

# 프로젝트 명세서

웹 IoT 프로젝트  
“ Wedding101, IoT 플랫폼 ”

A101  
팀장 김성환  
팀원 권영진  
팀원 김지현  
팀원 류제엽  
팀원 이동형  
팀원 이진욱

## 목차

1. 프로젝트 개요 .....	3
1) 소개 .....	3
2) Sub PJT 1. Back-End .....	3
3) Sub PJT 2. Front-End .....	4
4) Sub PJT 3. HardWare.....	4
2. 프로젝트 목표 .....	5
3. 필수 지식 습득 .....	5
4. 과제 명세 .....	7
1) Back-End .....	7
Req. 1. 회원관리	
Req. 2. 결혼정보	
Req. 3. 청첩장	
Req. 4. 앨범	
Req. 5. 미디어	
Req. 6. 리뷰 및 문의게시판	
2) Front-End .....	10
Req. 1. 레이아웃	
Req. 2. 메인 화면	
Req. 3. 서비스 신청	
Req. 4. 모바일 청첩장	
Req. 5. 앨범	
Req. 6. 리뷰 및 문의	
Req. 7. 회원	
Req. 8. 관리자	
3) HardWare .....	15
Req. 1. Raspberry Pi OS 설치	
Req. 2. 환경 세팅	
Req. 3. 회로도 작성	
Req. 4. 서버 연결	
Req. 5. GUI 구성	
Req. 6. 자동 실행	
5. 프로젝트 실행 및 배포 .....	18
6. 기술 스택 .....	21
7. 별첨 .....	22

# 1. 프로젝트 개요

---

## 소개

Wedding101이란 결혼(Wedding)과 101(입문)의 합성어입니다. 101은 학문에서 처음 배우는 입문수업을 의미하는데, 결혼이라는 일생일대의 경험이 좋은 추억으로 남을 수 있도록 본 서비스가 좋은 길잡이가 될 수 있음을 의미하고 있습니다. Wedding101은 예비 부부가 결혼 과정에서 지나치기 쉬운 순간들을 사진과 영상으로 담아 디지털 앨범을 제공하는 서비스입니다. 본 프로젝트는 IoT 플랫폼을 이용한 미디어와 디지털 앨범이라는 웹 서비스를 통해 아날로그와 디지털이 통합된 사용자 경험을 제공하고자 합니다.

독특하지만 익숙한 경험을 제공하는 IoT 부스를 통해 영상과 사진을 남기는 데 거부감이 없도록 하였으며, 서비스 신청자는 특별한 날의 기록을 간직할 수 있습니다. 웹 측면에서는 Wedding101이 전달하고자 하는 바를 한 눈에 알아볼 수 있도록 하면서 모바일 환경에 익숙한 대중에게 모바일 청첩장, 앨범관리 서비스를 제공하면서 익숙함을 잃지 않고자 하였습니다.

프로젝트의 진행은 3개의 Sub PJT로 진행되며, 각 Sub PJT의 진행은 아래 표와 다음 각 단계를 설명한 내용을 참고하시기 바랍니다.

Sub PJT 1. Back-End	Sub PJT 2. Front-End	Sub PJT 3. HardWare
<ul style="list-style-type: none"><li>- Spring으로 백엔드 구성</li><li>- MySQL로 DB 구성</li><li>- 회원, 앨범 관리</li></ul>	<ul style="list-style-type: none"><li>- Window에 기반한 환경 구성</li><li>- React로 온라인 웹 플랫폼 제작</li><li>- PC웹 기반의 반응형 웹</li></ul>	<ul style="list-style-type: none"><li>- Raspberry Pi로 하드웨어 구성</li><li>- API 제작</li><li>- BE 서버 연결</li></ul>

## Sub PJT 1. Back-End

Sub PJT 1에서는 ERD를 그리며 데이터베이스 구조를 설계하고 Spring을 사용하여 서비스의 백엔드 영역을 구축합니다. AWS의 S3를 이용해 이미지 및 동영상 업로드 기능을 구현합니다. 마찬가지로 AWS EC2를 통해 서버를 관리하게 됩니다. 주 기능으로는 회원관리, 앨범관리, 결혼정보 및 청첩장 관리, 미디어 관리, 리뷰 및 문의 페이지 등이 있습니다. 또한 IoT 기기에서 받아오는 미디어 정보를 DB에 저장하는 역할을 하기도 합니다.

## Sub PJT 2. Front-End

Sub PJT 2에서는 React를 사용하여 서비스의 프론트 영역을 구현합니다. 화면은 메인페이지, 서비스 신청페이지, 모바일 청첩장 페이지, 앨범 페이지, 리뷰 및 문의 페이지, 회원 페이지로 나뉘며 각 화면은 반응형으로 웹과 모바일에 차이를 둡니다. 구현이 완료되었다면 회원 및 게시판 관리를 위한 관리자용 페이지를 구현합니다.

## Sub PJT 3. HardWare

Sub PJT 3에서는 라즈베리파이와 터치스크린, Pi Cam, 마이크를 사용하여 서비스의 하드웨어를 구현합니다. 라즈베리파이가 부팅되자마자 제작한 프로그램이 전체화면으로 실행되며, 터치를 이용해 간단히 조작할 수 있습니다. 이용자는 이름, 관계, 수신자에 대한 정보를 설정한 이후 사진이나 영상 편지를 촬영할 수 있습니다. 사진과 영상이 문제 없이 기록되었다면 AWS 서버로 전송합니다.

