캡스톤디자인 계획 요약서

제출일: 2023.12.27.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 캡스톤디자인 제목 | Express Bus | | | |
| 캡스톤디자인 내용 | 국가가 발전할수록 도로의 편의성이 중요해진다. 이런 장거리 차량은 지하철, 버스와는 다르다. 이런 것들은 한국에서 매우 잘 발달되어 있다. 그 중 하나가 고속버스다. 특히, 터미널에 가지 않고도 버스표 가격을 확인할 수 있는 기능이 있어 개인이 더욱 편리하게 이용할 수 있다. 우즈베키스탄에도 고속버스가 있는데, 터미널 영업시간을 기다리는 동안 터미널에 가거나 전화로 티켓을 주문해야 한다. 그래서 우리는 사람들이 일을 더 쉽게 할 수 있도록 고속버스 예매 앱과 유사한 앱을 만들고 싶다. 우리는 집이나 다른 곳에서 전화 앱을 통해 티켓을 더욱 쉽게 주문할 수 있게 만들고 싶다. 이 앱을 개발한 이유는 매우 크다. 기술이 발전함에 따라 사람들은 휴대폰과 같은 기계를 많이 사용해야 하기 때문이다. 그렇다면 우리 같은 앱 개발자에게는 새로운 시대가 도래했다고 할 수 있다. 사람들에게는 일상생활에서 많이 사용하는 유용한 앱이 필요하다. | | | |
|  | | | | |
| 조편성  색인번호  202311\_4100-05 | 구분 | 학 번 | 이 름 | 연락처 |
| 조장 | 20101256 | (오타벡) | 010-8007-7577 |
| 조원 | 20101296 | Akhmadalieva  Zilolakhon(질럴라) | 010-5949-1420 |
| 지도교수 | 차광호 | | 강좌번호 | 41003 |
| 202312\_41003-05  익스프레 버스  **(Express Bus)**   |  |  |  |  | | --- | --- | --- | --- | | 지도교수 | : | 차광호 | | | 제출일 | : | 2023년 12월 27일 | | | 조장 | : | 20101256 | 오타벡 | | 조원 | : | 20101296 | Akhmadalieva Zilolakhon(질럴라) | |  |  |  |  | |  |  |  |  |  |  |  | | --- | --- | | EMB0000bba87295 | **정보통신대학 컴퓨터공학과**  **서울과학기술대학교** |   **목차**     1. 프로젝트 개요...................................................................................................... 4 2. 프로젝트 목표 및 내용..................................................................................... 5 3. 요구 사항 분석.................................................................................................... 6 4. 시스템 구조........................................................................................................... 8 5. 서버 및 DataBase..............................................................................................10 6. 프로젝트 개발 및 운영 환경..........................................................................15 7. 프로젝트 구현 및 피드백................................................................................ 8. 시스템 다이어그램........................................................................................... 9. 앱 화면.................................................................................................................... 10. 프로그램 결과와 효과........................................................................... 11. 프로젝트 개선 및 회망 사항............................................................... 12. 부록 소스코드........................................................................................... 13. 참고 자료....................................................................................................       **1. 프로젝트 개요**  Express Bus(고속버스) - 온라인 버스 티켓 예매 애플리케이션은 사용자가 휴대폰 앱 또는 웹 플랫폼을 통해 버스 여행을 예약하고 티켓을 구매하는 서비스를 제공하는 소프트웨어입니다. 이 애플리케이션은 전통적인 방식의 티켓 구매를 대체하고 디지털 환경에서 편리하게 예매를 할 수 있도록 돕습니다.  한국에서 IT 분야는 너무 발전되고 있는 상태이라서 이런 애플리케이션들이 이미 많이 가발되고 사용 중이지만 우리 나라에서는 이직 한국처럼 편리한 온라인 버스표 예매 시스템이 발전되지 않는 상태입니다.그래서 우리팀은 우리나라에 존재하는 기술적인 문제들을 분석하여 개발자음으로서 이런 문제들을 해결하는데 기여하고 싶어서 Express Bus (온라인 버스 티켓 예매 애플리케이션은)을 프로젝트로 선택해서 개발하게 되었습니다.  온라인 버스 티켓 예매 애플리케이션은 사용자들에게 여러 편의성들을 제공할 수 있습니다.  사용자가 쉽게 탐색하고 원하는 정보를 빠르게 찾을 수 있는 직관적이고 사용하기 쉬운 디자인을 가지고 있습니다. 여러 가지 기준에 따라 노선, 출발 시간, 도착지 등을 검색하고 비교할 수 있는 다양한 필터 및 검색 옵션 제공하고 있습니다. 다양한 도시 간의 노선과 출발 시간을 제공하여 사용자가 여러 옵션 중 선택할 수 있도록 도와 줍니다. 온라인 버스 티켓 예매 애플리케이션은 신용카드 결제 수단을 통해 편리한 온라인 결제 서비스 제공하고 있습니다. 