Course project report – Koko Coffee

Mingliang Ma

100400948

Info 1213 R95

1. Description of the Business

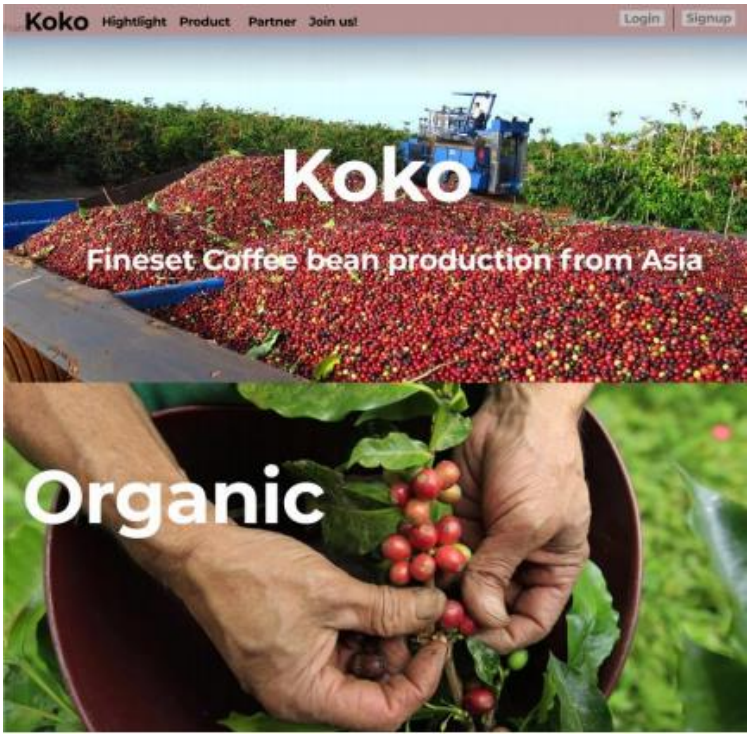
In this project, I managed to create 2 business webpages for a coffee business called ‘Koko coffee’. This company provides high quality organic coffee beans from Asia. Koko coffee is looking to promote their product as a coffee bean supplier to coffee stores like Starbucks and Tim Hortons with long-term contracts and large amount of coffee bean orders. They want to use their webpages to inform people who they are and what they do, and they also wish the webpage can inspire people to join their partnership, so they will have a section to allow people to submit their information.

2. Comparison

In the assignment 3, I made the webpage wireframes by Figma, now I will compare my actual webpages with the wireframes I made to see what have I achieved and lost.

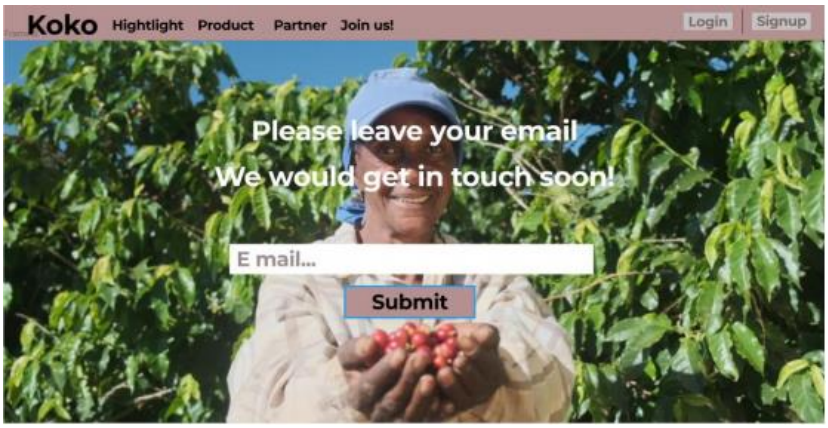
First, the color theme and pictures are different from the wireframes because I chose random photos and color when making the wireframe, and now It’s not possible to find the same photos. Therefore, I found some photos with same themes like coffee beans farms and workers to make the webpages.

Wireframe:

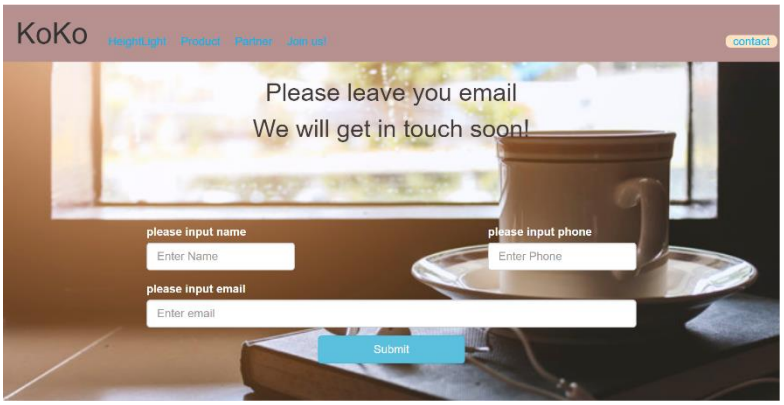


Actual webpage:

Second, the submit section is different too. I want the visitors to submit more information rather than just e-mail. It would help Koko company know their customers better. Overall, the actual webpages are different from the Figma wireframes, but it is more complete, more functional and more professional.
Wireframe:



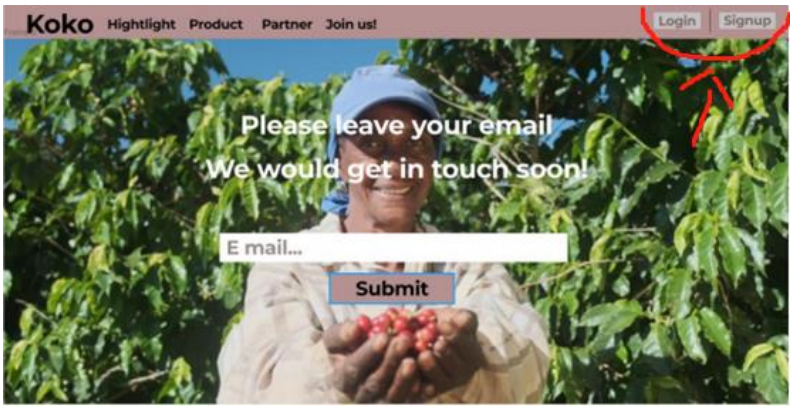
Webpage:



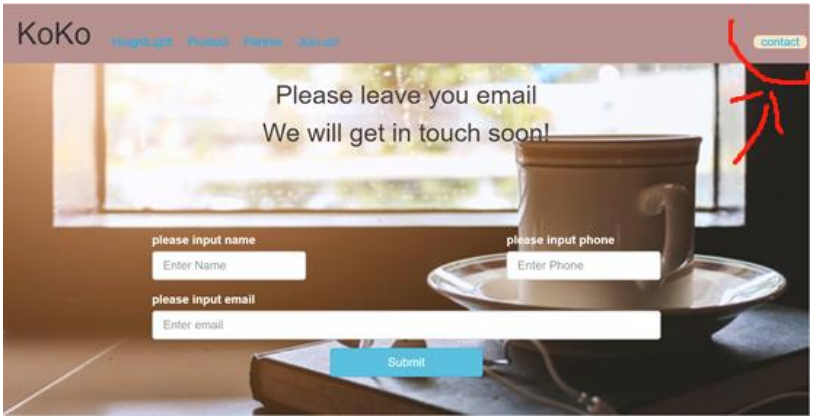
I also deleted the function of login and signup because I simply felt that a coffee supplier website would not need a membership function. Most suppliers either have an individual website for supply orders, or they simply receive orders by MSN and

sign contract in person. Therefore, I deleted this part, and make users to jump to second page where he/she can submit their information, so the company will get in touch with them soon for further cooperation.

Wireframe:



Webpage:



3. Instructions:

To run my project in GoormIDE, please follow the steps:

1. Access the project through: https://goor.me/V2Hbu
2. Setup MongoDB by the following steps:
    1. Open the 1st terminal and type the following command to install MongoDB

"apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4 && (echo "deb [arch=amd64] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse" | tee /etc/apt/sources.list.d/mongodb-org-4.0.list) && apt-get update && apt-get install -y mongodb-org && mkdir -p /data/db"

    2. Type "**mongod**" at the 1st terminal to run mongodb server.
    3. Open the 2nd terminal and type "**mongo**" to run mongodb client.
    4. On the terminal, type "cd /workspace/Koko-project", then type "npm i", then type "npm start", the webpages should now run.

4. Feature/Content Explanations

From business side, I used many photos as background in the webpages because I want to allow visitors to have a good expression to our company. And the

easy submission of visitor information is also convenient.

From technical perspective, I have done these functions:

1. I made the navbar hanging on the top of the webpages no matter how you scroll down. I made it by put the header's position to be "fixed" in css file.

```
.header {
  position: fixed;
  top: 0;
  z-index: 999;
  margin-bottom: 70px;
  width: 100%;
  height: 70px;
  display: flex;
  background-color: #B69190;
}
```

2. All information submitted in the second page will be collected in the database of MongoDB

```
    new TodoModel({
        contentName: req.body.contentName,
        contentPhone: req.body.contentPhone,
        contentEmail: req.body.contentEmail,
        updated_at: Date.now()
    }).save(function(err, todo, count) {
        console.log('content', todo, 'num', count);
        res.redirect('/');
    });
});
```

```
});
var ListSchema = new mongoose.Schema({
    user_id: String,
    contentName: String,
    contentPhone: String,
    contentEmail: String,
    updated_at: Date
});
```

3. All information submitted in the second page will be collected in a text file:

```
const {contentName, contentPhone, contentEmail} = req.body;
console.log('req.contentName', contentName);
var data = `Name: ${contentName}, Phone: ${contentPhone}, Email: ${contentEmail}`;


var writerStream = fs.createWriteStream('output.txt');


writerStream.write(data,'UTF8');


writerStream.end();
```

4. Recognize when users wants to make change on the webpage address, lead them to where they want to be:

```
var express = require('express');
var fs = require("fs");
var router = express.Router();
var mongoose = require('mongoose');
var TodoModel = mongoose.model('user');
var URL = require('url');
/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});
router.get('/contact', function(req, res, next) {
  res.render('contact.html');
});
router.post('/create', function(req, res) {
  console.log('req.body', req);
  const {contentName, contentPhone, contentEmail} = req.body;
  console.log('req.contentName', contentName);
  var data = `Name: ${contentName}, Phone: ${contentPhone}, Email: ${contentEmail}`;

```

5. Sending request from front-end webpages to back-end server when users submitting information, alert users when they submit empty slots:

```
                <button type="button" class="btn btn-info btn-sub" onclick="ceateItem()">Submit</b
            </div>
        </form>
    </div>
</body>

    <script src="http://cdn.bootcss.com/jquery/1.11.1/jquery.min.js"></script>
    <script src="http://cdn.bootcss.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>
    <script>
        var global = {};
        function ceateItem() {
            var obj = {};
            obj.contentName = $('.createContentName').val();
            obj.contentPhone = $('.createContentPhone').val();
            obj.contentEmail = $('.createContentEmail').val();
            if (obj.content != '') {
                $.post('/create', obj, function (res) {
                    alert("Submission complete!")
                })
            } else {
                alert('not null');
            }
        }
```