## 2. 프로젝트 목표

---

- 1) Spring와 React를 활용한 IoT 플랫폼 구축
- 2) 임베디드 KIT를 이용한 IoT 서비스 구현
- 3) AWS 서버를 통해 배포

## 3. 필수 지식 학습

---

프로젝트 진행을 위한 자기주도 학습에 실마리가 될 수 있는 필수 학습 대상 키워드와 유용한 참고 자료 링크 목록이 아래 표에 제시되어 있습니다. 아래 목록에 제시된 내용은 기초적인 내용에 해당하는 예이며, 이외 추가로 필요한 항목들을 직접 찾아 학습해 나가시기 바랍니다.

주제		참고 링크
반응형 SPA	ES6	JavaScript의 Array 객체에 대한 소개 <a href="https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array">https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array</a>
	React	React 공식 문서(React 기본 예시, CRA, JSX, Hook, Context 등) <a href="https://ko.reactjs.org/docs/hello-world.html">https://ko.reactjs.org/docs/hello-world.html</a>
	SPA	SPA 개념 설명 <a href="https://linked2ev.github.io/devlog/2018/08/01/WEB-What-is-SPA/">https://linked2ev.github.io/devlog/2018/08/01/WEB-What-is-SPA/</a>
	반응형 웹	반응형 웹 디자인 패턴 <a href="https://developers.google.com/web/fundamentals/design-and-ux/responsive/patterns?hl=ko">https://developers.google.com/web/fundamentals/design-and-ux/responsive/patterns?hl=ko</a>
	MySQL	MySQL 공식 문서 <a href="https://dev.mysql.com/doc/">https://dev.mysql.com/doc/</a>
임베디드	Raspbian	Raspbian OS 설치 가이드


키오스크 제작		<a href="https://geeksvoyage.com/raspberry%20pi4/installing-os-for-pi4/">https://geeksvoyage.com/raspberry%20pi4/installing-os-for-pi4/</a>
	Arduino	아두이노 기초 <a href="http://blog.naver.com/PostView.nhn?blogId=jamduino&amp;logNo=220812436210">http://blog.naver.com/PostView.nhn?blogId=jamduino&amp;logNo=220812436210</a>
	Arduino 연결	Node.js에서 시리얼 포트를 활용한 아두이노 연결 <a href="https://linked2ev.github.io/devlog/2018/08/01/WEB-What-is-SPA/">https://linked2ev.github.io/devlog/2018/08/01/WEB-What-is-SPA/</a>
	Express.js	Express.js의 이해 및 설치 <a href="http://webframeworks.kr/getstarted/express.js">http://webframeworks.kr/getstarted/express.js</a>
	Material-UI	Material UI Theme 사용법 <a href="https://bitly.kr/1LmbXZH">https://bitly.kr/1LmbXZH</a>
	pm2	프로세스 관리툴 중 하나인 pm2의 사용법 소개 <a href="https://blog.outsider.ne.kr/1197">https://blog.outsider.ne.kr/1197</a>
	crypto	암호화 모듈 crypto 사용 <a href="https://nodejs.org/api/crypto.html">https://nodejs.org/api/crypto.html</a>
임베디드 키오스크 완성	인증	인증과 권한의 개념 <a href="https://hanee24.github.io/2018/04/21/authentication-authorization/">https://hanee24.github.io/2018/04/21/authentication-authorization/</a>
	JWT	JWT 장단점 및 토큰 구성 개념 <a href="http://www.opennaru.com/opennaru-blog/jwt-ison-web-token/">http://www.opennaru.com/opennaru-blog/jwt-ison-web-token/</a>
	Socket	Socket의 정의 및 사용법 이해 <a href="https://en.wikipedia.org/wiki/Network_socket">https://en.wikipedia.org/wiki/Network_socket</a>
	SSR	SPA에서의 SSR과 CSR <a href="https://bit.ly/2NK7bfp">https://bit.ly/2NK7bfp</a>
	Next.js	Next.js 소개 <a href="https://nextjs.org/docs/getting-started">https://nextjs.org/docs/getting-started</a>
	CSS Animation	CSS Animation 소개 <a href="https://developer.mozilla.org/ko/docs/Web/CSS/CSS_Animations">https://developer.mozilla.org/ko/docs/Web/CSS/CSS_Animations</a>
	Infinite Scroll	React에서의 Infinite Scroll 구현 <a href="https://reactgo.com/react-infinite-scroll/">https://reactgo.com/react-infinite-scroll/</a>

## 4. 과제 명세

### Sub PJT 1. Back-End

#### Req. 1. 회원관리

#### 회원가입 및 로그인 페이지 관리

제목	웹 - 로그인 페이지
구성도	 <p>The image shows a mockup of the Naver login page. It features the Naver logo at the top, followed by a navigation bar with links for ID login, mobile number login, and QR code login. Below this is a form with fields for ID and password, a checkbox for 'Remember me', and a 'Login' button. There are also links for 'Forgot password', 'Forgot ID', and 'Sign up' at the bottom.</p>
설명	<ul style="list-style-type: none"> <li>로그인</li> <li>회원가입</li> <li>아이디/패스워드 찾기</li> <li>회원 정보 수정</li> <li>회원 탈퇴</li> </ul>

