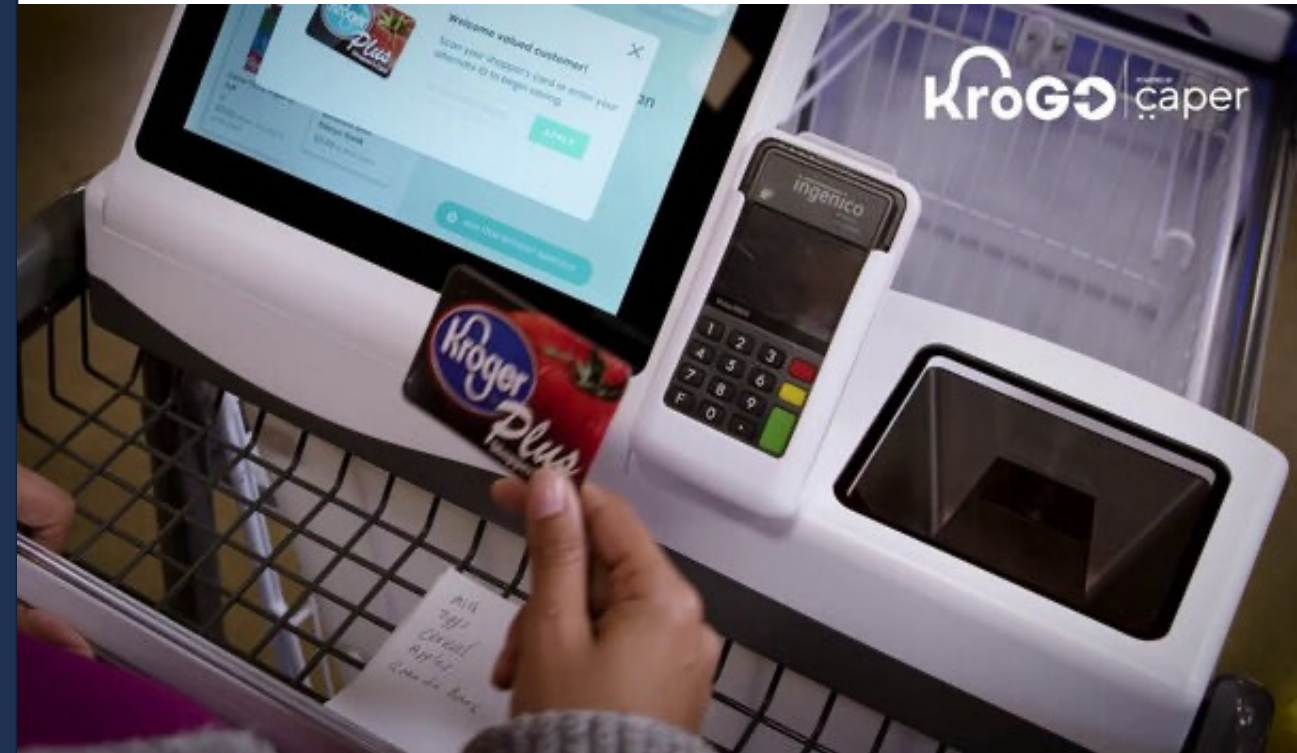


Smart Cart



목차

1. 프로젝트
팀 소개

3. 프로젝트 개요

5. 사용된 부품 소개

2. 문제인식

4. 시스템 동작 관련
기술적인 내용

6. Q&A

프로젝트 팀 소개

- 팀명: 삼점일사

- 팀원 별 역할

- 김인후(팀원): LCD 화면 출력,바코드 스캐너 코딩 및 설계, 데이터 베이스 코드 작성, 서버 구축 및 발표

- 김유림(팀장): NFC 코딩 및 바코드 스캐너 코딩 및 설계, (결과물에 활용되지 못했던) 적외선, 블루투스 센서 코딩 및 설계, 보고서 작성, PPT 제작

설계 목적

문제인식(대면으로 인한 바이러스 감염, 정확성이 낮고 신속하지 않은 계산(효율성이 낮음), 불필요한 인력 낭비)

