**개발 완료 보고서**

**제출일 : 2024. 02. 06**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **팀명** | 코드야 싸우자 | | | | |
| **참여인원** | 박상호(팀장), 이수빈, 김다능 | | | | |
| **프로젝트 소개** | | | | | |
| **프로젝트 명** | 다나와 자동차 판매 사이트 만들기 | | | | |
| **활동 일시** | 02 / 01 ~ 02 / 06 | | | **장소** | 광주인력개발원  공학 1관 생산정보시스템 |
| **개발 환경** | C++ / Windows / Qt Creator Designer / DataBase | | | | |
| **개발 목적** | 그동안 학습했던 부분을 이용하여 자동차 판매 사이트를 구현 Qt의 ui출력되는 부분과 DB에 연결된 정보를 주고 받아 해당 사이트를 구현해 본다. | | | | |
| **업무분담** | **박상호** | | 1. DB수집  2. 메인화면 전체 화면출력구성  3. 브랜드화면 해당화면출력, 버튼기능 구현  4. 문서작성 | | |
| **이수빈** | | 1. DB수집  2. 회원가입, 로그인 기능 구현  3. 메인화면 버튼기능 구현  4. 모델별 검색화면 기간, 차종조회 기능  5. 문서작성 | | |
| **김다능** | | 1. DB수집  2. 전체 화면 및 기능구현  3. 문서작성 | | |
| **일정표** |  | | | | |
| **요구분석서** |  | | | | |
| **코드설명** | **박상호** |  | | | |
| **이수빈** | void LoginPage::gotoHome()  로그인페이지의 홈으로 이동한다.  void LoginPage::gotoLogin()  로그인창으로 이동한다. 기존의 로그인창에 입력되었던 텍스트는 지워진다.  void LoginPage::gotoJoin()  회원가입창으로 이동한다. 기존의 회원가입창에 입력되었던 텍스트는 지워진다.  휴대폰번호와 아이디의 중복검사값이 초기화되고 회원등록버튼이 비활성화 된다.  void LoginPage::gotoSID()  아이디찾기 화면으로 이동한다. 기존의 아이디찾기 창에 입력되었던 텍스트는 지워진다.  void LoginPage::gotoSPW()  비밀번호찾기 화면으로 이동한다. 기존의 비밀번호찾기 창에 입력되었던 텍스트는 지워진다.  void LoginPage::ProgressJoin()  회원등록이 진행된다. 데이터 공백을 검사하고 유저 테이블에 유저정보를 저장한다.  void LoginPage::CKPhoneNum()  기존 유저 테이블에서 신규 휴대폰번호의 중복을 검사한다.  void LoginPage::CKID()  기존 유저 테이블에서 신규 아이디의 중복을 검사한다.  void LoginPage::ProgressLogin()  입력된 아이디, 비밀번호를 유저 테이블과 대조해 일치 여부를 확인해 로그인이 진행되고 판매실적화면으로 넘어간다.  void LoginPage::SearchID()  입력된 이름과 휴대폰번호를 유저 테이블에서 검색해 해당하는 아이디를 반환한다.  void LoginPage::SearchPW()  입력된 이름, 아이디, 휴대폰번호를 유저 테이블에서 검색해 해당하는 비밀번호를 반환한다.  void LoginPage::pushButtonClicked()  광고 사이트를 새 브라우저로 연결한다.  connect(ui->pushButton\_6,SIGNAL(clicked()),this->parent(),SLOT(ShowSales()));  시그널 발생시 메인 윈도우의 슬롯함수에 연결한다. 판매실적화면으로 넘어간다.  l=new LoginPage(this);  ui->mainstack->addWidget(l);  ui 페이지(클래스) 연결 연결할 페이지는 모두 위젯 클래스여야 한다.  connect(ui->btnModel,SIGNAL(clicked()),m,SLOT(delCarType()));  연결된 페이지(클래스)로의 시그널-슬롯 연결  void MainWindow::ShowSales(void)  로그인 화면에서 판매실적 화면으로 넘어올때 초기화면을 세팅한다.  void MainWindow::btnStackBrandClicked()/void MainWindow::btnStackCarClicked()  브랜드별/차종별 탭 선택시 화면을 세팅한다.  void MainWindow::ChangeBStackPage()/void MainWindow::ChangeCStackPage()  브랜드별/차종별 탭의 페이지를 넘긴다.  void MainWindow::on\_btnWhole\_clicked()  void MainWindow::on\_btnBrand\_clicked()  void MainWindow::on\_btnModel\_clicked()  전체/브랜드/모델 탭 선택시 화면을 세팅한다.  void ModelTab::ShowDefault()  모델텝 초기화면(23년 12월 전체 모델 판매량 