예매 후에는 티켓 예매 확인서를 제공합니다. 좌석맵을 통해 실시간으로 남은 좌석을 확인하고 사용자가 원하는 좌석을 선택할 수 있는 기능 제공과 결제 정보 및 개인 정보를 안전하게 보호하기 위해 암호화 및 보안 프로토콜 적용하면서 사용자에게 편리성을 제공할 수 있습니다.  이러한 기능들은 사용자에게 편리하고 빠른 예매 과정을 제공하여 여행 계획을 관리하는 데 도움이 되며, 버스 운영자에게는 효율적인 예매 및 운영 관리 도구를 제공합니다.  **2. 프로잭트 목표 및 내용**  온라인 버스 티켓 예매 애플리케이션을 개발하는 목표는 다양한 측면에서 사용자 및 운영자에게 효율적이고 편리한 서비스를 제공하는 데에 있습니다. 아래는 주요 목표들입니다:  **사용자 경험 개선:**   * **직관적이고 사용하기 쉬운 인터페이스:** 사용자가 쉽게 검색하고 예매할 수 있는 직관적이고 사용하기 편한 디자인을 제공하여 사용자 경험을 최적화합니다. * **다양한 기능 제공:** 다양한 노선, 좌석 선택, 가격 비교, 실시간 정보 제공 등 다양한 기능을 제공하여 사용자에게 다양한 선택지를 제공합니다.   **다양한 노선과 유연한 일정 제공:**   * **다양한 도시 간 노선 제공:** 사용자가 원하는 여행 목적지 간에 다양한 노선을 제공하여 여행 옵션을 확장합니다. * **다양한 출발 시간과 일정 제공:** 다양한 시간대와 일정을 제공하여 사용자가 여행 일정을 유연하게 계획할 수 있도록 합니다.   **안전하고 신속한 서비스 제공:**   * **실시간 정보 제공:** 실제 버스 운행 상태, 예상 도착 시간 등을 실시간으로 제공하여 사용자에게 정확한 정보를 전달합니다.   **편리한 결제 및 예매 관리:**   * **간편한 결제 수단 제공:** 간편한 결제 방법을 통합하여 사용자가 편리하게 결제할 수 있도록 합니다.   **효율적인 운영 및 관리 도구 제공:**  **- 운영자 대시보드 및 리포팅:** 운영자가 예매 상태, 수익 등을 효과적으로 모니터링할 수 있는 대시보드 및 리포팅 도구를 제공합니다.  - **자동화된 예매 관리 시스템:** 예매 및 좌석 관리를 효율적으로 수행하는 자동화된 시스템을 도입하여 운영 효율성을 높입니다.  이러한 목표를 달성함으로써 온라인 버스 티켓 예매 애플리케이션은 사용자와 운영자 모두에게 효과적이고 만족스러운 서비스를 제공할 수 있을 것입니다.  **요구 사항 분석**  온라인 버스 티켓 예매 애플리케이션을 개발하기 위해 요구 사항 분석이 중요합니다. 다양한 스테이크홀더들의 기대와 애플리케이션의 기능, 성능, 보안 등에 대한 요구 사항을 정확하게 수집하고 문서화해야 합니다.  **사용자 요구 사항:**  **- 사용자 로그인 및 등록: 사용자가 개인 정보 입력함으로써 시스템에 등록하여 로그인 할 수 있어야 한다.개인 정보 잘 못 입력한 경우에 로그인 못 하게 안전한 시스템을 제공해야 한다.**  **- 검색 및 예매:** 출발지, 도착지, 출발 일자, 시간 등을 기반으로 다양한 노선을 검색하고 예매할 수 있어야 합니다.  **- 좌석 선택:** 사용자는 원하는 좌석을 선택하고 예매할 수 있어야 합니다.  **- 간편한 결제 옵션:** 간편한 결제 수단을 지원하고 사용자에게 편리한 결제 경로를 제공해야 합니다.  **- 다양한 노선 및 일정:** 다양한 노선과 운행 일정을 제공하여 사용자가 선택할 수 있어야 합니다.  **- 실시간 정보 제공:** 실제 운행 중인 버스의 정보, 예상 도착 시간 등을 실시간으로 정보를 제공해야 합니다.  **- 온라인 티켓 확인 및 발급:** 예매 후에는 티켓 예매 확인서를 제공하고, 필요한 경우에는 이메일을 통해 티켓을 발급해야 합니다.  **- 보안:** 결제 정보와 개인 정보를 안전하게 보호해야 합니다.  **- 성능:** 빠른 검색 및 예매 프로세스를 제공하여 사용자 경험을 향상시켜야 합니다.  **- 직관적인 사용자 인터페이스:** 사용자가 쉽게 예매할 수 있는 직관적이고 사용하기 쉬운 디자인을 제공해야 합니다.  **운영자 요구 사항:**  **- 대시보드 및 리포팅:** 운영자는 예매 상태, 운행 중인 버스의 정보 다양한 정보 등을 모니터링할 수 있는 대시보드를 사용할 수 있어야 합니다.  **- 예매 및 좌석 관리:** 운영자는 예매 및 좌석 관리를 효과적으로 수행할 수 있는 도구를 제공받아야 합니다.  **- 기술 스택:** 개발에 사용될 프론트엔드 및 백엔드 기술 스택을 정의해야 합니다.  - **데이터베이스:** 여러 사용자 정보, 예매 기록 등을 안전하게 저장하기 위한 데이터베이스 시스템을 선택해야 합니다.  - **새로운 정보 입력 및 수정: 운영자가 새로운 버스 정보를 입력하여 정보를 수정할 수도 있어야 한다.**  **- 사용자 정보 모니터링: 운영자가 사용자 정보 예매 정보 등을 확인할 수 있어야 한다.**  **시스템 구조**  **그림 [1.1]**    **그림 [1.2]**    온라인 버스 티켓 예매 애플리케이션의 시스템 구조는 여러 구성 요소로 구성되어 있으며, 사용자 인터페이스, 서버 측 로직, 데이터베이스, 외부 서비스 등이 포함한다.  **사용자 인터페이스 (UI):**  **- 웹 애플리케이션 및 모바일 앱:** 사용자는 웹 브라우저를 통한 웹 애플리케이션 또는 휴대폰 앱을 통해 서비스에 접근합니다.  - **검색 및 예매 인터페이스:** 사용자는 출발지, 도착지, 일자, 시간 등을 입력하고 노선을 검색하며, 좌석을 선택하고 결제를 진행한다  **애플리케이션 서버 (Application Server):**  **- 비즈니스 로직 처리:** 예매, 결제, 좌석 관리, 알림 서비스 등과 같은 비즈니스 로직을 처리합니다.  - **사용자 및 예매 데이터 관리:** 사용자 프로필, 예매 정보 등을 데이터베이스에 저장하고 관리한다  **데이터베이스 (Database):**  **- 사용자 데이터베이스:** 사용자 계정 정보, 예매 내역, 프로필 등의 데이터를 저장한다.  - **좌석 및 노선 데이터베이스:** 좌석 정보, 노선 일정 및 가격 정보 등을 저장한다  **외부 서비스 및 API:**  **- 결제 게이트웨이:** 간편한 온라인 결제 수단을 처리하기 위한 외부 결제 서비스와의 통합이 이루어집니다.  **보안 층:**  **- SSL/TLS 프로토콜 사용:** 사용자 및 결제 정보의 안전한 전송을 보장하기 위해 SSL/TLS 프로토콜을 사용한다  - **인증 및 권한 관리:** 사용자 인증 및 권한 부여를 통해 시스템의 보안을 강화한다.  