경제성

고객이 직접 결제하는 시스템이므로 최소한의 인원만을 사용해 인건비 절감 효과를 생성

+

기능

초음파 센서를 통해 물체를 인식해 바코드 작동, 카트에서 상품 바코드 스캔, NFC를 통한 결제 및 입출입

+

편리성

줄을 기다려 결제할 필요 없이 그 자리에서 스캔, 서비스시 발생할 수 있는 사소한 피로감 차단

+

청결성

상호간의 대면을 줄일 수 있어 바이러스 확산 방지에 효과적 (EX) COVID-19)

프로젝트 개요



마트에 입장을 할때,
NFC 스캔을 통해 사용자 인식 후 입장

>



물건을 바코드 스캐너

>



명칭, 갯수, 가격이 LCD화면에 전달



여러 물품을 바코드 스캔 후 LCD화면

>



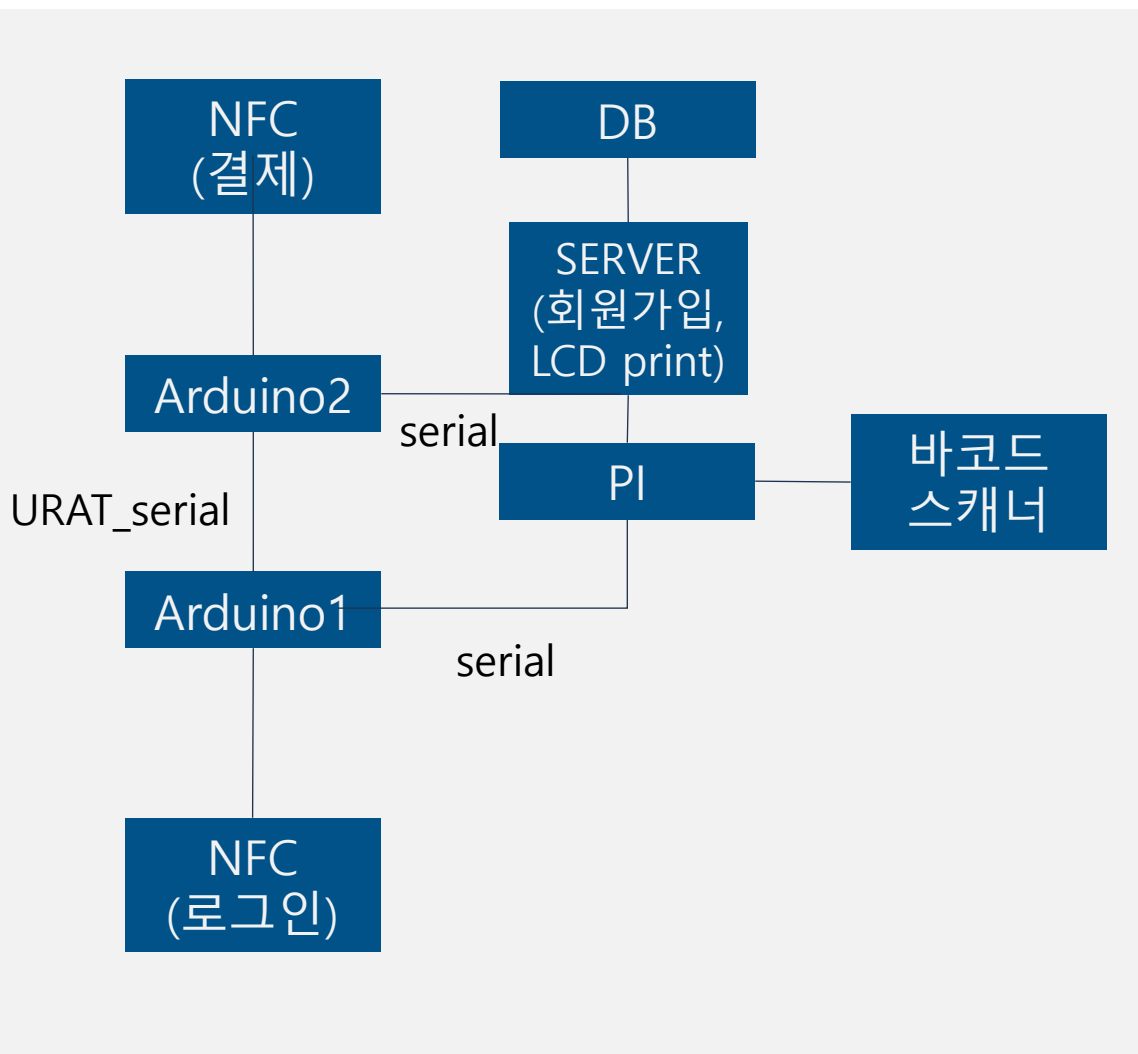
nfc 카드를 이용해 사용자를 인식하고
자동으로 결제가 진행됨.