#### Req. 1-1. 회원 가입 기능 구현

기능 상세	<ul style="list-style-type: none"> <li>회원가입 API 구현</li> <li>ID, e-mail, PW 필수 입력</li> <li>정규표현식을 이용하여 FE에서 1차로 통과한 ID, e-mail, PW등의 유효성 검사</li> <li>PW는 암호화하여 통신 및 DB에 저장</li> </ul>
-------	--

Req. 1-2. 회원 로그인, 로그아웃 기능 구현

기능 상세	<ul style="list-style-type: none"> <li>• 회원 로그인 API 구현</li> <li>• ID, e-mail, 패스워드를 필수 입력 항목으로 설정</li> <li>• 패스워드는 암호화하여 통신</li> </ul>
----------	--

Req. 1-3. 회원 아이디 찾기 기능 구현

기능 상세	<ul style="list-style-type: none"> <li>• 아이디 분실 시 회원 이름과 이메일을 통해 찾기 가능</li> </ul>
----------	---

Req. 1-4. 회원 비밀번호 찾기 기능 구현

기능 상세	<ul style="list-style-type: none"> <li>• 비밀번호 분실 시 아이디와 이메일을 통해 찾기 가능</li> <li>• </li> <li>• 임시 비밀번호 발급하여 이메일을 통해 전송 (SMTP)</li> </ul>
----------	--

Req. 2. 결혼정보

기능 상세	<ul style="list-style-type: none"> <li>• 결혼 정보 등록하기</li> <li>• 결혼 정보 시퀀스와 회원 시퀀스를 통해 관리</li> <li>• 결혼일, 예식장 정보는 필수, 예비 부부 이름과 번호도 필수</li> <li>• 계좌, 혼주 관계, 부모 이름 및 생사 여부 등은 선택으로 관리</li> <li>• 청첩장 정보와 연계</li> </ul>
----------	--

Req. 2-1. 서비스 신청 시 받은 정보 관리

기능 상세	<ul style="list-style-type: none"> <li>• 필수, 선택 필드 분리하여 관리</li> <li>• FE 지도 api 정보를 받아와 관리</li> <li>• FE 캘린더 api 정보를 받아와 관리</li> </ul>
----------	--



### Req. 3. 청첩장

기능 상세	<ul style="list-style-type: none"> <li>예시 템플릿 n개 관리</li> <li>선택한 템플릿 정보와 결혼 정보 조합</li> <li>청첩장 수정 정보 관리</li> <li>청첩장 사진 관리(파일관리)</li> </ul>
----------	---

#### Req. 3-1. 청첩장 생성

기능 상세	<ul style="list-style-type: none"> <li>결혼 정보를 불러와 선택한 템플릿에 삽입</li> <li>템플릿에 예시로 작성된 문구는 수정 가능</li> <li>첨부 파일이 삽입된 템플릿은 사진 수정 가능</li> </ul>
----------	--

#### Req. 3-2. 청첩장 수정 기능 구현

기능 상세	<ul style="list-style-type: none"> <li>결혼 정보 제외, 청첩장 정보(템플릿의 문구와 수정)만 수정 가능</li> </ul>
----------	--

### Req. 4. 앨범

기능 상세	<ul style="list-style-type: none"> <li>유저가 선택 or 입력한 앨범명, 앨범색, 커버 사진 등 불러오기 및 수정</li> <li>미디어 신규 등록에 따른 알람 표시(추가기능)</li> <li>앨범 서비스 이용 건당 리뷰 작성기회 1번 부여</li> <li>앨범 삭제기능</li> </ul>
----------	---

#### Req. 4-1. 앨범 리뷰 작성 기능 구현

기능 상세	<ul style="list-style-type: none"> <li>해당 앨범에 대한 리뷰 작성 여부 확인 후 리뷰 작성 가능</li> <li>리뷰 작성은 서비스를 사용 중인 회원만 가능</li> </ul>
----------	--

### Req. 5. 미디어

기능 상세	<ul style="list-style-type: none"> <li>미디어 CRUD</li> <li>시간순, 이름순 정렬</li> <li>QR에서도 api 호출 가능</li> <li>미디어 북마크 기능</li> </ul>
----------	--

Req. 6. 리뷰 및 문의게시판

기능 상세	<ul style="list-style-type: none"><li>• 게시글 작성 및 조회</li><li>• 자신의 글은 수정가능</li><li>• 게시글 최신 순 정렬</li><li>• 리뷰 별점관리</li><li>• 리뷰 내 '도움이되는리뷰' 기능</li></ul>
----------	---