이러한 구조는 전반적인 시스템의 개요를 보여주며, 실제 구현은 프로젝트의 특정 요구 사항과 스케일에 따라 세부적으로 조정될 수 있습니다.  **서버 및 DataBase**  온라인 버스 티켓 예매 애플리케이션의 서버 및 데이터베이스 구성은 전체 시스템의 핵심이며, 효율적인 운영과 안정성을 보장하기 위해 신중한 설계가 필요한다  우리 Express Bus 프로젝트의 서버를 SetUp 할 때 Proxy(프록시) 서버를 이용했다.  [프록시 서버는 클라이언트가 자신을 통해서 다른 네트워크 서비스에 간접적으로 접속할 수 있게 해 주는 컴퓨터 시스템이나 응용 프로그램을 가리킨다. 서버와 클라이언트 사이에 중계기로서 대리로 통신을 수행하는 것을 가리켜 '프록시', 그 중계 기능을 하는 것을 프록시 서버라고 부른다. 프록시 서버 중 일부는 프록시 서버에 요청된 내용들을 캐시를 이용하여 저장해 둔다. 이렇게 캐시를 해 두고 난 후에, 캐시 안에 있는 정보를 요구하는 요청에 대해서는 원격 서버에 접속하여 데이터를 가져올 필요가 없게 됨으로써 전송 시간을 절약할 수 있게 됨과 동시에 불필요하게 외부와의 연결을 하지 않아도 된다는 장점을 갖게 된다. 또한 외부와의 트래픽을 줄이게 됨으로써 네트워크 병목 현상을 방지하는 효과도 얻을 수 있게 된다.](https://www.bing.com/ck/a?!&&p=4ccdd9f64b459ee4JmltdHM9MTcwMzM3NjAwMCZpZ3VpZD0xZDc5MDhiZC1iNTZhLTY2NjItMjhhNS0xYjhiYjQ0MjY3YTQmaW5zaWQ9NTcwNQ&ptn=3&ver=2&hsh=3&fclid=1d7908bd-b56a-6662-28a5-1b8bb44267a4&psq=what+is+proxy+server&u=a1aHR0cHM6Ly9rby53aWtpcGVkaWEub3JnL3dpa2kvJUVEJTk0JTg0JUVCJUExJTlEJUVDJThCJTlDXyVFQyU4NCU5QyVFQiVCMiU4NA&ntb=1" \o "ko.wikipedia.org" \t "_blank)  **그림 [2.1] 그림 [2.2]**    **그림 [2.3] Proxy Server SetUp**    **서버 구성:**  **- 웹 서버(Web Server):**   * 사용자와 애플리케이션 서버 간의 통신을 담당한다 * 클라이언트의 요청을 받아 애플리케이션 서버로 라우팅한다 * 클라이언트가 여청하면 항상 요청 status, success 및 data 를 제공한다 * 정적 파일 (이미지, CSS, JavaScript 등)을 제공한다   **- 애플리케이션 서버(Application Server):**   * 비즈니스 로직을 처리하고 클라이언트 요청에 대한 응답을 생성한다 * 예매, 결제, 좌석 관리, 알림 서비스 등의 핵심 비즈니스 기능을 담당한다 * 외부 서비스 및 데이터베이스와의 통합을 담당한다   **- 보안 층 (Security Layer):**   * SSL/TLS 프로토콜을 사용하여 데이터의 암호화를 수행한다. * 사용자 password가 HashedPassword으로 보안되어 있다 * 사용자 인증 및 권한 부여를 담당하여 시스템 보안을 강화한다.   **데이터베이스 구성:**  우리 프로젝트의 데이더베이스를 설계하는데 MongoDb를 선택했다.  **그림 [3.1]**    **그림 [3.2]**    몽고DB(MongoDB←HUMONGOUS)는 크로스 플랫폼 도큐먼트 지향 데이터베이스 시스템이다. NoSQL 데이터베이스로 분류되는 몽고DB는 JSON과 같은 동적 스키마형 도큐먼트들(몽고DB는 이러한 포맷을 BSON이라 부름)을 선호함에 따라 전통적인 테이블 기반 관계형 데이터베이스 구조의 사용을 삼간다. 이로써 특정한 종류의 애플리케이션을 더 쉽고 더 빠르게 데이터 통합을 가능케 한다. 아페로 GPL과 아파치 라이선스를 결합하여 공개된 몽고DB는 자유-오픈 소스 소프트웨어이다.  뉴욕시에 기반을 둔 회사인 10gen (현재의 몽고DB)에서 2007년 10월, 계획된 PaaS(서비스형 플랫폼) 제품의 구성 요소로 처음 개발하였으며 10gen이 상용 지원 및 기타 서비스를 제공한 2009년에 오픈 소스 개발 모델로 전향하였다. 그 뒤로 몽고DB는 크레이그리스트, 이베이, 포스퀘어, 소스포지, 뉴욕 타임즈, 구글, 페이스북와 같은 수많은 주요 웹사이트 및 서비스에 백엔드 소프트웨어로 채택되고 있다. 몽고DB는 가장 유명한 NoSQL 데이터베이스 시스템이다.  **사용자 데이더베이스:**   * 사용자 정보를 저장하는데 사용한다. * 사용자 계정, 프로필 정보 등을 보관한다. * 사용자가 운영자 여부도 저장한다. * 사용자가 운영자한테 Block 여부도 저장한다   **그림 [4.1] DataBase User Schema**    **Bookings DataBase Schema:**   * 버스 정보, 예매한 사용자 정보 등 저장한다 * 좌석 정보, 노선 일정, 가격 정보 등을 저장한다. * 좌석 예매 상태를 추적하고, 노선 및 일정 정보를 제공한다 * Transaction 정보 생선된 정보 등 저장한다.   **그림 [4.2]**  **Buses DataBase Schema:**   * 좌석 정보, 노선 일정, 가격 정보 등을 저장한다. * 좌석 예매 상태를 추적하고, 노선 및 일정 정보를 제공한다 * 버스 정보와 출발 ,도착 시간 등 정보 저장한다. * 예매 된 좌석 정보 버스 status 정보 저장한다.   **그림 [4.3]**    **프로젝트 개발 및 운영 환경**     이 앱을 개발한 이유는 매우 크다. 기술이 발전함에 따라 사람들은 휴대폰과 같은 기계를 많이 사용해야 하기 때문이다. 그렇다면 우리 같은 앱 개발자에게는 새로운 시대가 도래했다고 할 수 있다. 사람들에게는 일상생활에서 많이 사용하는 유용한 앱이 필요하다.  고속버스 예매앱의 경우, 우즈베키스탄에도 한국과 마찬가지로 공휴일이 많다. 이때 가족과 멀리 떨어져 일하고 공부하는 사람들은 명절을 맞아 가족에게로 돌아온다. 가장 많이 사용되는 교통수단은 기차, 버스, 택시입니다. 그러나 지금까지 고속버스 이용자 수는 다른 교통수단보다 많다. 휴대폰으로 쉽게 티켓을 주문하면 매우 쉽다  온라인 버스 티켓 예매 애플리케이션의 개발 및 운영 환경은 안정성, 확장성, 보안성 등을 고려하여 설계되어야 합니다. 아래는 이러한 환경을 위한 주요 고려 사항입니다.  **프론트엔드 기술:**  우리가 이 앱을 개발하기 위해서 JavaScript를 선택했다 .JavaScript는 웹 기반 Application 개발하는데 King라고 해도 되기 때문이다.  **그림 [5.1]**  **자바스크립트**는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용되며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 또한 Node.js와 같은 런타임 환경과 같이 서버 프로그래밍에도 사용되고 있다. 자바스크립트는 본래 넷스케이프 커뮤니케이션즈 코퍼레이션의 브렌던 아이크가 처음에는 모카라는 이름으로, 나중에는 라이브스크립트라는 이름으로 개발하였으며, 최종적으로 자바스크립트가 되었다. 자바스크립트가 썬 마이크로시스템즈의 자바와 구문이 유사한 점도 있지만, 이는 사실 두 언어 모두 C 언어의 기본 구문에 바탕을 뒀기 때문이고, 자바와 자바스크립트는 직접적인 연관성은 약하다. 이름과 구문 외에는 자바보다 셀프나 스킴과 유사성이 많다. 자바스크립트는 ECMA스크립트의 표준 사양을 가장 잘 구현한 언어로 인정받고 있으며 ECMAScript 5까지는 대부분의 브라우저에서 기본적으로 지원되었으나 ECMAScript 6 이후부터는 브라우저 호환성을 위해 트랜스파일러로 컴파일된다.  JavaScript FrontEnd 개잘하는데 다양한 Framwork tools 갖고 있다. 그들은 ReactJs,NextJs,AngularJS,VuiJs 등 유영한 것등이 있다. 우리는 그 중 ReactJS 를 선택했다.  **ReactJS**  **그림 [5.2]**  https://lh7-us.googleusercontent.com/64H2R2KulS92IYnHoSSTbSGuDma_cK_4k6neuVcvWbHnN1nIPwi4m1cX5bv7g7g4cjdQta2G_tIS_v1f9uwhAo6peRJ1s3CzdqyZrAmpHxwnFQbNGBhPZbEz8XGkmgoXwNit9aZO-1xLNtl4ISfOeGk **React JS**는 자바스크립트 라이브러리의 하나로서  사용자 인터페이스를 만들기 위해 사용된다. 페이스북과 개별 개발자 및 기업들 공동체에 의해 유지보수된다.  JSX(JavaScript XML)는 XML같은 문법을 사용하는 ECMAScript의 확장이다.HTML과 모습이 비슷해 보이는 JSX는 수많은 개발자들에게 친숙한 문법을 사용하여 컴포넌트 렌더링을 구조화하는 방법을 제공한다. 리액트 컴포넌트들은 일반적으로 JSX를 사용하여 작성되지만 꼭 그렇게 할 필요는 없다. (컴포넌트들은 순수 자바스크립트로 작성할 수도 있다) JSX는 PHP를 위해 페이스북이 개발한 다른 확장 문법인 XHP와 유사하다.  UI를 디자인하는데 우리 팀은 ReactJS와 함께 ANT Design도 이용하기로 했습니다.  **ANT Design:**  **그림 [5.3**  **Ant Design**은 대화형 사용자 인터페이스를 구축하는 데 유용한 사용하기 쉬운 구성 요소가 포함된 [React.js](https://www.geeksforgeeks.org/reactjs-tutorials/) UI 라이브러리입니다. 사용과 통합이 매우 쉽습니다. react를 사용하여 웹 애플리케이션을 디자인하는 현명한 옵션 중 하나입니다. 쉽게 사용할 수 있는 고품질 구성 요소를 제공합니다.  **Axios:**  **그림 [5.4**  우리 팀은 웹 브라우저에서 HTTP 요청을 위한 JavaScript Library Axios을 쓰기로 했다. React에서 백엔드 통신은 일반적으로 HTTP 프로토콜을 사용하여 이루어집니다. 많은 개발자가 XML HTTP 요청 인터페이스와 HTTP 요청을 위한 Fetch API에 익숙하지만 프로세스를 더욱 단순화하는 Axios라는 또 다른 강력한 라이브러리가 있습니다. 널리 사용되는 라이브러리인 Axios는 주로 REST 엔드포인트에 비동기 HTTP 요청을 보내는 데 사용됩니다. 이 라이브러리는 CRUD 작업을 수행하는 데 매우 유용합니다.   1. 이 인기 있는 라이브러리는 백엔드와 통신하는 데 사용됩니다. Axios는 JS ES6에 기본으로 제공되는 Promise API를 지원합니다. 2. Axios를 사용하여 애플리케이션에서 API 요청을 수행합니다. 요청이 이루어지면 반환으로 데이터를 얻은 다음 프로젝트에서 이 데이터를 사용합니다. 3. 이 라이브러리는 개발자들 사이에서 매우 인기가 있습니다. GitHub에서 확인하면 78,000개의 별을 찾을 수 있습니다.   **React Router vs. React Router DOM**  **그림 [5.5]**    우리 Express Bus Page Navigation 을 위한 React Router vs. React Router DOM을 사용하였다.왜냐하면 라우팅은 사용자 요청과 작업을 기반으로 웹사이트의 페이지를 탐색하는 데 필수적인 기술이다. React Router 라는 별도의 라이브러리를 사용하면 React 애플리케이션에서 라우팅을 활성화하고 애플리케이션에서 여러 경로를 정의할 수 있다. 그러나 React-router 또는 React-router-dom 패키지를 설치할지 여부는 혼란스러울 수 있습니다. 이 기사에서는 차이점과 어떤 패키지를 어디에 사용해야 하는지 분석하여 이러한 혼란을 해결한다  **Why Is React Router Needed?**  React는 단일 페이지 애플리케이션을 구축하는 데 이상적인 유명한 JavaScript 프레임워크이다. 웹 사이트 구축을 위한 최고의 솔루션 중 하나이지만 React에는 기본적으로 많은 고급 기능이나 라우팅이 포함되어 있지 않다. 따라서 React Router는 이러한 단일 페이지 애플리케이션이 여러 뷰를 렌더링하기 위한 탁월한 탐색 선택이다.  **What is React Router?**  React Router는 React 애플리케이션의 다양한 뷰 구성 요소 간 라우팅을 위한 널리 사용되는 표준 라이브러리이다. 이는 사용자 인터페이스를 URL과 동기화하는 데 도움이 된다. 또한 React Router를 사용하면 지정된 URL에 대해 표시할 보기를 정의할 수 있다. React Router와 관련된 세 가지 주요 패키지는 다음과 같다.  React-router : 경로 일치 알고리즘 및 후크를 포함하여 React Router의 핵심 기능을 포함하다.  React-router-dom : 반응 라우터 의 모든 것을 포함 하고 몇 가지 DOM 관련 API를 추가항다  React-router-native : React-Router 의 모든 것을 포함 하고 몇 가지 React Native 관련 API를 추가항다.  **React Router DOM이란 무엇인가?**  React-Router-dom 의 주요 기능은 웹 애플리케이션에서 동적 라우팅을 구현하는 것이다. 