>



결제 완료 후 디지털 화면

구조도 및 기술적인 내용



사용한 언어

- C / C++
- Python
- JavaScript
- Html / Css
- Node.js, Express.js, MongoDB

사용한 기능

- Mongo DB
- SERVER
- NFC
- Brcode
- Urat.Serial
- Serial
- Lcd touch display
- RASBERRYPI
- Arduino

사용된 부품 소개



소프트웨어-아두이노



하드웨어-아두이노 우노



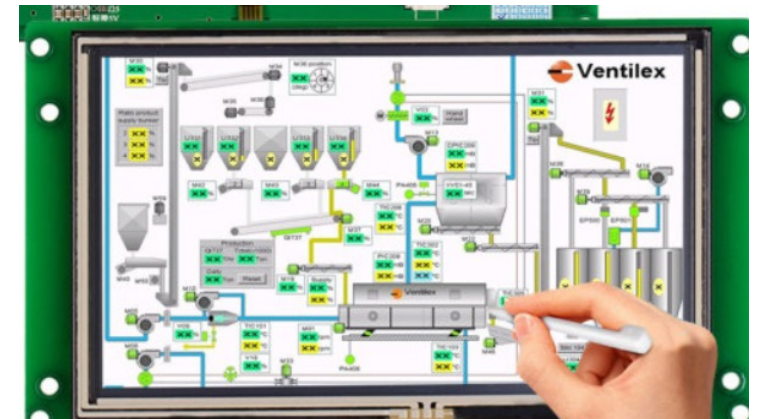
하드웨어-바코드스캐너



하드웨어-아두이노 RFID-RC522



하드웨어-라즈베리파이4B



하드웨어-5인치 lcd 패널

Q&A

감사합니다.

센서 코드

▼ WEB_회원가입

▼ 폴더 구조 & code

- models
 - User.js

```
const mongoose = require('mongoose')

// schema
const userSchema = mongoose.Schema({
  username:{type:String, required:[true,'Username is required!'], unique:true},
  password:{type:String, required:[true,'Password is required!'], select:false},
  name:{type:String, required:[true,'Name is required!']},
  money : {type : Number}
},{
  toObject:{virtuals:true}
})

// virtuals
userSchema.virtual('passwordConfirmation')
  .get(()=>{ return this._passwordConfirmation; })
  .set((value)=>{ this._passwordConfirmation=value; })

userSchema.virtual('originalPassword')
  .get(()=>{ return this._originalPassword; })
  .set((value)=>{ this._originalPassword=value; })

userSchema.virtual('currentPassword')
  .get(()=>{ return this._currentPassword; })
  .set((value)=>{ this._currentPassword=value; })

userSchema.virtual('newPassword')
  .get(()=>{ return this._newPassword; })
  .set((value)=>{ this._newPassword=value; })

// password validation
userSchema.path('password').validate((v) => {
  const user = this;

  // create user
  if(user.isNew){
    if(!user.passwordConfirmation){
      user.invalidate('passwordConfirmation', 'Password Confirmation is required.')
    }

    if(user.password !== user.passwordConfirmation) {
      user.invalidate('passwordConfirmation', 'Password Confirmation does not matched!')
    }
  }

  // update user
  if(!user.isNew){
    if(!user.currentPassword){
      user.invalidate('currentPassword', 'Current Password is required!')
    }
    else if(user.currentPassword !== user.originalPassword){
      user.invalidate('currentPassword', 'Current Password is invalid!')
    }

    if(user.newPassword !== user.passwordConfirmation) {
      user.invalidate('passwordConfirmation', 'Password Confirmation does not matched!')
    }
  }
});

// model & export
const User = mongoose.model('user',userSchema)
module.exports = User
```

