public IActionResult Detail(string name, string projectName)
{
    // 그림을 가져오기 위한 Web Root Path 설정
    string webRootPath = _hostingEnvironment.WebRootPath;

    // 그림 정보를 가져와서 뿌려주기
    String route = "/" + name + "/" + projectName;
    String picture = webRootPath + route;

    // 해당 경로에 있는 파일을 읽어와서 숫자에 따른 사진 갯수와, 경로를 반환
    System.IO.DirectoryInfo di = new System.IO.DirectoryInfo(picture);
    int filelength = di.GetFiles().Length - 1;
    ViewBag.Route = route;
    ViewBag.Picture = filelength;
    return View();
}
```

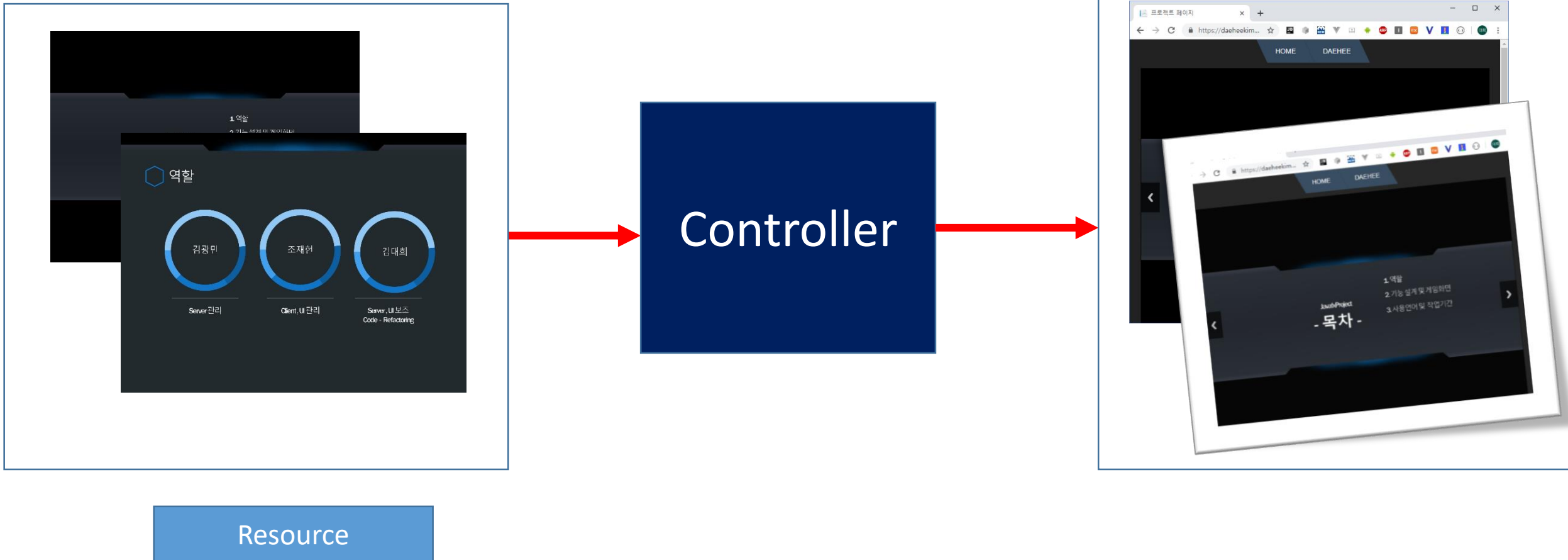
# 세부 내용 출력 시간 개선

## 기존 구조




# 세부 내용 출력 시간 개선

## 개선 구조



# 세부 내용 출력 시간 개선

HOME DAEHEE YONGSEOK



## Spring Blog Site

Spring + Mybatis + Maven을 이용한 반응형 웹 사이트

김대희

Elements Console Sources Network

