**Capstone Design Core Source Code**

Group name : 인슐린 (insulin)

Member : 최자은, 김경서, 정혜선, 김초희, 오유석

|  |
| --- |
| **DBManager.java** |
| public class DBManager extends SQLiteOpenHelper {  final String TAG = DBManager.class.getName();  public Context context;  public DBManager(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {  super(context, name, factory, version);  this.context = context;  }  /\*\*  \* Database가 존재하지 않을 때, 딱 한번 실행된다.  \* DB를 만드는 역할을 한다.  \*  \* @param db  \*/  @Override  public void onCreate(SQLiteDatabase db) {  //String 보다 StringBuffer가 Query 만들기 편하다.  StringBuffer sb = new StringBuffer();  sb.append(" CREATE TABLE SUGARBLOOD ( ");  sb.append(" datedata TEXT, ");  sb.append(" timedata TEXT, ");  sb.append(" SUGARB TEXT) ");  StringBuffer sb1 = new StringBuffer();  sb1.append(" CREATE TABLE EATLIST ( ");  sb1.append(" eatdatedata TEXT, ");  sb1.append(" eattimedata TEXT, ");  sb1.append(" eatlist TEXT) ");  // SQLite Database로 쿼리 실행  db.execSQL(sb.toString());  db.execSQL(sb1.toString());  Toast.makeText(context, "Table 생성 완료", Toast.LENGTH\_SHORT).show();  }  /\*\*  \* Application의 버전이 올라가서  \* Table 구조가 변경되었을 때 실행된다.  \*  \* @param db  \* @param oldVersion  \* @param newVersion  \*/  @Override  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  Toast.makeText(context, "버전이 올라갔습니다.", Toast.LENGTH\_SHORT).show();  }  public void insert(String query) {  SQLiteDatabase db = getWritableDatabase();  db.execSQL(query);  Log.d(TAG, "execute insert query");  db.close();  }  public void delete(String query) {  SQLiteDatabase db = getWritableDatabase();  db.execSQL(query);  Log.d(TAG, "execute delete query");  db.close();  }  public String select(String query) {  SQLiteDatabase db = getReadableDatabase();  Cursor cursor = db.rawQuery(query, null);  Log.d(TAG, "execute selete query");  String str = "";  //컬럼사이사이 ","를 삽입하여 리턴  for (int j = 0; j < cursor.getCount(); j++) {  if (j == 0) {  cursor.moveToFirst();  } else {  cursor.moveToNext();  }  for (int i = 0; i < cursor.getColumnCount(); i++) {  if (i == 0) {  str += cursor.getString(i) + ",";  } else if (i == cursor.getColumnCount() - 1) {  str += cursor.getString(i)+"--r--n";  } else {  str += cursor.getString(i) + ",";  }  }  }  Log.d(TAG, "select: " + str);  return str;  }  public void testDB() {  SQLiteDatabase db = getReadableDatabase();  }  } |
| **FirebaseMessagingService.java** |
| public class FirebaseMessagingService extends com.google.firebase.messaging.FirebaseMessagingService {  private static final String TAG = "FirebaseMsgService";  //TODO : 문자열 나누기 이상  private final String headsplitIndex="-h-r-n--";  // [START receive\_message]  @Override  public void onMessageReceived(RemoteMessage remoteMessage) {  String getMessage = remoteMessage.getData().get("message");  try {  String[] splitMsg = getMessage.split(headsplitIndex);  sendNotification(splitMsg[0], splitMsg[1]);  }catch (Exception e){  e.printStackTrace();  }  }  private void sendNotification(String messageBody, String realData) {  //알람을 선택시 어떤 액티비티로 바로 갈지 정함  Intent intent = new Intent(this, userfoodActivity.class);  intent.putExtra("foodData",realData);  intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);  PendingIntent pendingIntent = PendingIntent.getActivity(this, 0 /\* Request code \*/, intent, PendingIntent.FLAG\_ONE\_SHOT);  Uri defaultSoundUri= RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);  NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this)  .setSmallIcon(R.mipmap.ic\_launcher)  .setContentTitle("Diabetes Management")  .setContentText(messageBody)  .setAutoCancel(true)  .setSound(defaultSoundUri)  .setContentIntent(pendingIntent);  NotificationManager notificationManager =  (NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE);  notificationManager.notify(0 /\* ID of notification \*/, notificationBuilder.build());  }  } |