- public

- CSS
 - master.css
- routes
 - home.js

```
const express = require('express')
const router = express.Router()

// Home
router.get('/', (req, res)=>{
  res.render('home/welcome')
})
router.get('/about', (req, res)=>{
  res.render('home/about')
})

module.exports = router
```

- users.js

```
const express = require('express')
const router = express.Router()
const User = require('../models/User')

// Index
router.get('/', (req, res)=>{
  User.find({})
    .sort({username:1})
    .exec((err, users)=>{
      if(err) return res.json(err);
      res.render('users/index', {users:users});
    })
})

// New
router.get('/new', (req, res)=>{
  res.render('users/new')
})

// create
router.post('/', (req, res)=>{
  User.create(req.body, (err, user)=>{
    if(err) return res.json(err)
    res.redirect('/users')
  })
})

// show
router.get('/:username', (req, res)=>{
  User.findOne({username:req.params.username}, (err, user)=>{
    if(err) return res.json(err);
    res.render('users/show', {user:user});
  })
})

// edit
router.get('/:username/edit', (req, res)=>{
  User.findOne({username:req.params.username}, (err, user)=>{
    if(err) return res.json(err)
    res.render('users/edit', {user:user})
  })
})

// update
router.put('/:username', (req, res, next)=>{
  User.findOne({username:req.params.username})
    .select('password')
    .exec(function(err, user){
      if(err) return res.json(err)

      // update user object
      user.originalPassword = user.password;
      user.password = req.body.newPassword? req.body.newPassword : user.password;
```

```

    for(var p in req.body){
      user[p] = req.body[p]
    }

    // save updated user
    user.save((err, user)=>{
      if(err) return res.json(err)
      res.redirect('/users/'+user.username);
    })
  })
})

// destroy
router.delete('/:username', (req, res)=>{
  User.deleteOne({username:req.params.username}, (err)=>{
    if(err) return res.json(err)
    res.redirect('/users')
  })
})

module.exports = router

```

- views

- home

- about.ejs

```

<!DOCTYPE html>
<html>
  <head>
    <%- include('../partials/head') %>
  </head>
  <body>
    <%- include('../partials/nav') %>

    <div class="container mb-3">

      <h2 class="mb-3">About</h2>

      <p>이 사이트는 삼점일사 조의 김인후, 김유림이 제작 하였습니다.</p>

    </div>
  </body>
</html>

```

- welcome.ejs

```

<!DOCTYPE html>
<html>
  <head>
    <%- include('../partials/head') %>
  </head>
  <body>
    <%- include('../partials/nav') %>

    <div class="container mb-3">

      <div class="jumbotron">
        <h1>My Website</h1>
        <p>제 웹사이트를 방문해 주셔서 감사합니다!</p>
      </div>

    </div>
  </body>
</html>

```

- users

- edit.ejs

```

<!DOCTYPE html>
<html>
  <head>
    <%= include('../partials/head') %>
  </head>
  <body>
    <%= include('../partials/nav') %>

    <div class="container mb-3">

      <h3 class="mb-3">Edit User</h3>

      <form action="/users/<%= user.username %>?_method=put" method="post">

        <div class="form-group row">
          <label for="currentPassword" class="col-sm-3 col-form-label">Current Password</label>
          <div class="col-sm-9 col-sm-offset-3">
            <input type="password" id="currentPassword" name="currentPassword" value="" class="form-control">
          </div>
        </div>

        <hr></hr>

        <div class="form-group row">
          <label for="username" class="col-sm-3 col-form-label">Username</label>
          <div class="col-sm-9">
            <input type="text" id="username" name="username" value="<%= user.username %>" class="form-control">
          </div>
        </div>

        <div class="form-group row">
          <label for="name" class="col-sm-3 col-form-label">Name</label>
          <div class="col-sm-9">
            <input type="text" id="name" name="name" value="<%= user.name %>" class="form-control">
          </div>
        </div>

        <div class="form-group row">
          <label for="money" class="col-sm-3 col-form-label">Money</label>
          <div class="col-sm-9">
            <input type="text" id="money" name="money" value="<%= user.money %>" class="form-control">
          </div>
        </div>

        <div class="form-group row">
          <label for="newPassword" class="col-sm-3 col-form-label">New Password</label>
          <div class="col-sm-9 col-sm-offset-3">
            <input type="password" id="newPassword" name="newPassword" value="" class="form-control">
          </div>
        </div>

        <div class="form-group row">
          <label for="passwordConfirmation" class="col-sm-3 col-form-label">Password Confirmation</label>
          <div class="col-sm-9 col-sm-offset-3">
            <input type="password" id="passwordConfirmation" name="passwordConfirmation" value="" class="form-control">
          </div>
        </div>

        <p>
          <small>*Required</small>
        </p>

        <div class="buttons">
          <a class="btn btn-primary" href="/users/<%= user.username %>">Back</a>
          <button type="submit" class="btn btn-primary">Submit</button>
        </div>

      </form>

    </div>
  </body>
</html>