Req. 6-1. 관리자 기능

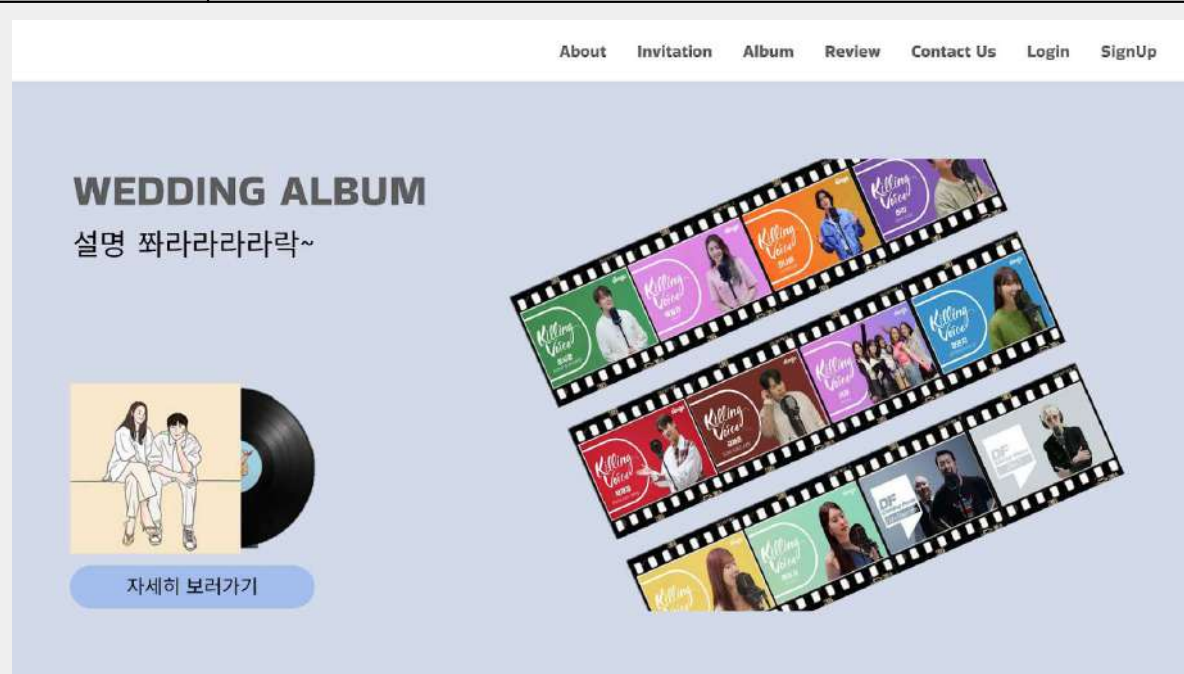
기능 상세	<ul style="list-style-type: none"><li>• 관리자는 모든 게시글 삭제, 수정 권한 보유</li><li>• 관리자는 댓글을 통해 답변가능</li></ul>
----------	---

## Sub PJT 2. Front-End

### Req. 1. 레이아웃

Req. 1-1	레이아웃 내용 구성
기능 명세	<ul style="list-style-type: none"> <li>Header에 로고, 메뉴, 회원관리 버튼 배치</li> <li>Footer에 팀 정보 추가</li> <li>메인화면을 제외한 모든 페이지에서 왼쪽 Navbar 구현</li> <li>메인화면은 스크롤핸들러로 4개 페이지 구현</li> </ul>

### Req. 2. 메인 화면 - ABOUT 페이지

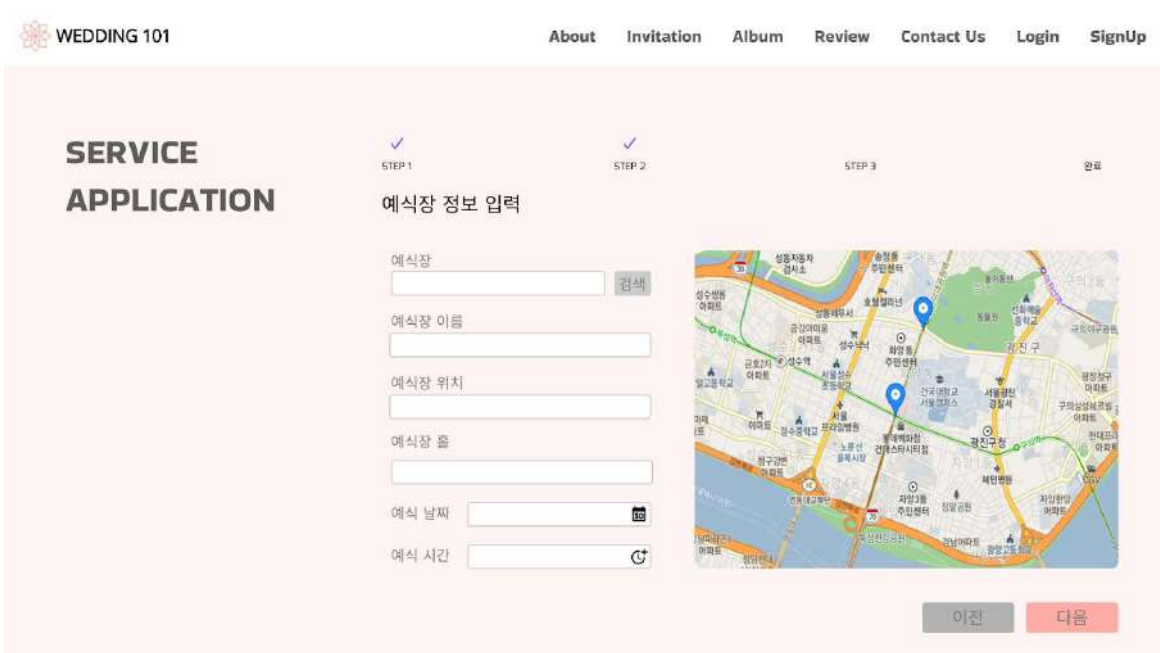
Req. 2	메인화면
	

Req. 2-1	메인화면 구체화
기능명세	<ul style="list-style-type: none"> <li>메인#1 첫화면에 웨딩사진과 글로 서비스 설명</li> <li>메인#2 모바일 청첩장 예시 삽입</li> <li>메인#3 앨범 예시 삽입</li> <li>메인#4 결혼식순에 맞는 타임 스탬프로 서비스 개요 삽입</li> </ul>

