|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **□ 수행평가 - [디지털 핵심 실무인재양성] 프로젝트형 IOT 서비스 개발** | | | | | | |
|  |  |  | |  | |  |
| **과정명** | | **프로젝트형 IOT 서비스 개발** | | | | |
| **교과목명** | | 인터페이스 개발 | | **훈련교사** | | 이진만 |
| **단원구성** | | - 기반 언어 프로그래밍  - 데이터베이스 활용  - 웹 프로그래밍  - 웹 애플리케이션 구현 프로젝트  - 사용자 관리 및 보안 프레임워크 구축 프로젝트 | | | | |
| **수행날짜** | | 2021.02.01 | 훈련생명 | |  | |
| **과제개요 I** | | | | | | |
| 1. Python을 이용한 응용 Appication 개발 2. Lotto 게임을 위한 프로그램 기획 및 설계 3. 요구사항   - 당청금은 랜덤하게 생성한다.  - 당첨번호는 렌덤하게 중복이 되지 않도록 생성한다.  - 순위에 따라 당첨금을 차등 지급한다.  - 게임을 끝내고 다시 시작할 수 있다.   1. Python 파일과 실행 결과를 제출 | | | | | | |
| 작성 내용   1. Lotto 게임을 위한 프로그램 기획 및 설계   # 1. 시나리오  # 사용자로부터 6가지 숫자를 입력받는다.  # 당첨금 누적시스템  # 3등까지 당첨금 지급  # 단, 1등 상금은 4개 이상 맞춘 사람만 받을 수 있음  # 금액 범위 (100~10,000)  # 본인 포함 100명의 참가자  # 2. 전제 조건 :  # - 당첨금은 랜덤하게 만든다.  # - 당첨 번호는 랜덤하게 만든다.  # - 순위에 따라 당첨금을 차등 지급한다.  # - 게임을 끝내고 다시 시작할 수 있다.  2. Python 코드  import random  #Lotto Game  def print\_waring():  """ 조건에 맞지 않는 Input이 들어왔을 경우 """  print("1~45의 숫자 6개를 중복되지 않게 다시 입력해주세요")  def gen\_lotto():  """ 1~45의 중복되지 않는 6개의 숫자 list를 return """  visited = [0]\*46  ret = []  while len(ret)<6:  next = random.randint(1,45)  if visited[next]==0:  visited[next] = 1  ret.append(next)  return ret  def get\_prize(i, n):  """  상금을 주어진 조건에 맞춰서 분배  1등상의 경우 4개 이상 맞춘 사람만 받을 수 있음  2등상의 경우 2개 이상 맞춘 사람만 받을 수 있음  """  ret = 0  for idx in range(len(acc\_price)):  if i<4 and idx==0:  continue  if i<2 and idx<2:  continue  if acc\_price[idx]==0:  continue  ret = acc\_price[idx]  acc\_price[idx] = 0  break  return int(ret/n)  #Lotto Game이 진행된 회차  week = 1  # 누적된 상금을 저장하는 List  acc\_price = [0] \* 3 # 3등까지 당첨금  # User를 포함한 100명의 Lotto번호를 저장하는 List  guest = [[] for i in range(100)]  while True:  # 금번 발생한 상금을 내림차순으로 저장하는 List  price = []  for i in range(3):  price.append(random.randint(100, 10000))  price.sort(reverse=True)  # 상금을 축적  for i in range(3):  acc\_price[i] += price[i]  print("\n\n\n1조의 ",week,"번째 Lotto Game을 찾아주셔서 감사합니다. 현재 누적 상금은",sep='')  print("1등 : %d만원\n2등 : %d만원\n3등 : %d만원입니다\n\n\n"%(acc\_price[0],acc\_price[1],acc\_price[2]))  # Key : Lotto Game 참여자의 맞춘 번호의 수  # Value : 참여자의 번호(User의 경우 99번)  score\_board = {6:[],5:[],4:[],3:[],2:[],1:[],0:[]}  # Lotto 번호를 발생  lotto = gen\_lotto()  for i in range(99):  # 참여자의 Lotto번호를 갱신  guest[i].clear()  guest[i] = gen\_lotto()  # 사용자의 숫자 입력  while True:  user\_num = input("중복 안 되는 숫자 입력[6개] : ")  if user\_num.find(' ')==-1:  print\_waring()  else:  user\_num = user\_num.split(' ')  while user\_num.count('')>0:  user\_num.remove('')  print()  if len(user\_num) != 6:  print\_waring()  continue  guest[99].clear()  for i in range(len(user\_num)):  if user\_num[i].isdecimal() and int(user\_num[i])>0 and int(user\_num[i])<46:  guest[99].append(int(user\_num[i]))  else:  print\_waring()  break  if len(guest[99]) ==6:  break  #--------------debug-----------------#  # for idx, g in enumerate(guest):  # print(idx, " : ",g)  #score\_board 갱신  for idx in range(100):  cnt = 0  for n in guest[idx]:  if n in lotto:  cnt += 1  score\_board[cnt].append(idx+1)  #상금 분배  for i in range(6,0,-1):  if len(score\_board[i]) != 0:  money = get\_prize(i, len(score\_board[i]))  if money==0:  break  print("축하드립니다!!!")  print(score\_board[i])  print("상금 ",money,"만원을 획득하셨습니다!", sep='')  print("다음 주에 다시 찾아와주세요 >\_<")  3. Python 실행결과    첫 번째 입력처럼 1~45이외의 숫자가 들어오거나 중복된 숫자가 들어올 경우 다시 입력을 받는다.  100명의 참가자 중에 상금을 획득한 사람의 번호와 금액을 함께 출력한다. | | | | | | |