```

- index.ejs

```

<!DOCTYPE html>
<html>
  <head>
    <%= include('../partials/head') %>
  </head>
  <body>
    <%= include('../partials/nav') %>

    <div class="container mb-3">

      <h3 class="mb-3">Users</h3>

      <ul class="list-group">
        <% if(users == null || users.length == 0){ %>
          <li class="list-group-item"> There is no user yet.</li>
        <% } %>
        <% users.forEach(function(user) { %>
          <li class="list-group-item">
            <a href="/users/<%= user.username %>"><%= user.username %></a>
          </li>
        <% }) %>
      </ul>

    </div>
  </body>
</html>

```

■ new.ejs

```

<!DOCTYPE html>
<html>
  <head>
    <%= include('../partials/head') %>
  </head>
  <body>
    <%= include('../partials/nav') %>

    <div class="container mb-3">

      <h3 class="contentBoxTop mb-3">New User</h3>

      <form action="/users" method="post">

        <div class="form-group row">
          <label for="username" class="col-sm-3 col-form-label">Username*</label>
          <div class="col-sm-9">
            <input type="text" id="username" name="username" value="" class="form-control">
          </div>
        </div>
        <div class="form-group row">
          <label for="name" class="col-sm-3 col-form-label">Name*</label>
          <div class="col-sm-9">
            <input type="text" id="name" name="name" value="" class="form-control">
          </div>
        </div>
        <div class="form-group row">
          <label for="money" class="col-sm-3 col-form-label">Money</label>
          <div class="col-sm-9">
            <input type="text" id="money" name="money" value="" class="form-control">
          </div>
        </div>
        <div class="form-group row">
          <label for="password" class="col-sm-3 col-form-label">Password*</label>
          <div class="col-sm-9">
            <input type="password" id="password" name="password" value="" class="form-control">
          </div>
        </div>
        <div class="form-group row">
          <label for="passwordConfirmation" class="col-sm-3 col-form-label">Password Confirmation*</label>
          <div class="col-sm-9 col-sm-offset-3">
            <input type="password" id="passwordConfirmation" name="passwordConfirmation" value="" class="form-control">
          </div>
        </div>
        <p>
          <small>*Required</small>
        </p>
      </form>
    </div>
  </body>
</html>

```

```

        <div class="form-group">
          <button type="submit" class="btn btn-primary">Submit</button>
        </div>
      </form>

    </div>
  </body>
</html>

```

- show.ejs

```

<!DOCTYPE html>
<html>
  <head>
    <%= include('../partials/head') %>
  </head>
  <body>
    <%= include('../partials/nav') %>

    <div class="container mb-3">

      <h3 class="contentBoxTop"><%= user.username %></h3>

      <form class="user-form" action="/users" method="post">
        <fieldset disabled>
          <div class="form-group row">
            <label for="name" class="col-sm-3 col-form-label">Name</label>
            <div class="col-sm-9">
              <input class="form-control" type="text" id="name" name="name" value="<%= user.name %>">
            </div>
          </div>
          <div class="form-group row">
            <label for="money" class="col-sm-3 col-form-label">Money</label>
            <div class="col-sm-9">
              <input class="form-control" type="text" id="money" name="money" value="<%= user.money %>">
            </div>
          </div>
        </fieldset>
      </form>

      <div>
        <a class="btn btn-primary" href="/users">Back</a>
        <a class="btn btn-primary" href="/users/<%= user.username %>/edit">Edit</a>
        <form action="/users/<%= user.username %>?_method=delete" method="post" class="d-inline">
          <a class="btn btn-primary" href="javascript:void(0)" onclick="confirm('Do you want to delete this?')?this.pa
        </form>
      </div>

    </div>
  </body>
</html>