```
<!doctype html>
<html lang="kr" class="-webkit-">
  <head>...</head>
  <body style="background-color:#222">
    <div class="container">
      ::before
      <div class="row">...</div>
      <div class="row">
        ::before
        <div class="col-md-1"></div>
        <div class="col-md-10">
          <div class="callbacks_container">
            <ul class="rslides callbacks callbacks1" id="slider4" box-shadow:"10px 10px 5px #fff">
              <li id="callbacks1_s0" class="slide callbacks1_on" style="float: left; position: relative; opacity: 1; z-index: 2; transition: opacity 500ms ease-in-out 0s;">
                 == $0
                <a class="prev" href="#no-js-slider16"></a>
                <a class="next" href="#no-js-slider1"></a>
              </li>
              <li id="callbacks1_s1" class="slide" style="float: none; position: absolute; opacity: 0; z-index: 1; display: list-item; transition: opacity 500ms ease-in-out 0s;">...</li>
              <li id="callbacks1_s2" class="slide" style="float: none; position: absolute; opacity: 0; z-index: 1; display: list-item; transition: opacity 500ms ease-in-out 0s;">...</li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

html body div div div div #slider4 li#callbacks1\_s0.slide.callbacks1\_on img


Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

element style {

# 세부 내용 출력 시간 개선

HOME DAEHEE



## Spring Blog Site

Spring + Mybatis + Maven을 이용한 반응형

김대희

Elements Console Sources Network Performance

```
<!doctype html>
<html lang="kr" class="-webkit">
  <head>...</head>
  <body style="background-color:#222">
    <div class="container">
      ::before
      <div class="row">...</div>
      <div class="row">
        ::before
        <div class="col-md-1"></div>
        <div class="col-md-10">
          <div class="callbacks_container">
            <ul class="rslides callbacks callbacks1" id="slider4" box-shadow:"10px 10px 5px #fff">
              <!-- 해당되는 경로의 사진을 보여주고, 사진 갯수에 따라 Silder 작
              ->
              <li id="callbacks1_s0" class="slide callbacks1_on" style="float: left; position: relative; opacity: 1; z-index: 2; transition: opacity 500ms ease-in-out 0s;">
                 == $0
                <a class="prev" href="#no-js-slider16"></a>
                <a class="next" href="#no-js-slider1"></a>
              </li>
              <li id="callbacks1_s1" class="slide" style="float: none; position: absolute; opacity: 0; z-index: 1; display: list-item; transition: opacity 500ms ease-in-out 0s;">...</li>
              <li id="callbacks1_s2" class="slide" style="float: none;
```

html body div div div div #slider4 li#callbacks1\_s0.slide.callbacks1\_on img

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls + position 0

element style {

Console What's New

Highlights from the Chrome 73 update

# 세부 내용 출력 시간 개선

---

**속도가 빨라졌다**



# 세부 내용 출력 시간 개선

## 기존의 블로그 vs 리뉴얼 된 블로그

	기존의 블로그 (ASP.NET MVC 5)	리뉴얼 블로그 (.NET Core 2.X )
이미지 로드	Base64 Data	Directory
사용 Class	ImageUtil.ToBase64String	X

# 세부 내용 출력 시간 개선

---

## 무엇이 개선되었을까?

### 1. 이미지를 불러오는 속도의 개선

- 기존의 Base64 페이지의 경우 10페이지의 슬라이드를 불러오는데 10초 이상
- 경로를 통해 불러왔을 때는 기존의 1/10 ~ 1/20 속도

### 2. 세부 내용 페이지 실행 속도의 개선

- 기존에는 모든 이미지가 불러와져야 페이지가 실행됨
- 수정 후에는 일부 이미지가 불러와지지 않아도 바로 페이지가 실행됨

## DATABASE 연동

# DATABASE 연동

---

## 1. 기존의 XML 유지할까?

- 해당 글을 XML 파일형식으로 일일이 불러오는 것이 알맞지 않음
- 수정할 때 XML 파일로 글을 일일이 작성할 것인가에 대한 문제
- 이미지, 영상 삽입, 수정 등을 XML로 글 중간에 넣을 것인가

## 2. DB를 연동할까?

- 실제 블로그처럼 자유롭게 CRUD로 글을 쓸 수 있음
- C#에서의 DB 연동은 처음이라 낯설
- SQL문을 그대로 쓸 것인가? ORM을 쓸 것인가?

# DATABASE 연동

---

결정은?