순위)을 출력한다.  void ModelTab::btnMonthClicked()/void ModelTab::btnDurationClicked()  월/기간 선택 버튼 클릭시 화면을 세팅한다.  void ModelTab::UpdateTable(QSqlQuery& qry)  쿼리문을 받아 판매량 순위 테이블 위젯의 내용을 갱신한다.  void ModelTab::mMonthCurrentIndexChanged(int index)  월 선택 조회시 변경된 월의 판매량 순위 데이터를 조회하는 쿼리문을 생성한다.  void ModelTab::btnShowClicked()  기간 선택 조회시 조회버튼을 누른 시점의 기간 창에 설정된 시작달과 종료달을 읽어와 기간내 판매량 순위 데이터를 조회하는 쿼리문을 생성한다.  void ModelTab::setCarType1()  차종번호를 저장하는 변수 값을 세팅하고 해당하는 내용의 판매량 순위를 출력한다.  void ModelTab::delCarType()  차종번호를 저장하는 변수 값을 초기화한다. | | | |
| **김다능** | QString WholeTab::ConvertBrandName(int bnum)  브랜드이름 변환  void WholeTab::on\_Select\_Ask\_PushButton\_clicked()  push button 을 클릭하면 이벤트 발생  QsqlQuery QueryLoadData(DB\_Connection)  DB와 연결하여 sqlite로 데이터 가져오기  QString month = QString::number(ui->Select\_Month\_ComboBox->currentIndex()+1)  콤보박스의 인덱스 값을 가져오는 함수  QString exMonth = QString::number(ui->Select\_Month\_ComboBox->currentIndex());  콤보박스의 이전 컬럼값을 가져오는 함수  SELECT BRAND AS company,  BRAND의 컬럼값을 출력함, 이하 company라고 한다  sum(\"2023\_"+month+"\") AS sale ,  2023년 콤보박스의 인덱스값을 대입해 해당 컬럼값을 가져온다, 이하 sale라고 한다  ROUND((total(\"2023\_"+month+"\")/(SELECT SUM(\"2023\_"+month+"\") FROM CAR\_TB WHERE BRAND < 7))\*100, 0) as percent  출력되는 목록중 해당 목록의 백분위를 출력한다, 이하 percent라고 한다.  FROM CAR\_TB WHERE BRAND BETWEEN 1 AND 6  car\_tb 테이블의 BRAND 컬럼의 1부터 6까지의 데이터에서 출력한다.  GROUP BY BRAND  컬럼명 기준으로 같은 값의 데이터를 그룹화해서 출력  ORDER BY SUM(\"2023\_"+month+"\") DESC LIMIT 5;");  해당데이터값을 내림차순(desc)으로 정렬(order by)한다. 최대 출력 행은 5개이다.  ui->Brand\_Car\_Korea->setRowCount(NumberOFRowsToDisplay)  Brand\_Car\_Korea의 출력되는 행의 개수  DB\_Connection.open()  DB연결, 열기  SQlDatabase::database().transaction()  데이터베이스에 접근하여 작업하기 위한 함수  QueryLoadData.exec()  준비된 프로그램 실행  NumberOFRowsToDisplay() = @  출력되는 행의 개수 = @ | | | |
| **후기** | **박상호** |  | | | |
| **이수빈** | 파이썬으로 Qt를 활용하여 진행한 프로젝트에서는 클래스에 대한 이해가 충분히 되지 않아 잘 모르고 썼던 부분이 있었습니다. 하지만 C++를 제대로 공부하며 클래스에 대한 이해가 더해지니 Qt에서 각 ui별 클래스를 분리할 때 그 연결을 어떻게 다룰 것인지 고민해보고 적용하여 눈으로 확인 할 수 있었던 좋은 기회였습니다. 덕분에 팀원들과 역할 분담도 수월하게 진행되었다고 생각합니다. 기능은 막힘없이 구현된 부분도 있지만 생각했던 대로 구현되지 않은 기능들도 있었습니다. 기한 내에 마치기 위해 해당 기능은 조금 더 단순화하여 코드를 작성하기는 했지만 프로젝트를 마친 후 개인적으로 학습해 보충할 것입니다. 팀원들 모두 수고하셨습니다. | | | |
| **김다능** | Qt, DB, C++ 3가지를 이용하여 진행하는 프로젝트입니다.  이전 프로젝트에서도 느낀 바 이지만  계획서를 작성하고 계획서 대로 프로젝트를 이행하는 것, 각각 맡은 파트를 최대한 완성하는것, 완성을 위해선 각각의 파트를 합쳐서 하나의 프로젝트를 만들어야 하기 때문에 여러 세세하고 생각지 못한 부분들을 서로의 소통을 통하여 약속을 정하고 문제가 생길시 같이 해결해가는 것이 중요하다고 느꼈습니다.  개인프로젝트 때는 느끼지 못하였지만 이러한 경험을 통해 팀프로젝트의 핵심은 팀원간 소통이라고 생각합니다. | | | |