플랫폼과 애플리케이션의 요구 사항을 기반으로, React-router-dom은 구성 요소 기반 라우팅을 지원한다. 이는 React 애플리케이션이 브라우저에서 실행 중인 경우 라우팅에 이상적인 솔루션이다.  **React Redux:**  **그림 [5.6]**    **React Redux**는 Redux의 공식 React 바인딩이다. 이를 통해 React 구성 요소는 Redux Store에서 데이터를 읽고 Action을 Store에 전달하여 데이터를 업데이트할 수 있다. Redux는 단방향 데이터 흐름 모델을 통해 상태를 관리하는 합리적인 방법을 제공하여 앱의 확장을 돕는다.  **React Redux Toolkit:**  **그림 [5.7]**    Redux Toolkit은 효율적인 Redux 개발을 위한 저희의 견해를 반영한, 이것만으로도 작동하는 도구 모음이다. Redux Toolkit은 Redux 로직을 작성하기 위한 표준 방식이 되도록 만들어졌고, 사용하기를 강력히 추천한다.  이 안에는 저장소 준비, 리듀서 정의, 불변 업데이트 로직, 액션 생산자나 액션 타입을 직접 작성하지 않고도 전체 상태 "조각"을 만들어내는 기능까지 대부분의 Redux 사용 방법에 해당하는 유틸리티 함수들이 들어 있다. 거기다가 비동기 로직을 위한 Redux Thunk와 셀렉터 작성을 위한 Reselect 등의 널리 사용되는 애드온을 포함하고 있어 이들을 제대로 사용할 수 있게 해준다.  위에서 다양한 우리 프로젝트 개발하는데 사용하던 Library and Framework들을 설명하면서 기술해 봤다. 사실 그거들 빼고도 여러 Library들을 사용하였다. 우리 프로젝트 소스코드 파일 들어가서 client 펄더 열어서 package.json 파일 열고 dependencies 부분을 보면 확인할 수 있다.  **그림 [5.8]**    **백엔드 기술**  아시다시피 우리 Express Bus 프로젝트를 개발하는데 JavaScript를 선택했다고 위에서 언급했다. JavaScript로 백엔드 하려면 NodeJS 베스트라고 하면 된다.  **Node.js**는 크로스플랫폼 오픈소스 자바스크립트 런타임 환경으로 윈도우, 리눅스, macOS 등을 지원한다. Node.js는 V8 자바스크립트 엔진으로 구동되며, 웹 브라우저 바깥에서 자바스크립트 코드를 실행할 수 있다.  주로 확장성 있는 네트워크 애플리케이션과 서버 사이드 개발에 사용되는 소프트웨어 플랫폼이며, 논블로킹(Non-blocking) I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능을 가지고 있다.    내장 HTTP 서버 라이브러리를 포함하고 있어 웹 서버에서 아파치 등의 별도의 소프트웨어 없이 동작하는 것이 가능하며, 이를 통해 웹 서버의 동작에 있어 더 많은 통제를 가능케 한다.  **그림 [6.1]**     2009년 5월 27일 처음 소개된 Node.js는 오픈 소스 JavaScript 엔진인 크롬 V8에 비동기 이벤트 처리 라이브러리인 libuv를 결합한 플랫폼이다. 다시 말해, JavaScript로 브라우저 밖에서 서버를 구축하는 등의 코드를 실행할 수 있게 해주는 런타임 환경이다. Ryan Dahl이 처음 개발했으며, 처음엔 리눅스와 macOS만 지원되었으나 2011년 7월에 Windows 버전도 발표되었다.  2014년 12월 한때 Node.js의 포크인 io.js가 나타나면서 Node.js 0.12 버전, io.js 3.3 버전까지 서로 분열된 모습으로 이어지는 듯했지만, 2015년 9월에 Node.js 4.0 버전으로 병합되어 현재에 이르렀다.   대표적인 방법은 홈페이지에 대문짝만하게 나와있는 다운로드 버튼을 누르는 것이다. 다만 Node.js를 이용해 구현된 프로그램 혹은 웹사이트를 실행하거나 빌드하다 보면 프로그램마다 다른 노드 버전을 지원하는 것을 경험할 수 있는데, 이 때는 NVM(Node Version Manager)을 이용해서 쉽게 다른 버전의 node.js를 설치하고 전환할 수 있다. 사실 이 목적이 아니라도, node.js를 쉽게 업데이트하기 위해 NVM을 쓰게 된다.  **데이터베이스**  어떤 앱이나 시스템 개발하든 백엔드 데이더베이스가 너무 중요한다.우리는 데이더 베이스 관리하는데 MongoDB를 활용했다.  **그림 [6.2]**  **몽고DB(MongoDB**←HUMONGOUS)는 크로스 플랫폼 도큐먼트 지향 데이터베이스 시스템이다. NoSQL 데이터베이스로 분류되는 몽고DB는 JSON과 같은 동적 스키마형 도큐먼트들(몽고DB는 이러한 포맷을 BSON이라 부름)을 선호함에 따라 전통적인 테이블 기반 관계형 데이터베이스 구조의 사용을 삼간다. 이로써 특정한 종류의 애플리케이션을 더 쉽고 더 빠르게 데이터 통합을 가능케 한다. 아페로 GPL과 아파치 라이선스를 결합하여 공개된 몽고DB는 자유-오픈 소스 소프트웨어이다.  뉴욕시에 기반을 둔 회사인 10gen (현재의 몽고DB)에서 2007년 10월, 계획된 PaaS(서비스형 플랫폼) 제품의 구성 요소로 처음 개발하였으며 10gen이 상용 지원 및 기타 서비스를 제공한 2009년에 오픈 소스 개발 모델로 전향하였다. 그 뒤로 몽고DB는 크레이그리스트, 이베이, 포스퀘어, 소스포지, 뉴욕 타임즈, 구글, 페이스북와 같은 수많은 주요 웹사이트 및 서비스에 백엔드 소프트웨어로 채택되고 있다. 몽고DB는 가장 유명한 NoSQL 데이터베이스 시스템이다.  **MongoDB**는 NoSQL로 분류되는 크로스 플랫폼 도큐먼트 지향 데이터베이스 시스템이다. MySQL 처럼 전통적인 테이블-관계 기반의 RDBMS가 아니며 SQL을 사용하지 않는다.  이름의 mongo는 humongous를 줄인 표현이다. 즉 '매우 큰 DB'라는 뜻.  https://lh7-us.googleusercontent.com/r0Qd4tgjC95IR9JVeYmu9Aup-NzcW30kljq3z0w5rV4JoNNOFK693TkaPaoaVaGxlosykuyppg6HoM8hWpT0yz1C59XumhwUE7ApmB29TSYsiKACPp42QE2J5WPRO0z-y8WXNteY4OBrmCQF3e5gr1I  [6-1]  자바스크립트 런타임으로는 모질라의 SpiderMonkey를 사용한다.  2018년까지는 GNU Affero General Public License(AGPL)에 기반하여 공개되었다. 