## 도입하자

# DATABASE 연동

---

## 어떻게 DB를 사용할 것인가?

### 1. CRUD 및 페이지 작업은 하고 싶었음

- DB를 쓰는 만큼 블로그의 기본적인 기능은 작동시키고 싶었음
- 혼자서 글을 수정하고 테스트 하는 환경을 만들고 싶었음

### 2. SQL? ORM?

- 대학교 때 부터 지금까지 SQL문으로 대부분 CRUD 작업을 함
- **개인 홈페이지기 때문에 새로운 것을 해보고 싶어 ORM 사용**

# DATABASE 연동

```
// Blog Table Class
namespace DaeHeeKim.Models
{
    public class Blog
    {
        [Key] // Primary Key 설정
        public int BlogId { get; set; } // Type = "int" // ID 형식은 ***Id
        public string Title { get; set; } // Type = "varchar"
        public string Context { get; set; } // Type = "LONGTEXT"
        public DateTime Regdate { get; set; } // Type = "DATETIME"
        public string Thumbnail { get; set; } // Type = "VARCHAR"
    }
}
```

```
// DB 연동을 해주는 Class
namespace DaeHeeKim.Models
{
    public class BlogContext : DbContext
    {
        public BlogContext()
        {
        }

        public BlogContext(DbContextOptions<BlogContext> options) : base(options)
        {
        }

        public DbSet<Blog> Blog { get; set; } // Blog Table DbSet
    }
}
```

## ABOUT DETAIL BLOG

[번역글] React vs Angular : 둘 중 어떤 것이 당신의 프로젝트에 맞을까요?

### 소개

안녕하세요. Popit에서 글을 쓰고 있는 김대희라고 합니다. 백엔드 개발자로서, 프론트 개발자 분들이 쓰는 프레임워크 비교 기사를 보셨을텐데 이 글이 흥미롭고 재미있게 읽을 수 있어서 해당 글을 번역하게 되었습니다. 기사와 개발자들이 읽히는 부분이 많이 다를 수 있고, 다소간 메크업이 없더라도 많은 좋은 피드백 부탁드립니다. 원문글은 다음글에 참고하세요.

• React vs Angular Compared: Which One Suits Your Project Better?

React vs Angular : 둘 중 어떤 것이 당신의 프로젝트에 맞을까요?

세계에서 많이 쓰이는 두 종류의 Front-End 개발 도구들 어느것을 골라야 할지 고민하고 계십니까? 훌륭한 비교 분석이 될 수 있는 이 글을 읽어보시면 됩니다. 프로그래밍 세계에서 Angular와 React는 Front-End 개발자들에게 가장 널리 사용되는 JavaScript 프레임 워크 중 하나입니다. 더욱이, Angular와 React는 Stack Overflow Developer Survey 2018에서 조사한 모든 프로그래밍 언어에서 Node.js와 함께 소프트웨어 엔지니어가 가장 많이 사용하는 프레임워크입니다. 이러한 Front-End 프레임 워크는 모두 인기가 비슷하고 유사한 아키텍처를 사용하여 JavaScript를 기반으로 합니다. 그러면 이 두개의 프레임워크는 무엇이 다른가요? 지금부터 React와 Angular를 비교해 봅시다. 다음 단락에서 프레임 워크의 일반적인 특성을 살펴 보도록 하겠습니다. 추가로 React와 Angular의 비교가 더 필요하다면 [크로스 플랫폼 모바일 프레임워크 \(React Native 포함\)](#) 기사나 Angular와 다른 Front-End 프레임워크를 비교한 기사를 읽어보세요.