| **ImageFragment.java** |
| public class ImageFragment extends Fragment {  public static ImageFragment newInstance(int imageUrl) { //static으로 올려 메모리 리소스 절약 성능향상  Bundle args = new Bundle();  args.putInt("imageUrl", imageUrl);  ImageFragment fragment = new ImageFragment();  fragment.setArguments(args);  return fragment;  }  public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {  View view = inflater.inflate(R.layout.imagefragment, container, false);  ImageView imageView = (ImageView) view.findViewById(R.id.imagefragment\_imageview);  // imageView.setImageResource(getArguments().getInt("imageUrl"));  Glide.with(this).load(getArguments().getInt("imageUrl")).into(imageView);  return view;  }  } |
| **ListSugarblood.java** |
| public class ListSugarblood extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_list\_sugarblood);  ListView listview;  ListViewAdapter adapter;  //adapter 생성  adapter = new ListViewAdapter();  // 리스트뷰 참조 및 Adapter달기  listview = (ListView) findViewById(R.id.list\_sugarblood);  listview.setAdapter(adapter);  DBManager dbManager = new DBManager(getApplicationContext(), "SUGAR.db", null, 1);  String tmp = dbManager.select("SELECT \* FROM SUGARBLOOD");  //리턴된 테이블 행끼리 나눔  String[] arrRow = tmp.split("--r--n");  for(int i = 0; i<arrRow.length;i++) {  String[] arrCol = arrRow[i].split(",");  adapter.addItem(arrCol[0],arrCol[1],arrCol[2]);  }  // 위에서 생성한 listview에 클릭 이벤트 핸들러 정의.  listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {  @Override  public void onItemClick(AdapterView parent, View v, int position, long id) {  // get item  ListViewItem item = (ListViewItem) parent.getItemAtPosition(position) ;  String dayStr = item.getDayStr() ;  String timeStr = item.getTimeStr() ;  String sBloodStr = item.getsBloodStr() ;  }  }) ;  }  } |
| **MenuregiActivity.java** |
| public class MenuregiActivity extends AppCompatActivity {  private static final String TAG = MenuregiActivity.class.getName();  final int REQ\_CODE\_SELECT\_IMAGE = 1000;  final int REQ\_CODE\_CAMERA\_CALL = 1001;  final String serverAddr = "http://albaneo.ddns.net:9393/fcm/upload.php";  Button btn\_camera, btn\_gallery, btn\_menu\_list, btn\_upload;  ImageView iv\_picture;  RadioGroup radio\_mealtime;  JSONObject jsonObject;  private String cropImgUri;  private boolean Permit;  private Uri photoUri, outputUri = null;  @Override  protected void onCreate(@Nullable Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_menuregi);  String[] permissionList = {Manifest.permission.CAMERA, Manifest.permission.WRITE\_EXTERNAL\_STORAGE};  PermissionCheck(permissionList);  radio\_mealtime = (RadioGroup) findViewById(R.id.mealtimeradio);  btn\_upload = (Button) findViewById(R.id.btn\_upload);  btn\_gallery = (Button) findViewById(R.id.btn\_gallery);  btn\_camera = (Button) findViewById(R.id.btn\_camera);  btn\_menu\_list = (Button) findViewById(R.id.btn\_menu\_list);  iv\_picture = (ImageView) findViewById(R.id.iv\_picture);  btn\_menu\_list.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent(MenuregiActivity.this, userfoodActivity.class);  startActivity(intent);  }  });  btn\_camera.