그러나 클라우드 업체들이 MongoDB를 그대로 이용하면서 라이선스 계약 없이 수익을 챙기고, 오픈소스 커뮤니티에는 기여를 하지 않자 2018년 10월 라이선스를 AGPL에 기반한 Server Side Public License(SSPL)로 변경했다. SSPL은, 서버에서 프로그램을 실행할 경우 기존 AGPL에서는 라이선스 대상 프로그램의 소스코드만을 공개하면 되었던 것을 수정하여, 프로그램이 'Software as a Service' 형태로 제공될 경우 그 프로그램과 연동되어 실행되는 관리 소프트웨어 등을 모두 공개해야 한다는 조항으로 바꾸었다. SSPL은 오픈소스 커뮤니티에서 라이선스가 다른 프로그램에까지 영향을 미치는 것에 대해 비판을 받았고 오픈소스 이니셔티브에서는 라이선스 승인 절차를 중단했다.    MongoDB와 타사 데이터베이스 비교: 현재 상당수의 데이터베이스 관리 솔루션이 시중에 제공되어, 각 기업에 적합한 솔루션을 선택하는 것이 쉽지 않은 경우가 많다. 아래 제시된 몇 가지 일반적인 솔루션별 비교사항과 우수 사용 사례는 기업의 의사결정에 도움을 준다.  모바일 애플리케이션     MongoDB의 JSON 문서 모델을 활용하면 클라우드 기반 스토리지 솔루션뿐만 아니라 Apple iOS 및 Android 디바이스를 비롯한 백엔드 애플리케이션 데이터를 필요한 곳 어디에나 저장할 수 있다. 이러한 유연성은 보조 색인 및 지리 공간 색인 작성을 통해 여러 환경에서 데이터를 집계하도록 지원하는 것은 물론, 모바일 애플리케이션을 원활히 확장할 수 있는 기능을 개발자에게 제공한다.  **다중 언어 지원:**     Mongo DB의 탁월한 특징 중 하나는 바로 다중 언어를 지원하는 기능이다. Mongo DB는 이미 여러 버전으로 출시된 바 있으며 Python, PHP, Ruby, Node.js, C++, Scala, JavaScript 등 대중적인 프로그래밍 언어에 대한 드라이버 지원을 통해 지속적인 개발을 수행하고 있다.  Mongo DB 배포 및 설정  배포 과정은 두 가지 주요 활동을 수반합니다. 바로 MongoDB를 설치하고, 데이터베이스를 생성하는 과정이다.  MongoDB 설치하기   * Windows:Windows 환경에서 MongoDB를 설치하려면, Windows Server 2008 R2 또는 Windows Vista 이상을 실행한다. 사용할 데이터베이스 아키텍처 유형을 결정하면,MongoDB 다운로드 페이지에서 최신 버전의 플랫폼을 다운로드할 수 있다. * Mac: MacOS에 MongoDB를 설치하는 경우, 다음 두 가지 접근 방식을 활용할 수 있다. Windows 기반 환경의 설치 프로세스와 마찬가지로, 사용할 빌드 유형을 결정하면 개발자 웹사이트에서 MongoDB를 직접 설치할 수 있다. 하지만 Mac에서 보다 쉽게 * MongoDB를 설치하고 실행하는 일반적인 방법은 Homebrew를 실행하는 터미널 앱을 이용하는 것이다.   **ExpressJS**  우리 프로젝트 API 개발을 위해서 ExpressJS를 사용하였다.왜냐하면 플러그인으로 이용 가능한 수많은 기능들을 갖추고 있다.  **그림 [6.3]**  Express.js, 또는 간단히 익스프레스(Express)는 Node.js를 위한 웹 프레임워크의 하나로, MIT 허가서로 라이선스되는 자유-오픈 소스 소프트웨어로 출시되었다. 웹 애플리케이션, API 개발을 위해 설계되었다. Node.js의 사실상의 표준 서버 프레임워크로 불리고 있다.원 제작자 TJ Holowaychuk는 이 소프트웨어를 시나트라에 영향을 받은 서버로 기술하고 있으며 이는 플러그인으로 이용 가능한 수많은 기능들을 갖추면서 크기는 상대적으로 최소한임을 의미한다. 익스프레스는 몽고DB 데이터베이스 소프트웨어, AngularJS 프론트엔드 프레임워크와 함께 MEAN 스택의 백엔드 구성 요소이다.  **JSON wep token :** 서버와 안전적으로 정보 주고받기 위해서 JSWOn Wep Token을 사용하기로 했다. JSON 웹 토큰(JSON Web Token, JWT, "jot”은 선택적 서명 및 선택적 암호화를 사용하여 데이터를 만들기 위한 인터넷 표준으로, 페이로드는 몇몇 클레임(claim) 표명(assert)을 처리하는 JSON을 보관하고 있다. 토큰은 비공개 시크릿 키 또는 공개 비공개 키를 사용하여 서명된다. 이를테면 서버는 "관리자로 로그인됨"이라는 클레임이 있는 토큰을 생성하여 이를 클라이언트에 제공할 수 있다. 그러면 클라이언트는 해당 토큰을 사용하여 관리자로 로그인됨을 증명한다. 이 토큰들은 한쪽 당사자의 비공개 키(일반적으로 서버의 비공개 키)에 의해 서명이 가능하며 이로써 해당 당사자는 최종적으로 토큰이 적법한지를 확인할 수 있다. 일부 적절하고 신뢰할만한 수단을 통해 다른 당사자가 상응하는 공개키를 소유하는 경우 이 경우 또한 토큰의 적법성 확인이 가능하다. 토큰은 크기가 작고 URL 안전으로 설계되어 있으며 특히 웹 브라우저 통합 인증(SSO) 컨텍스트에 유용하다. JWT 클레임은 아이덴티티 제공자와 서비스 제공자 간(또는 비즈니스 프로세스에 필요한 클레임)의 인가된 사용자의 아이덴티티를 전달하기 위해 보통 사용할 수 있다.  **Nodemailer:** 프로젝크트 너티피케이션을 위한 우리 Nodemailer사용하였다. Nodemailer는 Node.js 애플리케이션에서 이메일을 쉽게 보낼 수 있는 Node.js 애플리케이션용으로 널리 사용되는 npm 모듈이다. Nodemailer에 대한 몇 가지 주요 사실Nodemailer는 npm 레지스트리를 통해 사용할 수 있는 오픈 소스 Node.js 모듈이다. 배달을 위해 SMTP 서버에 연결하여 Node.js 응용 프로그램에서 이메일을 보낼 수 있다. Nodemailer는 Gmail, Yahoo, SendGrid, MailGun 등과 같은 다양한 이메일 서비스에 연결하고 인증하는 방법을 알고 있는 "전송"을 사용하여 작동한다.HTML/텍스트 이메일, 첨부 파일, 템플릿, 대량 전송 등과 같은 기능을 지원한다.Nodemailer는 MIME 형식, SMTP 연결/인증 등과 같은 이메일 전송의 모든 복잡성을 처리한다.  **Nodemon:** 개잘하다가 코드 수정하면 자동으로 재실행해 주는데 우리에 Nodemon 도움이 너무 컸다. nodemon은 디렉토리의 파일 변경이 감지되면 노드 애플리케이션을 자동으로 다시 시작하여 Node.js 기반 애플리케이션을 개발하는 데 도움이 되는 도구이다. Nodemon은 코드나 개발 방법을 추가로 변경할 필요가 없다. nodemon은 의 대체 래퍼입니다. 를 사용하려면 스크립트를 실행할 때 명령줄에서 단어를 바꾼다.  **Payment service provider:** 우리 프로젝트 Payment 결제 시스템을 설정하는데 Stripe Payment service provider를 사용했다. Stripe는 판매자가 신용 카드 및 직불 카드 또는 기타 결제를 수락할 수 있도록 하는 결제 서비스 제공업체이다. 결제 처리 솔루션인 Stripe Payments는 대부분의 고유한 기능이 주로 온라인 판매에 맞춰져 있기 때문에 대부분의 온라인 판매를 온라인으로 하는 비즈니스에 가장 적합한다.  **UUID:** 사용자 ID를 보안하는데 UUID 를 사용하였다.UUID는 사용자 ID 및 보안은 최신 애플리케이션을 빌드하는 데 중요한 요소이며, 사용자 ID의 보안을 보장하기 위해 많은 조치가 마련되어 있다.인터넷에서 사용자를 식별하는 한 가지 방법은 UUID 또는 범용 고유 식별자이다. Microsoft의 GUID(Globally Unique Identifiers)와 마찬가지로 UUID는 인터넷에서 엔터티를 고유하게 식별하는 데 널리 사용되는 고유한 128비트 값이다.  UUID는 어떻게 작동한가?  IETF(Internet Engineering Task Force)는 RFC 4122의 UUID 프로토콜을 "공간과 시간에 걸쳐 고유성을 보장할 수 있는 128비트 길이의 식별자"로 정의한다.다른 UUID 버전의 생성은 다른 알고리즘과 방법으로 발생한다. v1 UUID는 생성 컴퓨터의 타임스탬프 및 Mac 주소를 사용하여 식별하는 반면, v4 UUID는 생성하는 웹 사이트에 따라 난수 생성기를 사용한다.  위에서 설명했던 Framework and Library들 빼고 또 몇 가지를 사용했는데 우리 프로젝트 소스코드 펄더 열고 package.json 파일을 열어보면 dependencies 부분에서 확인이 가능하다. **그림[6.4]**    **GIT and GitHub**  개발함으로써 누구든지 알아야 될 Tool 중 중요한 것들은 바로 Git and GitHub이다. 코드 저장하고 수정하고 Source control 하는데 너무 중요한 것들이다. 우리도 켑스톤 디자인 프로젝트 소스코드 컨트럴하는데 Git와 GitHub를 사용하였다.  **그림 [6.5]**  **Git:** Git은 소프트웨어 개발 중 소스 코드의 변경 사항을 추적하기 위한 분산 버전 제어 시스템이다. 프로그래머 간의 작업을 조정하기 위해 설계되었지만 모든 파일 집합의 변경 내용을 추적하는 데 사용할 수 있다. 그 목표에는 속도, 데이터 무결성 및 분산된 비선형 워크플로에 대한 지원이 포함된다.  **GitHub:** GitHub는 Git의 모든 분산 개정 제어 및 소스 코드 관리(SCM) 기능을 제공하고 자체 기능을 추가하는 웹 기반 Git 리포지토리 호스팅 서비스이다.  **IDE (Integrated Development Environment): IDE(통합 개발 환경)는 일반적으로 사용되는 개발자 도구를 소형 GUI(그래픽 사용자 인터페이스) 응용 프로그램에 결합하는 소프트웨어이다. 코드 편집기, 코드 컴파일러 및 코드 디버거와 같은 도구와 통합 터미널의 조합이다**  **우리 팀은 프로젝트 코드를 작성하고 테스트, 데부깅와 실행을 위해 VS Code 를 사용했다. 비주얼 스튜디오 코드(영어: Visual Studio Code) 또는 코드(Code)는 마이크로소프트가 마이크로소프트 윈도우, macOS, 리눅스용으로 개발한 소스 코드 편집기이다. 디버깅 지원과 Git 제어, 구문 강조 기능, SSH 접속 등이 포함되어 있으며, 사용자가 편집기의 테마와 단축키, 설정 등을 수정할 수 있다.**  **그림 [6.6]**  **비주얼 스튜디오 코드는 깃허브가 개발한 일렉트론 프레임워크를 기반으로 구동된다. 그러나 같은 일렉트론 기반의 편집기 아톰을 포크한 것은 아니며, 비주얼 스튜디오 온라인 에디터(코드명 "모나코")를 기반으로 개발되었다.**  **Supported Browsers and Devices**  **우리 프로젝트 Webbased Application이라서 Devices system 큰 성능을 요청하지 않는다. IOS ,Windows Android와 Ipad 등 다 쉽게 작동한다. 기본적으로 생성된 프로젝트는 모든 최신 브라우저를 지원한다. Internet Explorer 9, 10 및 11을 지원하려면 폴리필이 필요한다. 이전 브라우저를 지원하는 폴리필 세트의 경우 react-app-polyfill을 사용한다.**  **남어지는 아래 그림 참고:**  **그림 [6.7]**    **1. 부록 소스코드**      지식은 무제한이다. 유학의 가장 좋은 점 중 하나는 우리나라 발전에 기여할 수 있다는 점입니다. 그리고 한국에 와서 공부해야 할 것이 많다는 것도 깨달았습니다. 그 중 하나가 기술이다.    국가가 발전할수록 도로의 편의성이 중요해진다. 이런 장거리 차량은 지하철, 버스와는 다르다. 이런 것들은 한국에서 매우 잘 발달되어 있다. 그 중 하나가 고속버스다. 특히, 터미널에 가지 않고도 버스표 가격을 확인할 수 있는 기능이 있어 개인이 더욱 편리하게 이용할 수 있다. 우즈베키스탄에도 고속버스가 있는데, 터미널 영업시간을 기다리는 동안 터미널에 가거나 전화로 티켓을 주문해야 한다. 그래서 우리는 사람들이 일을 더 쉽게 할 수 있도록 고속버스 예매 앱과 유사한 앱을 만들고 싶다. 우리는 집이나 다른 곳에서 전화 앱을 통해 티켓을 더욱 쉽게 주문할 수 있게 만들고 싶다.      앱 이미지는 이 사진하고 비슷할 예정이다.  앱 이름을 “Express bus” 이다.  **3.  시스템 다이어그램**  프로그램 계획은 이런 방식으로 갈 예정이다:  https://lh7-us.googleusercontent.com/_bXwZW4DsBMMcUVKlzKAcOVsse0ToZ4VjX1jp18rNecBInropwbU8wByv6YdWUo1FM9b9Z2-CBtUqJjiBJSj0GIITD-RjQTmCLWAWeP0q_xURR5eglvzpxm0Wpc27HkpJQUVmEysJERpiIbScebQThU    [7-1]  프로그래밍 계획을 우리가 상의했고, 우리 나라에 무엇이 필요하는지를 알고 있으니까 미래 모든 것들을 생각해 놓다.  **4. 프로그램 결과와 효과**      결과는 아주 잘 될 예정이다. 우리 우즈베키스탄 에서도 티켓을 변하게 주문할 수 있다면 누구나 좋아할 것 같다.      고속버스 티머니 앱 제공기능 들:   1. 비회원도 예매가 가능하다. 2. 가드 결제 가능하다. 3. 버스 스케줄, 소요시간, 출발지/도착지, 가까운 터미널 등을 이용할 수 있다. 4. 모바일 티켓이랑 셀프 체크인 서비스를 이용할 수 있다. 5. 