### Angular와 ReactJS에 대한 간단한 설명

Angular는 Google에서 만들고 관리하는 Front-End 프레임 워크이며 대부분의 코드 편집기에서 작성할 수 있습니다. Angular는 MEAN 스택의 구성원이며(A), MEAN 스택은 동적인 웹 사이트와 웹 애플리케이션을 구축할 때 사용되는 JavaScript를 중심으로 구성된 무료 오픈 소스 Tool Set입니다. MEAN 스택은 MongoDB (NoSQL 데이터베이스), Express.js (Backend 프레임워크), Angular 또는 AngularJS (Front-End 프레임 워크) 및 Node.js (서버 플랫폼)로 구성됩니다. Angular 프레임워크는 개발자들이 쉽게 단일 페이지 웹 애플리케이션(SPA)을 만들 수 있게 해줍니다. Angular가 처음 출시되었을 때 가장 중요한 장점은 HTML 기반 문서를 통해 인스턴트 프로토타입을 할 수 있다는 점이었습니다. 이 기사에서는 AngularJS(Angular 1)과 Angular(2부터 이후)의 차이점을 해결하기 위해 일반적으로 Angular 7이라고 하는 최신 버전의 Angular에 중점을 둘 것입니다. Angular는 Forbes, WhatsApp, Instagram, healthcare.gov, HBO, Nike 등에서 사용하고 있습니다. React.js는 동적 사용자 인터페이스를 만들기 위해 Facebook에서 2011년에 만든 오픈 소스 JavaScript 라이브러리입니다. React는 Front-End 개발을 위한 HTML 요소를 사용하는 JavaScript 및 JSX를 기반으로 합니다. React에는 모바일 앱 개발을 위한 별도의 Cross-Platform 프레임 워크인 React Native가 있습니다. 링크된 글에서 React.js와 React Native 둘 다에 대한 심도 있는 정보를 제공하고 있습니다. React는 Netflix, PayPal, Uber, Twitter, Udemy, Reddit, Airbnb, Walmart 등에서 사용됩니다.

### Toolset : Framework vs. Library

프레임워크 생태계는 실제 엔지니어링을 진행할 때(개발할 때) 밀접하게 영향을 줍니다. 여기에서는 Angular 및 React와 함께 일반적으로 사용되는 주요 도구들을 살펴 보겠습니다. 우선, React는 실제로 프레임워크가 아니라 라이브러리입니다. 이는 추가 개발 도구 및 라이브러리를 여러 번 통합해야 한다는 것을 의미합니다. Angular는 프레임워크이기 때문에 이미 앱을 만들기 시작할 때 개발 도구를 가지고 있습니다.

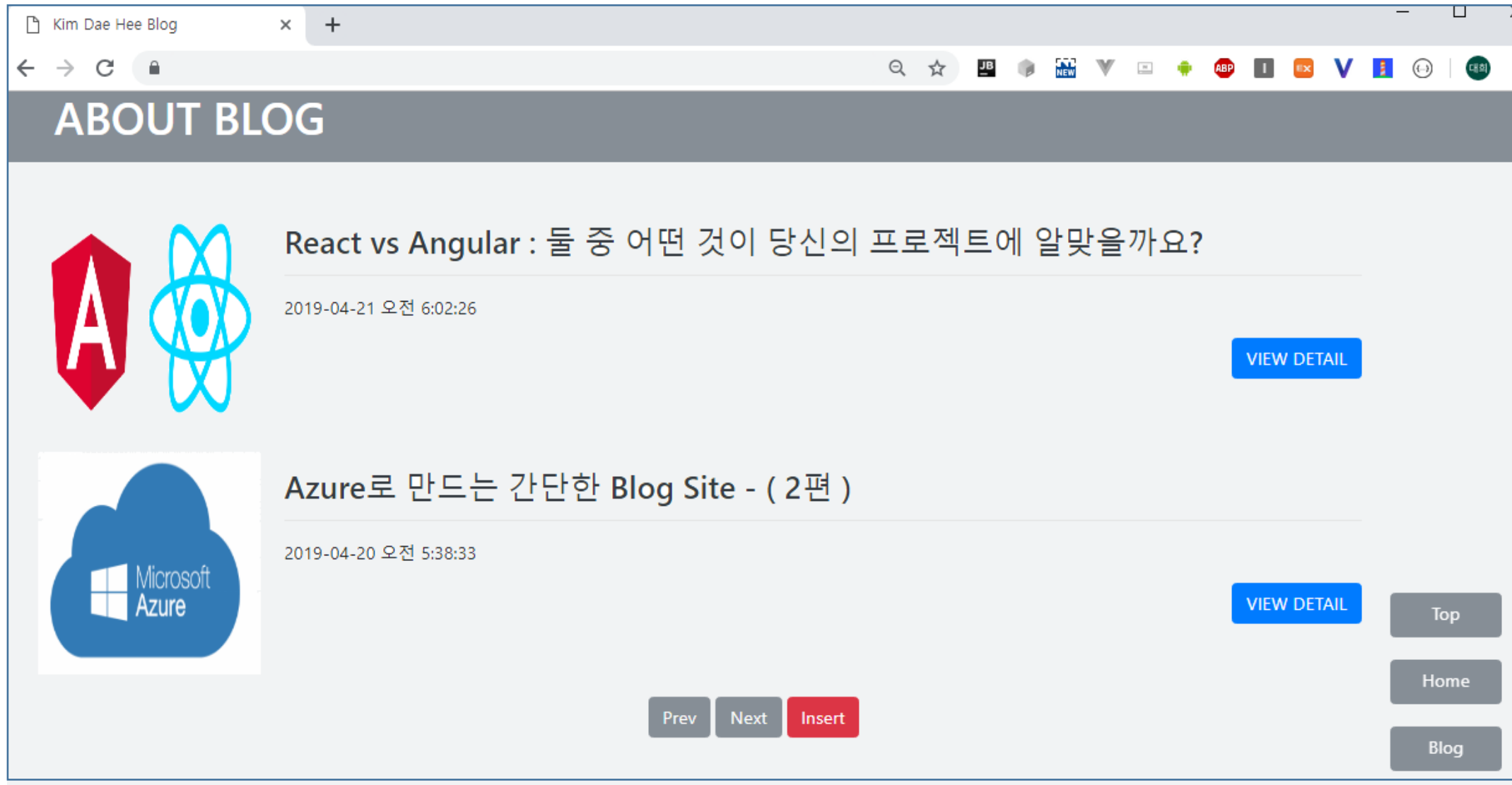
REACTJS vs. ANGULAR

```
// 블로그 상세 페이지
// URL : /Blog/Detail
public IActionResult Detail(int BlogId)
{
    // 해당 Id에 해당하는 Blog글이 미존재 하는 경우
    if (!ModelState.IsValid)
    {
        return View();
    }

    Blog DetailBlog = _BlogContext.Blog.Find(BlogId);
    return View(DetailBlog);
}
```