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (Permit) {  cameraCall();  } else {  Toast.makeText(MenuregiActivity.this, "권한을 거부하셨습니다. 권한을 설정해 주세요", Toast.LENGTH\_SHORT).show();  }  }  });  btn\_gallery.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if (Permit) {  galleryCall();  } else {  Toast.makeText(MenuregiActivity.this, "권한을 거부하셨습니다. 권한을 설정해 주세요", Toast.LENGTH\_SHORT).show();  }  }  });  btn\_upload.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (iv\_picture.getDrawable() != null) {  new Thread(new Runnable() {  @Override  public void run() {  ImageUploader(cropImgUri);  iv\_picture.setImageURI(null);  }  }).start();  } else {  Toast.makeText(MenuregiActivity.this, "이미지를 선택해 주세요", Toast.LENGTH\_SHORT).show();  }  }  });  }  @Override  protected void onResume() {  super.onResume();  setRadioBtn();  }  @Override  protected void onStop() {  super.onStop();  iv\_picture.setImageURI(null);  }  private void setRadioBtn() {  SimpleDateFormat sdf = new SimpleDateFormat("HH");  String time = sdf.format(new Date(System.currentTimeMillis()));  int now = Integer.parseInt(time);  if (now > 17 && now < 21) {  //저녁시간  radio\_mealtime.check(R.id.mealoption3);  } else if (now > 11 && now < 14) {  //점심시간  radio\_mealtime.check(R.id.mealoption2);  } else {  //아침시간  radio\_mealtime.check(R.id.mealoption1);  }  }  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  if (resultCode != RESULT\_OK) {  return;  } else {  switch (requestCode) {  case REQ\_CODE\_CAMERA\_CALL:  photoUri = data.getData();  cropImgUri = getImagePathToUri(photoUri);  Glide.with(this).load(photoUri).into(iv\_picture);  break;  case REQ\_CODE\_SELECT\_IMAGE:  photoUri = data.getData();  cropImgUri = getImagePathToUri(photoUri);  Glide.with(this).load(photoUri).into(iv\_picture);  break;  }  }  }  private void cameraCall() {  if (iv\_picture.getDrawable() != null) {  iv\_picture.setImageBitmap(null);  }  Intent intent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);  intent.putExtra(Intent.EXTRA\_ALLOW\_MULTIPLE, true);  startActivityForResult(intent, REQ\_CODE\_CAMERA\_CALL);  }  private void galleryCall() {  if (iv\_picture.getDrawable() != null) {  iv\_picture.setImageBitmap(null);  }  Intent intent = new Intent(Intent.ACTION\_PICK);  intent.setType(MediaStore.Images.Media.CONTENT\_TYPE);  intent.setData(MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);  intent.putExtra(Intent.EXTRA\_ALLOW\_MULTIPLE, true);  startActivityForResult(intent, REQ\_CODE\_SELECT\_IMAGE);  }  public Uri getImageUri(Context inContext, Bitmap inImage) {  ByteArrayOutputStream bytes = new ByteArrayOutputStream();  inImage.compress(Bitmap.CompressFormat.JPEG, 100, bytes);  String path = MediaStore.Images.Media.insertImage(inContext.getContentResolver(), inImage, "Title", null);  return Uri.parse(path);  }  private String getImagePathToUri(Uri data) {  String[] proj = {MediaStore.Images.Media.DATA};  Cursor cursor = managedQuery(data, proj, null, null, null);  int column\_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);  cursor.moveToFirst();  String imgPath = cursor.getString(column\_index);  return imgPath;  }  private String getImageNameToUri(Uri data) {  String[] proj = {MediaStore.Images.Media.DATA};  Cursor cursor = managedQuery(data, proj, null, null, null);  int column\_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);  cursor.moveToFirst();  String imgPath = cursor.getString(column\_index);  String imgName = imgPath.substring(imgPath.lastIndexOf("/") + 1);  return imgName;  }  public void PermissionCheck(final String[] permission) {  PermissionListener permissionlistener = new PermissionListener() {  @Override  public void onPermissionGranted() {  //Permission Granted  Log.i(TAG, permission + "Permission Granted");  Permit = true;  }  @Override  public void onPermissionDenied(ArrayList<String> deniedPermissions) {  //Permission Denied  Log.i(TAG, "Permission Denied");  Toast.makeText(MenuregiActivity.this, "권한이 거부되었습니다. 앱 사용시 기능을 사용할 수 없습니다.", Toast.LENGTH\_SHORT).show();  }  };  TedPermission.with(this)  .setPermissionListener(permissionlistener)  .setDeniedMessage("권한을 거부하시면 이 서비스를 사용할 수 없습니다.\n\n[설정] > [권한] 항목에서 권한을 설정해주세요.")  .setPermissions(permission)  .setGotoSettingButton(true)  .setGotoSettingButtonText("설정")  .check();  }  private void ImageUploader(String imgUri) {  SharedPreferences sharedPreferences = getSharedPreferences("userToken", MODE\_PRIVATE);  String userToken = sharedPreferences.getString("deviceToken", "");  SharedPreferences sharedPreferenceInfo = getSharedPreferences("userInfo", MODE\_PRIVATE);  String userMail = sharedPreferenceInfo.getString("userMail", "");  String[] user = userMail.split("@");  try {  OkHttpClient client = new OkHttpClient();  RequestBody requestBody = new MultipartBody.Builder()  .setType(MultipartBody.FORM)  //addformdatapart(서버에서 인식할 이름(키워드), 실제 저장될 이름, 속성)  .addFormDataPart("uploaded\_file", user[0] + ".jpg", RequestBody.create(MultipartBody.FORM, new File(imgUri)))  .build();  RequestBody body = new FormBody.Builder()  .add("upusermail", user[0])  .build();  Request request = new Request.Builder()  .url(serverAddr)  .post(requestBody)  .build();  Request request1 = new Request.Builder()  .url("http://albaneo.ddns.net:9393/fcm/index.php")  .post(body)  .build();  Response response = client.newCall(request1).execute();  Log.d(TAG,"index.php response"+response.body());  client.newCall(request).enqueue(new Callback() {  @Override  public void onFailure(Call call, IOException e) {  Log.d(TAG, "onFailure: 업로드 실패");  }  @Override  public void onResponse(Call call, Response response) throws IOException {  Log.d(TAG, "onResponse: Upload Success");    iv\_picture.setImageURI(null);  Intent intent = new Intent(MenuregiActivity.this, AppMainActivity.class);  startActivity(intent);  }  });  } catch (Exception e){  e.printStackTrace();  }  }  } |
| **SignInActivity.java** |
| public class SignInActivity extends AppCompatActivity implements GoogleApiClient.OnConnectionFailedListener {  static final String TAG = SignInActivity.class.getName();  static final int RC\_GOOGLE\_SIGN\_IN = 9001;  FirebaseAuth mFirebaseAuth;  FirebaseAuth.AuthStateListener mFirebaseAuthListener;  SignInButton mSigninGoogleButton;  GoogleApiClient mGoogleApiClient;  //다이얼로그 선언  ProgressDialog dialog;  private String userToken;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_signin);  Log.i(TAG, "Start");  dialog = new ProgressDialog(SignInActivity.this);  dialog.setProgressStyle(ProgressDialog.STYLE\_SPINNER);  dialog.setMessage("잠시만 기다려 주세요");  /\*  \* Firebase  \*/  mFirebaseAuth = FirebaseAuth.getInstance();  mFirebaseAuthListener = new FirebaseAuth.AuthStateListener() {  @Override  public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  FirebaseUser user = firebaseAuth.getCurrentUser();  if (user != null) {  Log.d(TAG, "sign in");  