실시간으로 도착 예정시간을 알아볼 수 있다.    이용방법:   1. 모바일 승차권: 터미널 매표소 갈 필요없이 QR 바코드 등을 버스 단말기에 스캔해서 탑승한다. 2. 현장발권: 출발지/도착지 배차를 선택하고 현자알행이 확인되는 경우 터미널 매표소나 무인발매기를 통해서 결제카드나 예매할 때 개인정보를 알려주면 티켓을 받을 수 있다. 버스 출발 10분 전까지 발권이 가능하다.   **5.  부록 소스코드**   우리가 준비하고 있던 코드들을 제시한다. 우선 우리가 먼저 해야 하는 단개들을 하나한식 알려준다. 먼저 github에서 개전을 열고 로그인으로 들어간다.  https://lh7-us.googleusercontent.com/iGtgqWiVxhaAtoE2CdlU6KxeODHMx0WxptBd9wTryFP47v_JP_8xb3tZebSiD-JKdYebfeAHu7kjw_FvRCav_tZGdF-ddZ6CAz50ZwgC3KCrpp7wlt5NcaAx86bEW9INb5y6jAH3E8miSDltyUfzExM  [8-1]  들어가고 나서 내가 팔러브하는 사람을 찾는다. 찾고 code를 클릭하고 들어간다.  https://lh7-us.googleusercontent.com/AMYrh1wsJBhdYnoOk8FIx7-Pw24aGHikkrxv029xkRYdbcHYtgbghHakpKOqFJVqHSz7gIyt3QY4Dsbvx-38nU-HYpYn5t0zr1_ZijKhVgmJvkKUWFiKJ-lAgQsQKR37ECYMJpzKWnt6aRyr7RLv36o  [8-2]  코드 부분에 있는 파일을 다운로드 한다.zip으로 다운로드된다.https://lh7-us.googleusercontent.com/RYdpmaLl9uu-g7BEziS_byORZXL9vx78SAZRe7Mx-h58P6foVllwKS9cp1GvLgblGjybBIP-_2ug0_W_FFShk2UCWkwpL5YykXsorD1WWNa_wp2CdJ5avD7eN9wfx1QckrIcN_PC_byLbZszAQ98RPI  [8-3]  https://lh7-us.googleusercontent.com/Ron3UjGuwHCgoH4u5uK_G1VDbpFJ_h4LAiqfhBCMX5mzsXK_-JPca9riukt7voFB4IOwle-TS2QHQrznfwdxKaC98E78M3fNATah5AMATQrk2h3rfcFAwxG4LarAetYhL439ZBF_2mBHhfYOAAUGXn0 [8-4]  다운로드 된 파일을 원하는 대에다가 저장하면 된다. 찾기 쉬워야 되서 바탕 화면에 저장한다.  visual studio에 들어가 코드를 실행시킨다.terminal 부분에서 new terminal을 선택한다.https://lh7-us.googleusercontent.com/7v2DFtjvL-IGNBMXwYG4wCaOy3QcA2HEOlDUf_9upeDJ7T-SQtlH0NRbJl4mGVTWzz_ViHy7LqsVOiBIYaVwJgPRjs_YiTri3NJVlYKjhvTiPgU-fPzXPsqi9MngYVDPAiH3tzaMVfAr0U3UiuoGby0  [8-5]  https://lh7-us.googleusercontent.com/hlPSreoZ0zpo7DlFCLVwedyYqRqBWzZJJNZQmmN0s-RSW6KHrHNqnai_9VDcgjDaIxN5yK3iPt1jw31ojhVwAIu6bmH1reRTM-7omE7m5X5dFikiXe28M8iIUe-oy57EXksap4oH8YcaEzgbVyNHRmU  [8-6]  https://lh7-us.googleusercontent.com/tHLbcpergteLWWqecEBJOj9eJX54bMv_uMWpt7oITkXNzlSDvmSY7EtfUlpgT6quP9KSq8t4I0KRCKbRz-rKdqPyEvNac2ECWT5C3IjmsbBT3m6iraH_A-luDtZzdDtHl2OvO2j0MRsYaZEkZtYzrC8  [8-7]  https://lh7-us.googleusercontent.com/tIC5PiH73Mz1kLLL7URMk0yZWbLIIz3rjuhw6jf-D21TYc7Ku20xb63Ktd579IVh30YTGlRJkNR3RVcZBhT4bb91gOWkmDPFkf3crFDJefN6s0qMaSnCTnFdHbtNgzQTqyc3HENyQq9wakrlGXhYZ-E  [8-8]  https://lh7-us.googleusercontent.com/-hv8pgRSoxYUnSrbBKzEAUIXp7RQfgu3K-_0rNjBT-zLsdUgDA4Ws2241B1sRlSu6RxsMLP_tiSnLt1w2RLu_H1e2aW5GbNuBkjEhi3cCeNQolHIlltANK7ZLMXa4aXtiOM8-KuFI3KoAT7w4urksH0  [8-9]  https://lh7-us.googleusercontent.com/OdoCA33dPdyDtUDo3KQYCk9wUqbso-QMw2q2VqGtaW1dhjqPSWzZSFFk8VCVOk3IM0FW4t7FeCTJWdsd-1AVTBrCogBJ-pHXEGlHUMP68W-43okcGGHKN3Rt42VTV5jijsz3XEhYc8Fgynb7xeR9VIw  [8-10]  https://lh7-us.googleusercontent.com/2HJ5eHeXoJgfaSXlG0NWH5biS6VycTEAcLMedjbRuxZfYwBKfy3OJwhmSWkp_J4oDcvTB-HglvdBqscRUkGZdYawhA94-5ygN-l2uI1SZVsUlFRM5BZWUSZJm_pVGfGlM7JGtzedvTfEqdZ0h_SBMls  [8-11]  https://lh7-us.googleusercontent.com/fxr2NhZTM5-O-_tVx800h23PKU0nyQYxLpFO0kko1aesyHp7rMYaozldy9OQ_4nQ2r2mNYQOsAc1wAXYX9sVuQSCGityA6a0DoGTG3-JaGemcpHmIKbK1UoEjR-p3NalB8vKVDUX6drkxMr_8pTzKiM  이렇게 앱을 실행하고 있는 계획이다.  [8-12]  **6. 참고 자료**   1. Shannon Bradshaw, Kristina Chodorow, Eoin Brazil(2020),「MongoDB The Definitive Guide: Powerful ANd Scalable Data」,O’Reilly Media, Incorporated. 2. Vipul A M, Prathamesh Sonpatki(2016),「ReactJS by Example-Building Modern Web Applications with React」, Packt Publishing.   하나몬,「JavaScript, Node.js」, 2021.01.04. | | | | |