Req. 2-2	버튼 및 메뉴에 주요 이벤트 기능 구현
기능명세	<ul style="list-style-type: none"> <li>• 로고 클릭 시 홈페이지로 이동</li> <li>• login / signup 버튼 클릭 시 해당 페이지로 이동</li> <li>• Invitation 버튼으로 신청 페이지 이동</li> <li>• Review 버튼으로 리뷰 게시판 페이지 이동 (로그인 상태에서만 접근 가능)</li> <li>• 모든 메인페이지 하단에 서비스 신청 버튼 절대위치 플로팅</li> <li>• 서비스 신청 버튼 - 로그인시 서비스 신청 페이지로, 비로그인시 로그인 페이지로 이동</li> </ul>

### Req. 3. 서비스 신청

Req. 3	서비스 신청 화면
--------	-----------



Req. 3-1	신청 프로세스 UI 구현
기능 명세	<ul style="list-style-type: none"> <li>• ProgressBar를 통해 신청프로세스 표시</li> <li>• 개인정보취급동의, 결혼정보, 예식장정보, 신청완료로 구성</li> <li>• 서비스 이용 약관동의 - 개인정보, 위치서비스, 프로모션 등</li> <li>• 결혼정보 - 신랑신부 이름, 번호(필수) + 혼주, 계좌 등(선택)</li> <li>• 예식장 정보 - 검색을 통한 식당 선택, 지도에 마커 구현</li> <li>• 예식장 정보 - 캘린더 api 이용해 날짜 시간 입력받기</li> <li>• 신청 완료 - 모바일 청첩장 및 앨범 생성페이지로 이동</li> </ul>

## Req. 4. 모바일 청첩장

Req. 4-1	모바일 웹 기반의 반응형 웹 - 예비 부부
	
설명	<ul style="list-style-type: none"> <li>• PC웹에서 예시 템플릿 선택 후 청첩장 정보 기입</li> <li>• 사진 업로드 UI를 통해 템플릿 예시 사진 개인화</li> <li>• 수정 기능을 통해 청첩장 수정 가능</li> </ul>

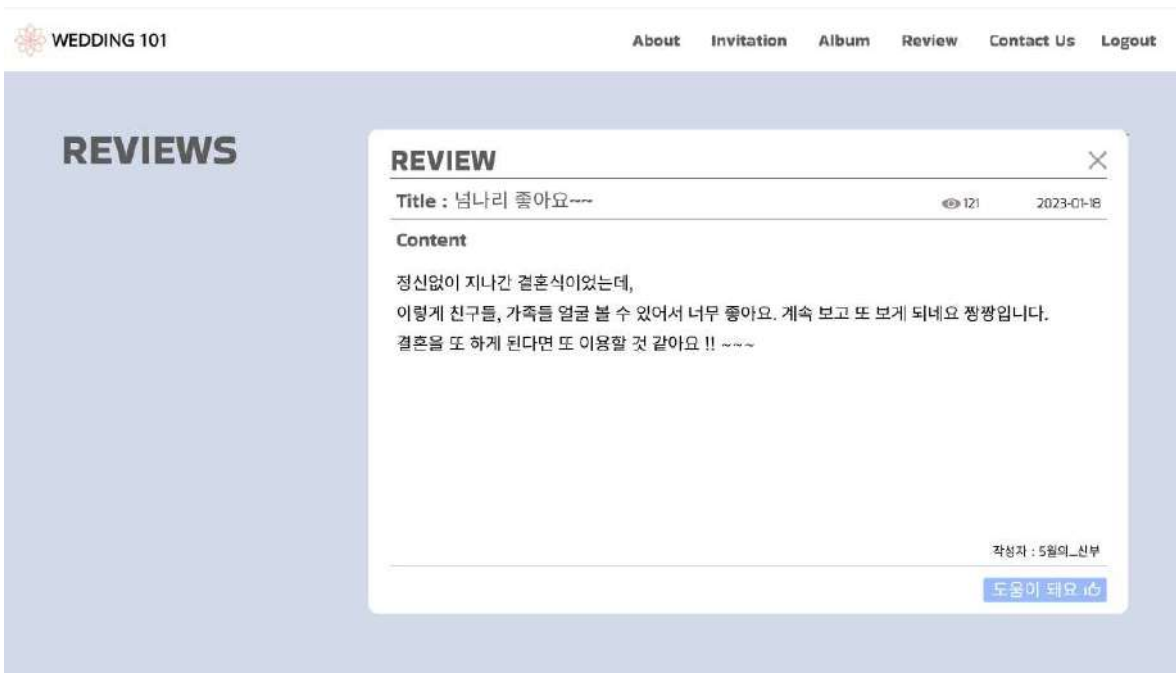
Req. 4-2	모바일 웹 기반의 반응형 웹 - 예비하객
	
설명	<ul style="list-style-type: none"> <li>● 모바일 최적화 템플릿 UI로 보여주기</li> <li>● 스크롤 시 축하영상 업로드 UI 구현</li> <li>● 현장 부스에서 QR 코드를 통해 업로드 페이지 확인 가능</li> </ul>