Intent intent = new Intent(SignInActivity.this, StartActivity.class);  dialog.dismiss();  startActivity(intent);  finish();  } else {  Log.d(TAG, "sign out");  }  }  };  /\*  \* Google Login  \*/  mSigninGoogleButton = (SignInButton) findViewById(R.id.sign\_in\_google\_button);  mSigninGoogleButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  dialog.show();  Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);  startActivityForResult(signInIntent, RC\_GOOGLE\_SIGN\_IN);  }  });  GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)  .requestIdToken(getString(R.string.default\_web\_client\_id))  .requestEmail()  .build();  mGoogleApiClient = new GoogleApiClient.Builder(this)  .enableAutoManage(this, this)  .addApi(Auth.GOOGLE\_SIGN\_IN\_API, gso)  .build();  }  @Override  protected void onStart() {  super.onStart();  mFirebaseAuth.addAuthStateListener(mFirebaseAuthListener);  }  @Override  public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {  Log.d(TAG, "onConnectionFailed:" + connectionResult);  }  @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  super.onActivityResult(requestCode, resultCode, data);  if (requestCode == RC\_GOOGLE\_SIGN\_IN) {  GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);  if (result.isSuccess()) {  String token = result.getSignInAccount().getIdToken();  AuthCredential credential = GoogleAuthProvider.getCredential(token, null);  mFirebaseAuth.signInWithCredential(credential);  } else {  Log.d(TAG, "Google Login Failed." + result.getStatus());  dialog.dismiss();  Toast.makeText(this, "로그인 실패 네트워크 확인후 다시 시도해주세요", Toast.LENGTH\_SHORT).show();  }  }  }  } |
| **sugarBloodActivity.java** |
| public class sugarBloodActivity extends AppCompatActivity {  private Button btn\_save;  private Spinner timeSpinner;  private EditText sugarEditText;  private String spinnerValue;  @Override  protected void onCreate(@Nullable Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.sugarblood);  timeSpinner = (Spinner) findViewById(R.id.time);  sugarEditText = (EditText) findViewById(R.id.et\_value);  btn\_save = (Button) findViewById(R.id.btn\_saveValue);  String[] spinnerOption = getResources().getStringArray(R.array.time);  ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, R.layout.support\_simple\_spinner\_dropdown\_item, spinnerOption);  timeSpinner.setAdapter(adapter);  setSpinnerValue();  btn\_save.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  btn\_saveValue();  sugarEditText.setText(null);  }  });  }  private void setSpinnerValue() {  SimpleDateFormat sdf = new SimpleDateFormat("HH");  String currentTime = sdf.format(new Date(System.currentTimeMillis()));  int current = Integer.parseInt(currentTime);  if (current > 4 && current < 6) {  timeSpinner.setSelection(0);  } else if (current >= 7 && current < 9) {  timeSpinner.setSelection(1);  } else if (current >= 9 && current < 10) {  timeSpinner.setSelection(2);  } else if (current >= 10 && current < 12) {  timeSpinner.setSelection(3);  } else if (current >= 13 && current < 15) {  timeSpinner.setSelection(4);  } else if (current >= 16 && current < 18) {  timeSpinner.setSelection(5);  } else if (current >= 19 && current < 20) {  timeSpinner.setSelection(6);  } else if (current >= 21 && current < 22) {  timeSpinner.setSelection(7);  } else {  timeSpinner.setSelection(0);  }  }  public void btn\_saveValue() {  SimpleDateFormat sdf = new SimpleDateFormat("yyyyMMdd");  String today = sdf.format(new Date(System.currentTimeMillis()));  String time = timeSpinner.getSelectedItem().toString();  String sugarblood = sugarEditText.getText().toString();  String insertQuery = "INSERT INTO SUGARBLOOD VALUES (\"" + today + "\" ,\"" + time + "\" ,\"" + sugarblood + "\")";  final DBManager dbManager = new DBManager(getApplicationContext(), "SUGAR.db", null, 1);  dbManager.insert(insertQuery);  Toast.makeText(sugarBloodActivity.this, "저장 되었습니다.", Toast.LENGTH\_SHORT).show();  finish();  }  } |