|  |
| --- |
| **과제개요 II** |
| Python, MariaDB, Django를 이용한 Web Application 개발  1. UserTb, ItemTb를 이용한 CRUD 프로그램 개발  2. HTML5 기반 Web Client Application 구현  3. Django 환경 셋팅 및 웹 서버 환경 셋팅  4. Python을 이용한 Web Server Application 구현  5. Django에서의 Session 구현  6. 지도 API를 이용한 지도 구현  7. Logger를 이용한 Log 파일 생성 |
| 작성 내용   1. 시스템구성도   Django를 활용하여 Web을 구축한다.  SQL문을 활용하여 DB의 usertb, itemtb를 조작한다.   1. 사용 스택   Python 3.6, Django 3.15   1. 데이터베이스 구축 SQL문   class Sql:  userlist = 'SELECT \* FROM usertb'  userlistone = "SELECT \* FROM usertb WHERE id = '%s'"  userinsert = "Insert INTO usertb VALUES('%s','%s','%s')"  userdelete = "DELETE FROM usertb WHERE id = '%s'"  userupdate = "UPDATE usertb SET pwd = '%s', name = '%s' WHERE id = '%s'"  itemlist = 'SELECT \* FROM itemtb'  itemlistone = "SELECT \* FROM itemtb WHERE id = %d"  iteminsert = "Insert INTO itemtb VALUES (null,'%s',%d,CURRENT\_DATE(),'%s')"  itemdelete = "DELETE FROM itemtb WHERE id = %d"  itemupdate = "UPDATE itemtb SET name = '%s', price = %d, imgname = '%s' WHERE id = %d"   1. 구동 화면   Homepage    User 추가화면  Item 리스트화면    Item detail화면   1. 구현 소스 – 일부만 제출 |
| 1. UserTb, ItemTb를 이용한 CRUD 프로그램 개발(frame.userdb.py)  class UserDb(Db):  def selectone(self, id):  conn = self.getConnection()  cursor = conn.cursor()  sql = Sql.userlistone%(id)  cursor.execute(sql)  u = cursor.fetchone()  user = User(u[0],u[1],u[2])  self.close(conn,cursor)  return user  def select(self):  conn = self.getConnection()  cursor = conn.cursor()  sql = Sql.userlist  cursor.execute(sql)  result = cursor.fetchall()  all = []  for u in result:  user = User(u[0],u[1],u[2])  all.append(user)  self.close(conn,cursor)  return all    def insert(self, id,pwd,name):  try:  conn = self.getConnection()  cursor = conn.cursor()  sql = Sql.userinsert % (id, pwd, name)  cursor.execute(sql)  conn.commit()  except:  conn.rollback()  raise Exception  finally:  self.close(conn,cursor)  def delete(self, id):  try:  conn = self.getConnection()  cursor = conn.cursor()  sql = Sql.userdelete % (id)  cursor.execute(sql)  conn.commit()  except:  conn.rollback()  raise Exception  finally:  self.close(conn,cursor)  def update(self, id,pwd,name):  try:  conn = self.getConnection()  cursor = conn.cursor()  sql = Sql.userupdate % (pwd, name, id)  cursor.execute(sql)  conn.commit()  except:  conn.rollback()  raise Exception  finally:  self.close(conn,cursor) |
| 2. HTML5 기반 Web Client Application 구현(base.html)  {% load static %}  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Title</title>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>  <link rel="stylesheet" type="text/css" href="{% static 'css/shop2.css' %}">  <script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=a188d022f9e19591c7c1200b7aca5610"></script>  <style>  </style>  </head>  <body>  {% block content %}  {% endblock %}  </body>  </html> |