```

- partials

- head.ejs

```

<meta name="viewport" content="width=device-width,initial-scale=1">

<!-- jquery & bootstrap -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6i
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y10
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8j0I
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
<!-- web font -->
<link href="https://fonts.googleapis.com/css?family=Open+Sans&display=swap" rel="stylesheet">

<!-- my css -->
<link rel="stylesheet" href="/css/master.css">

<!-- my js -->
<script src="/js/script.js"></script>

```

```
<title>My Website</title>
```

- nav.ejs

```
<nav class="navbar navbar-expand-sm navbar-light bg-light mb-3">
  <div class="container">
    <div class="navbar-brand">My Website</div>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav">
        <li class="nav-item"><a href="/" class="nav-link">Home</a></li>
      </ul>
      <ul class="navbar-nav ml-auto">
        <li class="nav-item"><a href="/users" class="nav-link">Users</a></li>
        <li class="nav-item"><a href="/users/new" class="nav-link">Sign Up</a></li>
      </ul>
    </div>
  </div>
</nav>
```

- index.js

```
const express = require('express')
const mongoose = require('mongoose')
const bodyParser = require('body-parser')
const methodOverride = require('method-override')
const app = express()

// DB setting
mongoose.connect('mongodb+srv://dmudmu:1234@dmudmu.5exnx.mongodb.net/myFirstDatabase?retryWrites=true&w=majority', {
}).then(() => console.log('DB ON!'))
.catch(error => console.log(error))

// Other settings
app.set('view engine', 'ejs')
app.use(express.static(__dirname+'/public'))
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({extended:true}))
app.use(methodOverride('_method'))

// Routes
app.use('/', require('./routes/home'))
app.use('/users', require('./routes/users'))

// Port setting
const port = 8080
app.listen(port, function(){
  console.log('SERVER ON')
})
```

- .prettierrc

```
{
  "semi" : false
}
```

▼ WEB_LCD

▼ 폴더 구조 & code

- models
 - Contact.js

```
//set
const mongoose = require('mongoose')
```



```
//schema
const contactSchema = mongoose.Schema({
  name:{type:String},
  price:{type:Number}
})

//model
const Contact = mongoose.model('contact', contactSchema)

//export
module.exports = Contact
```

- public

- CSS

- master.css

```
.contact h2{
  color: tomato;
}
.contact .contact-form label,
.contact-show label{
  display: inline-block;
  width: 70px;
}
.contact .contact-menu form{
  display: inline-block;
}
```

- routes

- contacts.js

```
const express = require('express')
const router = express.Router()
const mongoose = require('mongoose')
const Contact = require('../models/Contact')
const User = require('../../sibal2/board/models/User')
const SerialPort = require('serialport')
const { Readline } = require('serialport/lib/parsers')
const sp = new SerialPort('/dev/ttyACM0',{
  baudRate:9600
})
const parser = sp.pipe(new Readline({delimiter:'\r\n'}))
let num
let pay
let menu1
let menu2

// Index
router.get('/', (req,res)=>{
  if(num == 'login'){
    res.render('contacts/second')
  } else{
    res.render('contacts/first')
    num = 'login'
  }
})

router.get('/first', (req,res)=>{
  if(num == 'login'){
    res.render('contacts/second')
    num = 'next'
  } else{
    res.render('contacts/first')
  }
})

router.get('/menu', (req, res)=>{
  Contact.find({}, (err, contacts)=>{
    if(err) return res.json(err);
    res.render('contacts/home', {contacts:contacts});
  });
});
```

```

    });
  });

  router.get('/menu', (req,res)=>{
    if(pay=='pay'){
      res.render('contracts/finish')
    }else{
      res.render('contracts/home')
      pay = 'pay'
    }
  })

  // New
  router.get('/new', (req, res)=>{
    res.render('contacts/new')
  })

  // create
  router.post('/', (req, res)=>{
    Contact.create(req.body, (err, contact)=>{
      if(err) return res.json(err)
      res.redirect('/contacts')
    })
  })

  // show
  router.get('/:id', (req, res)=>{
    Contact.findOne({_id:req.params.id}, (err, contact)=>{
      if(err) return res.json(err)
      res.render('contacts/show', {contact:contact})
    })
  })

  //Sensor
  let total = 0
  sp.on('readable', ()=>{
    const str = sp.read().toString('utf-8')
    arr = str.split('')
    console.log(arr[0])
    if(arr[0] == 'q'){
      console.log('login')
      num = 'login'
    } else if(arr[0] == 'w'){
      console.log('pay')
      pay = 'pay'
    } else if(arr[0] == 'A'){
      console.log('apple')
      menu1 = 'apple'
      total ++
    } else if(arr[0] == '4'){
      console.log('banana')
      menu2 = 'banana'
      total ++
    }
  })

  sp.on('error', (err)=>{
    console.log('error', err.message)
  })

  module.exports = router