| **GoogleImageCrawler.py** |
| from icrawler.builtin import GoogleImageCrawler  import csv  def google\_crawler(CSV,MAX\_NUM=1000):  f = open(CSV, 'r')  rdr = csv.reader(f)  query\_list = []  for list in rdr:  query\_list.extend(list)  for i in range(len(query\_list)):  query = query\_list[i]  query = str(query)  google\_crawler = GoogleImageCrawler(parser\_threads=2,downloader\_threads=4,  storage = {'root\_dir':query})  google\_crawler.crawl(keyword = query,max\_num=MAX\_NUM,date\_min=None,date\_max=None,  min\_size=(200,200),max\_size=None)  f.close()  if \_\_name\_\_ == "\_\_main\_\_":  google\_crawler("C:/Users/User/Desktop/CapstoneProject/foodlist.csv",300) |
| **classify.py** |
| # -\*- coding: utf-8 -\*-  """Inception v3 architecture 모델을 retraining한 모델을 이용해서 이미지에 대한 추론(inference)을 진행하는 예제"""  import numpy as np  import tensorflow as tf  imagePath = 'C:/Users/User/Desktop/CapstoneProject/uploads/menu.jpg' # 추론을 진행할 이미지 경로  modelFullPath = 'C:/Users/User/Desktop/CapstoneProject/outputs/output\_graph.pb' # 읽어들일 graph 파일 경로  labelsFullPath = 'C:/Users/User/Desktop/CapstoneProject/outputs/output\_labels.txt' # 읽어들일 labels 파일 경로  def create\_graph():  """저장된(saved) GraphDef 파일로부터 graph를 생성하고 saver를 반환한다."""  # 저장된(saved) graph\_def.pb로부터 graph를 생성한다.  with tf.gfile.FastGFile(modelFullPath, 'rb') as f:  graph\_def = tf.GraphDef()  graph\_def.ParseFromString(f.read())  \_ = tf.import\_graph\_def(graph\_def, name='')  def run\_inference\_on\_image():  answer = None  if not tf.gfile.Exists(imagePath):  tf.logging.fatal('File does not exist %s', imagePath)  return answer  image\_data = tf.gfile.FastGFile(imagePath, 'rb').read()  # 저장된(saved) GraphDef 파일로부터 graph를 생성한다.  create\_graph()  with tf.Session() as sess:  softmax\_tensor = sess.graph.get\_tensor\_by\_name('final\_result:0')  predictions = sess.run(softmax\_tensor,  {'DecodeJpeg/contents:0': image\_data})  predictions = np.squeeze(predictions)  top\_k = predictions.argsort()[-5:][::-1] # 가장 높은 확률을 가진 5개(top 5)의 예측값(predictions)을 얻는다.  f = open(labelsFullPath, 'rb')  lines = f.readlines()  labels = [str(w).replace("\n", "") for w in lines]  for node\_id in top\_k:  human\_string = labels[node\_id]  score = predictions[node\_id]  print('%s (score = %.5f)' % (human\_string, score))  answer = labels[top\_k[0]]  return answer  if \_\_name\_\_ == '\_\_main\_\_':  run\_inference\_on\_image() |
| **index.php** |
| <!doctype html>  <head>  <meta charset="UTF-8">  <title>FCM Push Notification</title>  </head>  <body>  <?php  $splitIndex = "-r-n--";  $usermail = $\_POST['userMail'];    require("config.php");  $conn = mysqli\_connect(DB\_HOST, DB\_USER, DB\_PSW, DB\_NAME);  $sql = "Select Token From users WHERE User\_Mail = '$usermail'";  $queryresult = mysqli\_query($conn,$sql);  $python = "C:/Users/User/AppData/Local/Programs/Python/Python36/python.exe";  $pythonScript = "C:/Bitnami/wampstack-7.1.11-0/apache2/htdocs/fcm/test.py".' '.$usermail;  //$cmd에 인식결과 저장  $cmd = shell\_exec($python." ".