| 3. Django 환경 셋팅 및 웹 서버 환경 셋팅  Django startapp myshop  setting.py의 INSTALLED\_APPS에 myshop추가  STATICFILES\_DIRS, templates 경로 설정 |
| 4. Python을 이용한 Web Server Application 구현  python manage.py runserver |
| 5. Django에서의 Session 구현  request.session['suser'] = id  로그인 한 뒤 그 정보를 session에 저장한다. |
| 6. 지도 API를 이용한 지도 구현  <style>  #map{  widows: 800px;height:800px;  }  </style>  <script>  $(document).ready(function(){  var mapContainer = document.getElementById('map'), // 지도를 표시할 div  mapOption = {  center: new kakao.maps.LatLng(37.557112, 126.923904), // 지도의 중심좌표  level: 3 // 지도의 확대 레벨  };  var map = new kakao.maps.Map(mapContainer, mapOption); // 지도를 생성합니다    // 마커를 표시할 위치입니다  var position = new kakao.maps.LatLng(37.557112, 126.923904);  // 마커를 생성합니다  var marker = new kakao.maps.Marker({  position: position,  clickable: true // 마커를 클릭했을 때 지도의 클릭 이벤트가 발생하지 않도록 설정합니다  });  // 아래 코드는 위의 마커를 생성하는 코드에서 clickable: true 와 같이  // 마커를 클릭했을 때 지도의 클릭 이벤트가 발생하지 않도록 설정합니다  // marker.setClickable(true);  // 마커를 지도에 표시합니다.  marker.setMap(map);  // 마커를 클릭했을 때 마커 위에 표시할 인포윈도우를 생성합니다  var iwContent = '<div style="padding:5px;">LoveCafe!</div><div style="padding:5px;">LoveCafe! <br><a href="https://map.kakao.com/link/map/LoveCafe!,37.557112, 126.923904" style="color:blue" target="\_blank">큰지도보기</a> <a href="https://map.kakao.com/link/to/LoveCafe!,33.450701,126.570667" style="color:blue" target="\_blank">길찾기</a></div>', // 인포윈도우에 표출될 내용으로 HTML 문자열이나 document element가 가능합니다  iwPosition = new kakao.maps.LatLng(37.557112, 126.923904); //인포윈도우 표시 위치입니다  // var iwContent = '<div style="padding:5px;">LoveCafe!</div>', // 인포윈도우에 표출될 내용으로 HTML 문자열이나 document element가 가능합니다  iwRemoveable = true; // removeable 속성을 ture 로 설정하면 인포윈도우를 닫을 수 있는 x버튼이 표시됩니다  // 인포윈도우를 생성합니다  var infowindow = new kakao.maps.InfoWindow({  content : iwContent,  removable : iwRemoveable  });  // 마커에 클릭이벤트를 등록합니다  kakao.maps.event.addListener(marker, 'click', function() {  // 마커 위에 인포윈도우를 표시합니다  infowindow.open(map, marker);  });  });  </script>  <section>  <div id="map">    </div>  </section>  카카오 서비스를 활용하여 지도 생성  클릭 이벤트나 여러 필요한 기능들을 추가 |
| 7. Logger를 이용한 Log 파일 생성  LOGGING = {  'version': 1,  # 기존의 로깅 설정을 비활성화 할 것인가?  'disable\_existing\_loggers': False,  # 포맷터  # # 로그 레코드는 최종적으로 텍스트로 표현됨  # # 이 텍스트의 포맷 형식 정의  # # 여러 포맷 정의 가능  'formatters': {  'format1': {  'format': '[%(asctime)s] %(levelname)s [%(name)s:%(lineno)s] %(message)s',  'datefmt': '%d/%b/%Y %H:%M:%S'  },  'format2': {  'format': '%(levelname)s %(message)s'  },  },  # 핸들러  # 로그 레코드로 무슨 작업을 할 것인지 정의  # 여러 핸들러 정의 가능  'handlers': {  # 로그 파일을 만들어 텍스트로 로그레코드 저장  'file': {  'level': 'DEBUG',  'class': 'logging.FileHandler',  'filename': LOG\_FILE,  'formatter': 'format1',  },  # 콘솔(터미널)에 출력  'console': {  'level': 'DEBUG',  'class': 'logging.StreamHandler',  'formatter': 'format2',  }  },  # 로거  # 로그 레코드 저장소  # 로거를 이름별로 정의  'loggers': {  'users': {  'handlers': ['file'],  'level': 'DEBUG',  },  'items': {  'handlers': ['console'],  'level': 'DEBUG',  }  },  }  로그를 생성하고 views로 이동  logger = logging.getLogger('users')  logger를 설정한 뒤 원하는 곳에서  logger.debug('user id:'+id) 추가 |