```

- home.js

```

const express = require('express')
const router = express.Router()

// Home
router.get('/', (req, res)=>{
  res.redirect('/contacts')
})

module.exports = router

```

- views

- contacts

- first.ejs

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h2>LOGIN</h2>
</body>
</html>
```

- second.ejs

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h2><%= contact.name %>님 안녕하세요.</h2>
</body>
</html>
```

- home.ejs

```
<!DOCTYPE html>
<html>
<head>
  <%= include('../partials/head') %>
</head>
<body>

  <div class="contact contact-index">
    <h2>CART</h2>
    <ul>
      <%= contacts.forEach(function(contact) { %>
        <li>
          <a href="../contacts/<%= contact._id %>"><%= contact.name %></a> <%= contact.price %>원
        </li>
      <%= }) %>
    </ul>
  </div>
</body>
</html>
```

- finish.ejs

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h2>thank you</h2>
</body>
</html>
```

- partials

- head.ejs

```
<link rel="stylesheet" href="/css/master.css">
<title>CART</title>
```

- nav.ejs

```
<nav>
  <div>Contact Book</div>
  <ul>
    <li><a href="/contacts">Index</a></li>
    <li><a href="/contacts/new">New</a></li>
  </ul>
</nav>
```

- index.js

```
//Module setting
const express = require('express')
const mongoose = require('mongoose')
const bodyParser = require('body-parser')
const methodOverride = require('method-override')
const app = express()

// DB setting
mongoose.connect('mongodb+srv://dmudmu:1234@dmudmu.5exnx.mongodb.net/myFirstDatabase?retryWrites=true&w=majority', {
}).then(() => console.log('DB ON!'))
.catch(error => console.log(error))

// Other setting
app.set('view engine', 'ejs')
app.use(express.static(__dirname+'/public'))
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({extended:true}))
app.use(methodOverride('_method'))

// Routes
app.use('/', require('./routes/home'))
app.use('/contacts', require('./routes/contacts'))

// Port setting
const port = 3000
app.listen(port, ()=>{
  console.log('SERVER ON')
})
```

- .prettierrc

```
{
  "semi" : false
}
```

▼ Arduino_One

```
#include <SoftwareSerial.h>
#include <SPI.h>
#include <MFRC522.h>

SoftwareSerial A_Serial(3,5);
#define RST_PIN 9
#define SS_PIN 10
```

```

MFRC522 rc522(SS_PIN, RST_PIN);

void setup() {
  A_Serial.begin(9600);
  Serial.begin(9600);
  SPI.begin();
  rc522.PCD_Init();
}
void loop() {
  if(!rc522.PICC_IsNewCardPresent() || !rc522.PICC_ReadCardSerial()){
    if(rc522.uid.uidByte[0] == 160 && rc522.uid.uidByte[1] == 163 && rc522.uid.uidByte[2] == 161 && rc522.uid.uidByte[3] == 165){
      A_Serial.write('q');
      for(int i = 0; i<4; i++){
        rc522.uid.uidByte[i] = 0;
      }
      delay(1000);
    }
  }

  if(Serial.available()){
    A_Serial.write(Serial.read());
  }
}

```

▼ Arduino_Two