# DATABASE 연동



# DATABASE 연동

## ABOUT DETAIL BLOG

### [번역글] React vs Angular : 둘 중 어떤 것이 당신의 프로젝트에 알맞을까요?

#### 소개

안녕하세요. Popit에서 글을 쓰고 있는 김대희라고 합니다. 백엔드 개발자로서, 프론트 개발자 분들이 쓰는 프레임워크 비교 기사를 보게되었는데 이 글이 흥미롭고 재미있게 읽을 수 있어서 해당 글을 번역하게 되었습니다. 기사와 개발자분들이 접하는 부분이 많이 다를 수 있고, 다소간 매끄럽지 않더라도 많은 좋은 피드백 부탁드립니다. 원본글은 다음글에 참고하세요.

- [React vs. Angular Compared: Which One Suits Your Project Better?](#)

### React vs Angular : 둘 중 어떤 것이 당신의 프로젝트에 알맞을까요?

세계에서 많이 쓰이는 두 종류의 Front-End 개발 도구중 어느것을 결정하려고 시도하고 있습니까? 훌륭한 비교 분석이 될 수 있는 이 글을 읽어보시면 됩니다. 프로그래밍 세계에서 Angular와 React는 Front-End 개발자에게 가장 널리 사용되는 JavaScript 프레임 워크 하나입니다. 더욱이, Angular와 React는 [Stack Overflow Developer Survey 2018](#)에서 조사한 모든 프로그래밍 언어에서 [Node.js](#)와 함께 소프트웨어 엔지니어가 가장 많이 사용하는 프레임워크입니다. 이러한 Front-End 프레임 워크는 모두 인기가 비슷하고 유사한 아키텍처를 사용하며 JavaScript를 기반으로 합니다. 그러면 이 두개의 프레임워크는 무엇이 다를까요? 지금부터 React와 Angular를 비교해 봅시다. 다음 단락에서 프레임 워크의 일반적인 특성을 살펴 보도록 하겠습니다. 추가로 React와 Angular의 비교가 더 필요하다면 [크로스 플랫폼 모바일 프레임워크](#) (React Native 포함) 기사나 Angular와 [다른 Front-End 프레임워크를 비교한 기사](#)를 읽어보세요.