#### Req. 5. 앨범

Req. 5-1	앨범 화면
설명	<ul style="list-style-type: none"> <li>● 앨범정보 UI</li> <li>● 앨범 수정 삭제 UI</li> <li>● 리뷰 작성 버튼 구현 - 리뷰 게시판으로 이동</li> </ul>
Req. 5-2	미디어 목록 페이지
설명	<ul style="list-style-type: none"> <li>● 날짜, 이름 순 정렬 UI, 검색 UI 구현</li> <li>● 중앙에 메인 미디어, 우측에 서브 미디어 배치</li> <li>● 미디어 휴지통 페이지 UI</li> </ul>

	<ul style="list-style-type: none"> <li>● 통합본 신청을 위한 미디어 즐겨찾기 UI, 즐겨찾기 페이지 구현</li> </ul>
--	---

## Req. 6. 리뷰 및 문의

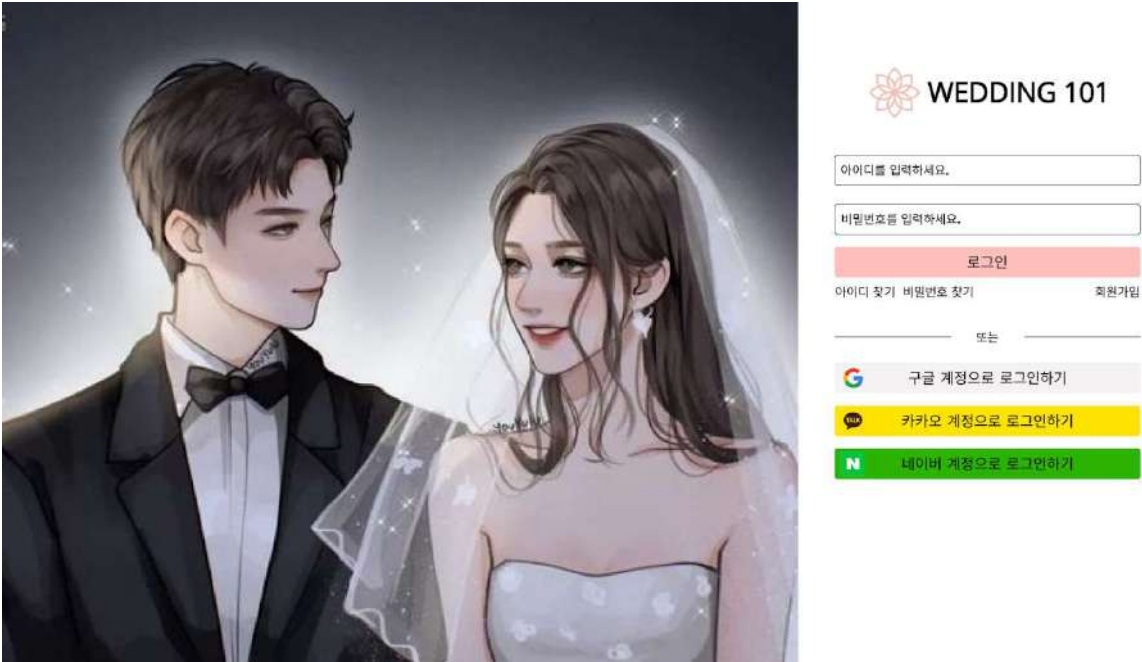
Req. 6-1	리뷰게시판 구현
	
설명	<ul style="list-style-type: none"> <li>● 서비스 이용자만 글작성 가능</li> <li>● 앨범 페이지에서 리뷰 게시판 이동 버튼 구현</li> <li>● 리뷰 목록에서 특정 리뷰 선택 시 모달창으로 리뷰 상세 UI 구현</li> <li>● 리뷰 별점과 좋아요 UI</li> <li>● 댓글 구현 (추가 기능)</li> </ul>

Req. 6-2	문의 게시판 구현
	
설명	<ul style="list-style-type: none"> <li>• 회원만 글 작성 가능</li> <li>• 문의 선택 시 모달창으로 문의 상세 UI 구현</li> <li>• 댓글 구현</li> <li>• 관리자만 댓글 작성 가능</li> </ul>

## Req. 7. 회원

Req. 7-1	회원가입 화면
설명	<ul style="list-style-type: none"> <li>• 일반 회원가입 페이지 구현</li> </ul>



Req. 7-2	회원 로그인 화면 구현
	
설명	<ul style="list-style-type: none"> <li>• 회원 로그인 페이지 구현</li> <li>• 로그인 하단에 간편가입버튼 구현(oauth)</li> </ul>

## Req. 8. 관리자

Req. 8-1	회원 및 게시판 관리를 위한 관리자 기능 구현
설명	<ul style="list-style-type: none"> <li>• 회원 관리 화면 - 회원 조회, 계정 잠금 기능</li> <li>• 통계 관리 화면 - 월별 서비스 신청 수, 유입경로, 서비스 이용고객 관련 정보 등</li> <li>• 영상통합본 승인 관리 화면 - 영상통합본 신청 로그 확인, 승인 기능</li> <li>• 게시판 관리 화면 - 광고 및 도배관리, 답글 작성 기능</li> </ul>

## Sub PJT 3. HardWare

### Req. 1. 라즈베리파이 OS 설치 및 초기 환경설정

Req. 1-1	sd카드에 라즈베리파이 os설치
설명	<ul style="list-style-type: none"> <li>● raspberry pi imager 설치</li> <li>● sd 카드 포맷 후 os설치</li> </ul>

Req. 1-2	라즈베리파이 초기 설정
설명	<ul style="list-style-type: none"> <li>● 시간 설정</li> <li>● 해상도 설정</li> <li>● WIFI 연결 및 IP 고정 설정</li> <li>● SSH, VNC 등 원격 연결 설정</li> </ul>