```

#include <SoftwareSerial.h>
#include <SPI.h>
#include <MFRC522.h>

SoftwareSerial A_Serial(3,5);
#define RST_PIN 9
#define SS_PIN 10

MFRC522 rc522(SS_PIN, RST_PIN);

void setup(){
  A_Serial.begin(9600);
  Serial.begin(9600);
  SPI.begin();
  rc522.PCD_Init();
}
char sibal = 'r';
void loop(){
  if(!rc522.PICC_IsNewCardPresent() || !rc522.PICC_ReadCardSerial()){
    if(rc522.uid.uidByte[0] == 160 && rc522.uid.uidByte[1] == 163 && rc522.uid.uidByte[2] == 161 && rc522.uid.uidByte[3] == 165){
      Serial.write('w');
      for(int i = 0; i<4; i++){
        rc522.uid.uidByte[i] = 0;
      }
      delay(1000);
    }
  }
  if(A_Serial.available()){
    Serial.write(A_Serial.read());
  }
}

```

▼ RaspberryPi

```

import RPi.GPIO as GPIO
import time
import serial

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

trig = 2
echo = 3
count = 0
lst = [10]
distance = 0

GPIO.setup(trig, GPIO.OUT)
GPIO.setup(echo, GPIO.IN)

```

```

try :
    while True :
        GPIO.output(trig, False)
        time.sleep(0.5)

        GPIO.output(trig, True)
        time.sleep(0.00001)
        GPIO.output(trig, False)

        while GPIO.input(echo) == 0:
            pulse_start = time.time()

        while GPIO.input(echo) == 1:
            pulse_end = time.time()

        pulse_duration = pulse_end - pulse_start
        distance = pulse_duration * 17000
        distance = round(distance, 2)
        sp = serial.Serial('/dev/ttyACM0', 9600)

        if distance < 10 :
            print(distance)
            lst[count] = input()
            sp.write(lst[count])
            print(lst[count])
            count += 1

GPIO.cleanup

```

이번 프로젝트는 컴퓨터정보공학과를 다니면서 다른 전공자와 함께한 첫 프로젝트였습니다. 처음에는 제 실력이 부족한 걸 누구보다 잘 알고 있기 때문에 다른 팀원에게 피해가 가진 않을까, 내가 가장 잘할 수 있는 건 어떤 걸까라는 생각에 프로젝트가 시작하기 전부터 하루하루 지날수록 스트레스가 심해졌습니다. 하지만 팀원인 인후가 방향성과 제작부분에 있어서 잘 이끌어주어 프로젝트를 잘 마무리 할 수 있었습니다.

처음에 각자 아이디어를 생각해냈습니다. 저는 ‘살균 무드등’을 생각해왔고 다른 팀원은 ‘스마트 카트’를 생각했습니다. 어떤 아이디어를 선택지에 대해 창의성, 완성도 그리고 준비 기간 등 여러 가지 방면으로 생각해본 결과 더 창의적이고 기술성이 높은 스마트 카트가 적합해 선택하게 되었습니다. 각자 파트를 나눠 저는 센서 위주로 코드를 작성하고 다른 팀원은 DB 와 웹 페이지 위주로 코드를 작성했습니다. 혼자하는 프로젝트가 아닌만큼 어려움이 많았습니다. 제가 짠 코드가 팀원의 소스코드와 연결시 연동이 되지 않아 작동하지 않아 블루투스연결과 적외선 센서 코드를 다 버리게 되어 속상한 일도 있었습니다. 하지만 그 과정 속에도 평소 잘 모르는 다른 센서의 코드에 대해 배울 수 있어 센서활용프로그래밍에 대한 이해도가 증가되었습니다. 팀 프로젝트를 하면서 팀원과 서로 즉각적인 피드백을 받을 수 있어서 좋았고 프로젝트 내용 관련 소스에 대해 공유하며 색다른 방법들을 통해 소스활용도를 높일 수 있게 되었습니다. 계획과 과정은 나름 잘 흘러갔지만 시간과 정보수집부족으로 프로젝트를 완성하지 못해 아쉽지만 조금이나마 실력을 성장시킬 수 있었고 팀원들 간에 협업도 경험하며 말 그대로 팀플레이가 무엇인가를 몸소 체험한 것 같습니다. 앞으로도 개발을 할 때 많은 팀 프로젝트들을 겪어야 하는데 이번 기회는 프로그래머가 꿈인 저에게 큰 밑거름이 되는 중요한 경험이었다고 생각합니다.