#### Angular와 React.JS에 대한 간단한 설명

Angular는 Google에서 만들고 관리하는 Front-End 프레임 워크이며 대부분의 코드 편집기에서 작성할 수 있습니다. Angular는 MEAN 스택의 구성원이며(A), MEAN 스택은 동적인 웹 사이트와 웹 애플리케이션을 구축할 때 사용되는 Javascript를 중심으로 작동하는 무료 오픈 소스 Tool Set입니다. MEAN 스택은 MongoDB (NoSQL 데이터베이스), Express.js (Back단 프레임워크), Angular 또는 AngularJS (Front-End 프레임 워크) 및 Node.js (서버 플랫폼)로 구성됩니다. Angular 프레임워크는 개발자들이 동적 단일 페이지 웹 애플리케이션(SPA)을 만들 수 있게 해줍니다. Angular가 처음 출시되었을 때 가장 주요한 장점은 HTML 기반 문서를 동적 인 콘텐츠로 변환 할 수 있다는 점이었습니다. 이 기사에서는 AngularJS(Angular 1)과 Angular(2버전 이후)의 차이점을 해결하기 위해 일반적으로 Angular 7이라고하는 최신 버전의 Angular에 중점을 둘 것입니다. Angular는 Forbes, WhatsApp, Instagram, healthcare.gov, HBO, Nike 등에서 사용을 하고 있습니다. [React.js](#)는 동적 사용자 인터페이스를 만들기 위해 Facebook에서 2011 년에 만든 오픈 소스 JavaScript 라이브러리입니다. React는 Front-End 개발을 위한 HTML 요소의 생성을 허용하는 JavaScript 및 [JSX](#)를 기반으로 합니다. React에는 모바일 앱 개발을 위한 별도의 Cross-Platform 프레임 워크 인 React Native가 있습니다. 링크된 글에서 React.js와 React Native 둘 다에 대한 심도 있는 정보를 제공하고 있습니다. React는 Netflix, PayPal, Uber, Twitter, Udemy, Reddit, Airbnb, Walmart 등에서 사용됩니다.

#### Toolset : Framework vs. Library

프레임워크 생태계는 실제 엔지니어링을 진행할 때(개발할 때) 밀접하게 영향을 끼칩니다. 여기에서는 Angular 및 React와 함께 일반적으로 사용되는 주요 도구들을 살펴 보겠습니다. 우선, React는 실제로 프레임워크가 아니라 라이브러리입니다. 이는 추가 개발도구 및 라이브러리를 여러 번 통합해야 한다는 것을 의미합니다. Angular는 프레임워크이기 때문에 이미 앱을 만들기 시작할 모든 개발 도구들을 가지고 있습니다.