$pythonScript);  if($cmd){  // 출력 결과 다듬기  $result = str\_replace("b'","",$cmd);  $result = str\_replace("' ","",$result);  $result = str\_replace(' ',"",$result);  $result = str\_replace("(score=","",$result);  $result = str\_replace(")","",$result);  $result = str\_replace("\n",$splitIndex,$result);  ?>  <form action="push\_notification.php" method="post" id="foodData">  <input type="hidden" name="userMail" value="<?php echo $usermail;?>">  <input type = "hidden" name="message" value="<?php echo $cmd;?>" >  </form>  <script type="text/javascript">  this.document.getElementById("foodData").submit();  </script>  <?php  }else{  echo "Error Occured";  }  ?>  </body>  </html> |
| **upload.php** |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8"/>  <title>FCM Push Notification</title>  </head>  <body>  <?php  $usermail = $\_POST['upusermail'];  //업로드한 이미지의 경로  $root\_dir = "uploads/";  $save\_path = $root\_dir . basename( $\_FILES['uploaded\_file']['name']);  move\_uploaded\_file($\_FILES['uploaded\_file']['tmp\_name'],$save\_path);  ?>  <form action="index.php" method="post" id="upload">  <input type="hidden" name="userMail" value="<?php echo $usermail;?>">  </form>  <script type="text/javascript">  this.document.getElementById("upload").submit();  </script>  </body>  </html> |
| **push\_notification.php** |
| <?php  $headsplitIndex = "-h-r-n--";  $userMail = $\_POST['userMail'];  function send\_notification ($tokens, $message)  {  $url = 'https://fcm.googleapis.com/fcm/send';  $fields = array(  'registration\_ids' => $tokens,  'data' => $message  );  $headers = array(  'Authorization:key =' . GOOGLE\_API\_KEY,  'Content-Type: application/json'  );  $ch = curl\_init();  curl\_setopt($ch, CURLOPT\_URL, $url);  curl\_setopt($ch, CURLOPT\_POST, true);  curl\_setopt($ch, CURLOPT\_HTTPHEADER, $headers);  curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);  curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, 0);  curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);  curl\_setopt($ch, CURLOPT\_POSTFIELDS, json\_encode($fields));  $result = curl\_exec($ch);  if ($result === FALSE) {  die('Curl failed: ' . curl\_error($ch));  }  curl\_close($ch);  return $result;  }  //데이터베이스에 접속해서 토큰들을 가져와서 FCM에 발신요청  include\_once 'config.php';  $conn = mysqli\_connect(DB\_HOST, DB\_USER, DB\_PSW, DB\_NAME);  //아이디를 키로 토큰값 가져온다  $sql = "Select Token From users WHERE User\_Mail = '$userMail'";  $result = mysqli\_query($conn,$sql) or die($conn);  $tokens = array();  if(mysqli\_num\_rows($result) > 0 ){  while ($row = mysqli\_fetch\_assoc($result)) {  $tokens[] = $row["Token"];  }  }  mysqli\_close($conn);  //폼에서 입력한 메세지를 받음  $myMessage = $\_POST['message'];  $myMessage = "사진 인식 완료. 알림을 터치해 주세요".$headsplitIndex.$myMessage;  if ($myMessage == ""){  $myMessage = "새글이 등록되었습니다.";  }  $message = array("message" => $myMessage);  $message\_status = send\_notification($tokens, $message);  ?> |
| **register.php** |
| <?php  if(isset($\_POST["Token"]) ){  $token = $\_POST["Token"];  if(isset($\_POST["userMail"])){  $tmpmail = $\_POST["userMail"];  $mail = explode('@',$tmpmail);  }else {  $mail = "";  }  require 'config.php';  $query = "INSERT INTO users VALUES ('$mail[0]','$token') ON DUPLICATE KEY UPDATE Token = '$token', User\_Mail = '$mail[0]'";  $conn = mysqli\_connect(DB\_HOST, DB\_USER, DB\_PSW, DB\_NAME);  if($conn){  mysqli\_query($conn, $query);  mysqli\_close($conn);    }  }else {  echo "failed";  }  ?> |