Req. 1-3	윈도우 상의 원격제어 연결
설명	<ul style="list-style-type: none"> <li>● VNC Viewer 설치</li> <li>● IP확인 후 연결</li> <li>● SSHkey 발급 후 자동 연결</li> </ul>

### Req. 2. 환경 세팅(라즈베리파이 쉘)

Req. 2-1	라이브러리 목록
설명	<ul style="list-style-type: none"> <li>● sudo apt-get update &amp;&amp; sudo apt-get upgrade</li> <li>● tmux (협업)</li> <li>● build-essential cmake pkg-config (개발자 도구)</li> <li>● libjpeg-dev libtiff5-dev libjasper-dev libpng-dev (이미지 파일 형식 Load)</li> <li>● libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev (비디오 코덱)</li> <li>● libfontconfig1-dev libcairo2-dev libgdk-pixbuf2.0-dev libpango1.0-dev libgtk2.0-dev libgtk-3-dev (GUI 모듈)</li> <li>● libatlas-base-dev gfortran (OpenCV 종속성 모듈)</li> <li>● libhdf5-dev libhdf5-serial-dev libhdf5-103 (HDF5)</li> <li>● python3-pyqt5 (Qt GUI)</li> <li>● pip3 install imutils</li> <li>● pip3 install opencv-contrib-python</li> <li>● pip3 install opencv-python</li> </ul>

### Req. 3. 추가 HW 구성

Req. 3-1	필요 HW 목록
설명	<ul style="list-style-type: none"> <li>● raspberry pi 4 model B</li> <li>● usb 마이크</li> <li>● Pi Camera V2.1</li> <li>● 블루투스 스피커</li> <li>● 터치스크린 - 아이패드</li> </ul>

Req. 3-2	HW 구성 및 설정
설명	<ul style="list-style-type: none"> <li>● Pi Camera 사용을 위해 raspi-config를 통해 설정</li> <li>● \$ alsamixer 명령어로 마이크 녹음 감도 설정</li> <li>● 블루투스 스피커와 원격 연결</li> </ul>

### Req. 4. 서버 연결

Req. 4-1	DB 연결
설명	<ul style="list-style-type: none"> <li>● mysql-server &amp; mysql -client 설치</li> <li>● mysql connector 라이브러리를 이용해서 DB 연결 예정</li> <li>● db, cur 변수 활용해 db접근 및 쿼리문 전송</li> </ul>

Req. 4-2	데이터 송수신
설명	<ul style="list-style-type: none"> <li>● AWS ubuntu server를 기반하여 MySQL Server 와 통신</li> <li>● 결혼 정보를 받아와 하드웨어에 등록</li> <li>● 미디어 데이터와 입력 데이터를 하드웨어에서 server로 송신</li> </ul>

### Req. 5. GUI 구성

Req. 5-1	터치 스크린 GUI 연결
설명	<ul style="list-style-type: none"> <li>● raspberry pi에 Qtside2 설치</li> <li>● windows 에서 작성한 python코드 옮겨오기</li> </ul>

Req. 6. 자동 실행

Req. 7	전원 공급 시 프로그램 자동 실행
설명	<ul style="list-style-type: none"><li>• crontab, rc.local 등 맞는 방법 선택</li></ul>

## 5. 프로젝트 실행 및 배포

---

### EC2 인스턴스

Amazon Elastic Compute Cloud(Amazon EC2)는 500개가 넘는 인스턴스, 그리고 최신 프로세서, 스토리지, 네트워킹, 운영 체제 및 구매 모델의 옵션과 함께 워크로드의 요구 사항에 가장 잘 부합할 수 있도록 가장 포괄적이고 심층적인 컴퓨팅 플랫폼을 제공합니다.<sup>1</sup> AWS EC2 인스턴스를 통해 리눅스 기반 Ubuntu20.04LTS 서버를 구축하고 MobaXterm의 SSH 통신을 통해 인스턴스에 접속합니다.

### Nginx&uWSGI를 이용한 프로세스 관리

Nginx는 동시접속 처리에 특화된 웹서버 프로그램으로 다수의 사용자가 웹에 접속할 경우에 권장되는 사양입니다. 비동기 처리 방식을 채택하여 사용자들의 요청 간의 버퍼링과 대기 시간을 줄일 수 있다는 장점이 존재합니다. AWS 인스턴스 상에 Nginx를 설치하여 다수의 프로그램을 중계합니다. \$ sudo apt-get install nginx 명령어를 통해 Nginx 설치 후 환경설정을 세팅할 수 있습니다.

Nginx는 Node.js 환경에서 Proxy 서버로 활용될 수 있으나 Python을 사용하는 Django는 Nginx를 proxy 서버로 이용할 수 없습니다. 따라서 본 프로젝트에서는 Nginx와 Django를 이어주는 중간 다리 역할로서 uWSGI를 채용합니다.

uWSGI는 python 패키지를 사용할 수 있는 애플리케이션 컨테이너 서버로 애플리케이션 호출을 통해 애플리케이션 서버가 코드와 통신할 수 있도록 합니다. 일반적으로 서버가 접근할 수 있는 python 모듈에서 ‘애플리케이션’이라는 이름의 객체로 제공되곤 합니다. 본 프로젝트의 uWSGI는 실제 물리 서버가 아닌 가상 인스턴스(venv)에서 작동합니다.

```
$ cd /etc/nginx/sites-available
$ vi default
#default
...

server {
    listen 80 default_server;
}
```

위와 같이 Nginx의 default 파일을 수정하면 localhost의 8000번 포트 대신 Nginx에서 입력한 80번의 포트 번호로도 페이지에 접속할 수 있습니다.