## REACTJS vs. ANGULAR

# DATABASE 연동

---

## 기존의 블로그 vs 리뉴얼 된 블로그

	기존의 블로그 (ASP.NET MVC 5)	리뉴얼 블로그 (.NET Core 2.X )
DB연동	X	(Pomelo) Mysql

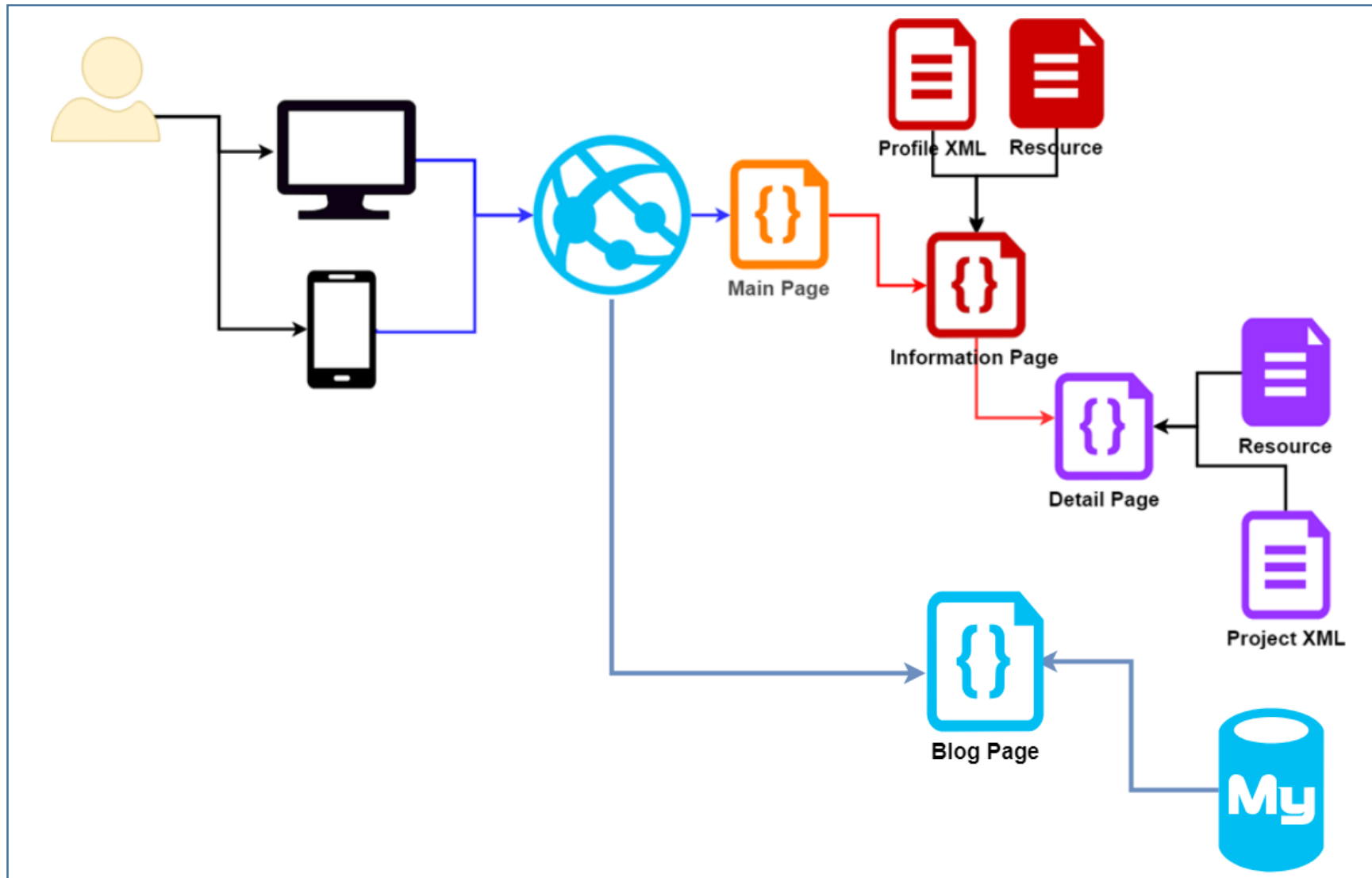
# 블로그 리뉴얼 도전

---

## 드디어

# 블로그 리뉴얼이 끝났다

# 블로그 리뉴얼 도전



# 결론

# 결론

## 기존의 블로그 vs 리뉴얼 된 블로그

	기존의 블로그 (ASP.NET MVC 5)	리뉴얼 블로그 (.NET Core 2.X )
URL 호출 Controller	ActionResult	ActionResult
작업환경	Windows	Windows, Ubuntu
공통 Layout	X	_ViewStart + _Layout (.cshtml)
DB연동	X	(Pomelo) Mysql
리소스관리	Resource/App_DATA	Wwwroot
공통 모델	X	_ViewImports (.cshtml)
사용 Class	Profile	Profile, Github, Linkdin
이미지 로드	Base64 Data	Directory



## 전체 과정으로

무엇을 얻었는가?

## 만드는 과정의 팀워크 만들어진 결과물의 보람

**기존의 만들어진 기능의 자유로운 수정  
새로운 기술에 대한 두려움의 사라짐**

## 개발하는 재미

# 참고자료

---

## 1. ASP.NET Core MSDN

-> <https://docs.microsoft.com/ko-kr/aspnet/core/?view=aspnetcore-2.2>

## 2. Korea Azure User Group

-> <https://www.facebook.com/krazure/>

## 3. Github Repo

-> <https://github.com/DaeHeeKim93>

## 4. Popit Azure ( Web Service 부분까지만 )

-> <https://www.popit.kr/azure%EB%A1%9C-%EB%A7%8C%EB%93%9C%EB%8A%94-%EA%B0%84%EB%8B%A8%ED%95%9C-blog-site-2%ED%8E%B8/>

**감사합니다.**