---

<sup>1</sup> AWS EC2, [https://aws.amazon.com/ko/ec2/features/?trk=ec2\\_landing](https://aws.amazon.com/ko/ec2/features/?trk=ec2_landing), 2022.08.15

## S3 이미지 등록

Amazon Simple Storage Service(Amazon S3)는 AWS에서 제공하는 객체 스토리지 서비스입니다. 스토리지 내에서 이미지를 포함한 데이터를 저장하고 보호할 수 있습니다. 본 프로젝트에서는 웹서비스와 커뮤니티 내에서 사용되는 다양한 이미지 파일들을 S3에 올림으로써 DB가 무거워지는 것을 방지하여 서비스 성능을 향상시키고자 합니다. 대표적으로 본 프로젝트의 S3 스토리지 내에는 wedding101 웹 서비스 과정에서 생성되는 미디어 파일이 포함되어 있습니다.

## 프로그램 배포

예전으로 거슬러 올라가면 서버를 점검하고 수정할 때마다 서버를 정지시키고 재실행해서 변경사항을 적용하곤 했습니다. 이건 당연히 거대한 서비스상의 결함입니다. 서버가 재실행되는 동안은 사용자들이 접속을 할 수 없고, 처리 중에 있던 서비스가 중간에 중단될 수 있기 때문입니다.

따라서 웹의 배포는 백그라운드 작업으로 진행되며, 지속적인 서비스를 고려한 ‘무중단 배포’를 목표로 합니다. 이를 위해서 우선 SSH 터미널 상에서 소스를 수정해서 저장하고, runserver를 통해 서버를 실행합니다. Ctrl+Z를 눌러 실행되던 프로그램을 일시 정지 시킨 후 shell로 돌아와 \$ bg명령어를 작성하면 해당 프로그램이 백그라운드에서 작동하게 됩니다. 추가적으로 \$ disown -h 명령어를 작성하면 ssh 연결이 끊어져도 프로세스가 종료되지 않습니다. restart는 프로세스를 무조건 kill한 후에 시작하기 때문에 사용해서는 안 됩니다.

단순한 상황에서는 이것만으로 중단이 사라지지만 이때 몇 가지 구멍에서 뚫김 현상이 발생할 수 있습니다.

1. new\_app이 완전히 동작하기 전에 요청을 받을 경우
2. old\_app이 요청 처리 도중에 SIGKILL로 죽을 경우
3. HTTP Keep-Alive 가 사용될 경우

## 6. 기술 스택

PART	기술 스택 / HW	상세 내용
형상관리	GitLAB	<ul style="list-style-type: none"> <li>https://lab.ssafy.com/s08-webmobile3-sub2/S08P12A101.git</li> </ul>
이슈관리	JIRA	<ul style="list-style-type: none"> <li>S08P12A101</li> </ul>
커뮤니케이션	Mattermost	<ul style="list-style-type: none"> <li></li> </ul>
	Notion	
디자인	Figma	<ul style="list-style-type: none"> <li>Ver. 2023.1.11</li> </ul>
OS	Window	<ul style="list-style-type: none"> <li>Windows 10 Enterprise</li> </ul>
	Raspbian	<ul style="list-style-type: none"> <li></li> </ul>
DB	MySQL	<ul style="list-style-type: none"> <li>MySQL 8.0.29 (64bits)</li> </ul>
	S3	<ul style="list-style-type: none"> <li></li> </ul>
Server	AWS EC2 ubuntu	<ul style="list-style-type: none"> <li></li> </ul>
	Ngnix	<ul style="list-style-type: none"> <li></li> </ul>
FE	React	<ul style="list-style-type: none"> <li>18.0 / npm 9.3.0</li> </ul>
	node.js	<ul style="list-style-type: none"> <li>18.13.0</li> </ul>
BE	SpringBoot	<ul style="list-style-type: none"> <li></li> </ul>
HW	Raspberry Pi	<ul style="list-style-type: none"> <li>Raspberry Pi 4 model B</li> </ul>
	PySide6 + 2	<ul style="list-style-type: none"> <li>라즈베리파이의 경우 ver.2</li> </ul>
	OpenCV	<ul style="list-style-type: none"> <li></li> </ul>
IDE	Visual Studio Code	<ul style="list-style-type: none"> <li>Visual Studio Code 1.74.3</li> </ul>
	IntelliJ IDEA	<ul style="list-style-type: none"> <li>Ver. 2022.3.1</li> </ul>
	PyCharm	<ul style="list-style-type: none"> <li>Ver. 2022.3.1</li> </ul>

## \* 별첨

각 명세서에 첨부된 별첨 목록을 첨부합니다.

### 웹 IoT 명세서 1. 반응형 SPA 제작

- Chocolatey 설치
- Node.js 설치
- VS Code 설치
- 프록시 서버 NginX 설치

### 웹 IoT 명세서 2. 웹 서버 및 임베디드 기반 키오스크 제작

- Raspbian OS 설치
- Raspbian OS 환경설정
- mySQL DB 설치
- 미들웨어 활용