

# Shopping Assistant



The University of Adelaide

## Supervisors



Pro. Ali

## Group members



Qiaoyang Luo  
a1728768



Qiming Shao  
a1699631



Junwei Li  
a1710640



Dr. Lingqiao

# Overview – Goals



## Goals

**System:** Front-end and back-end

**Platform:** Customer and seller

**Application:** Android and iOS

**Model:** Identifying more products



# Overview – Objectives



## Update the system's structure

Use web technique to develop the seller subsystem

## Build cross-platform App

Use React Native and Redux to develop the App

## Change computer vision model

Use image retrieval model to recognize more products

# Difficulty



# Allocations

Front-end App



01

Back-end Server



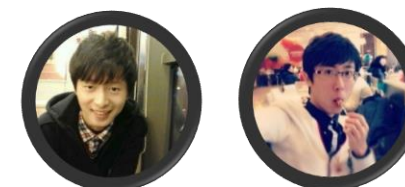
02

Front-end Web

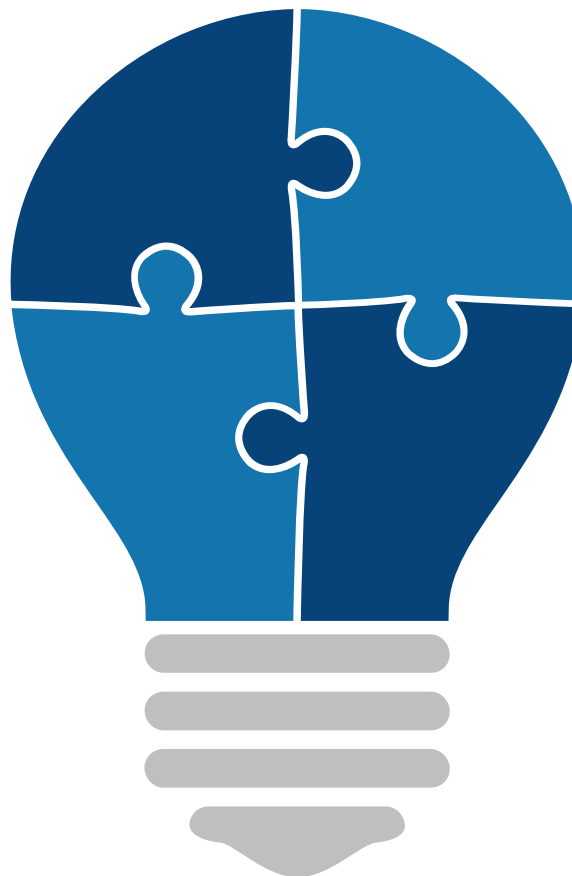


03

Research Model

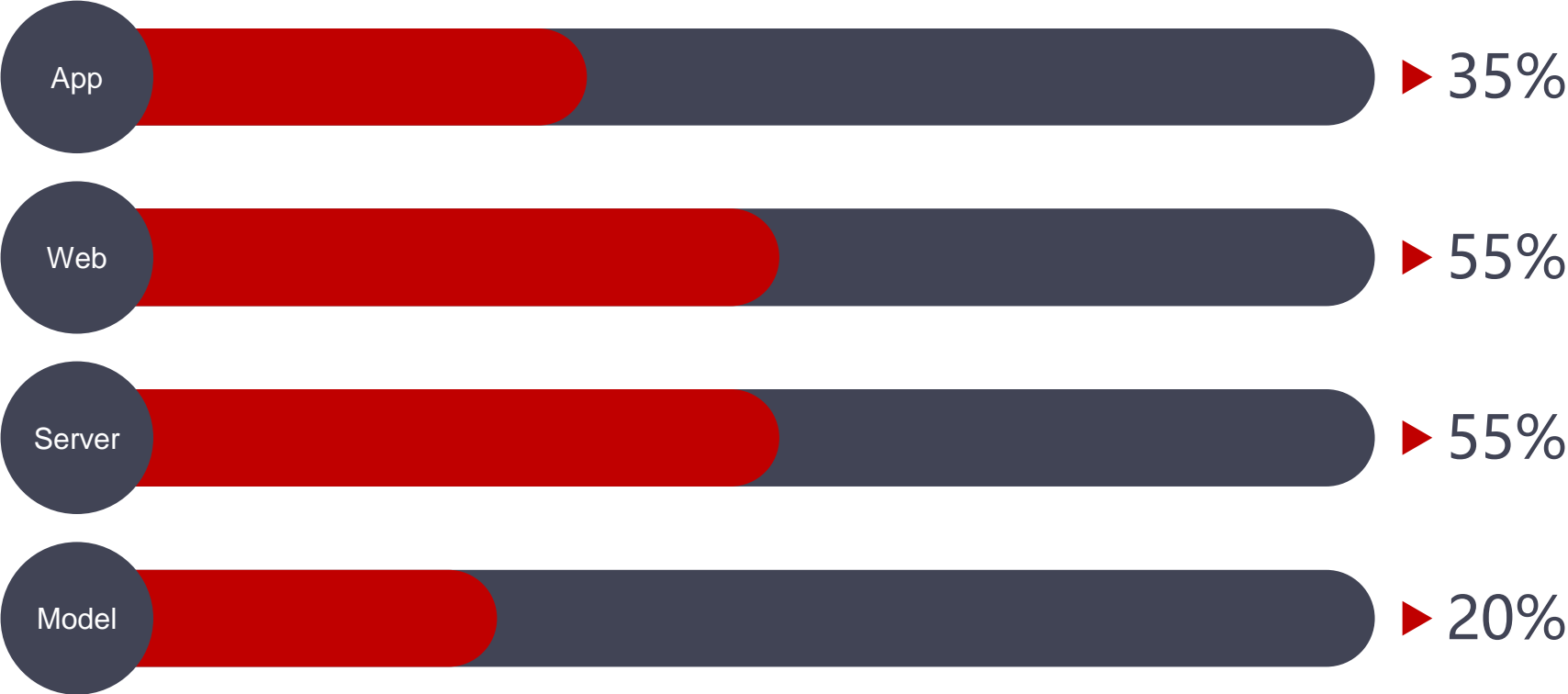


04



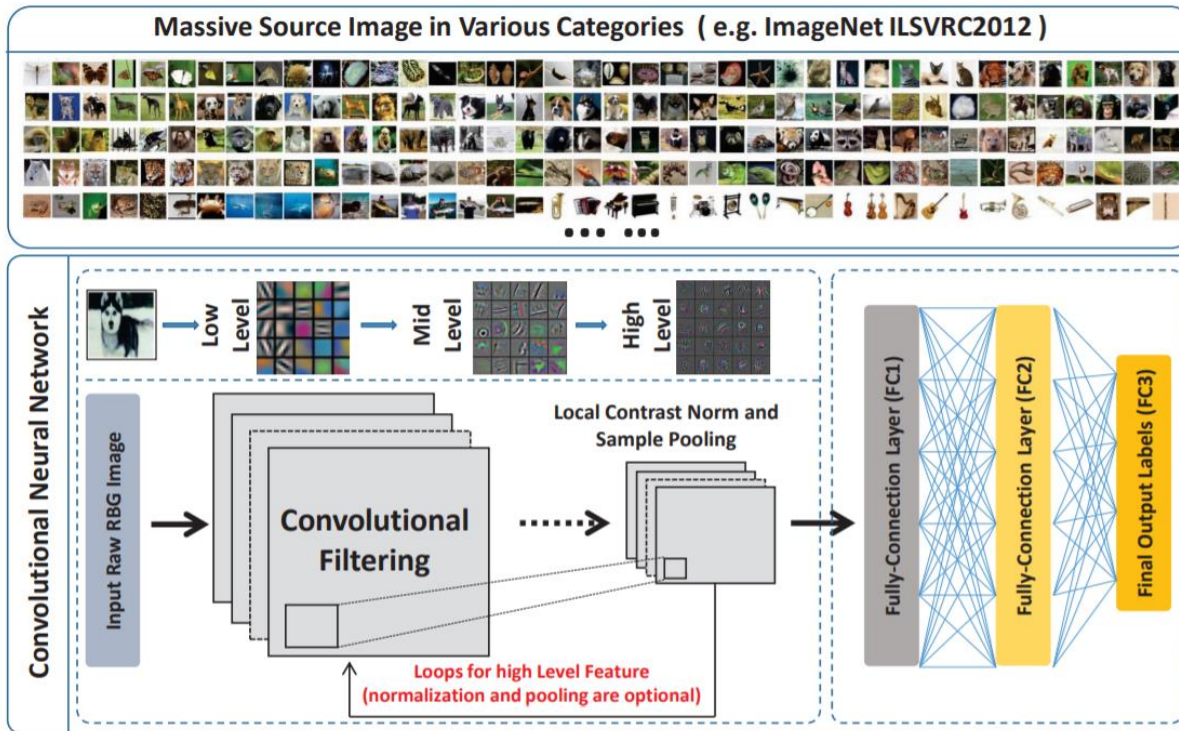


# Progress





# Image Retrieval Model



[1]

## Three steps

1. Select a Deep Learning Model
2. Refine the network with feature representation
3. Training the network with dataset

# React Native App



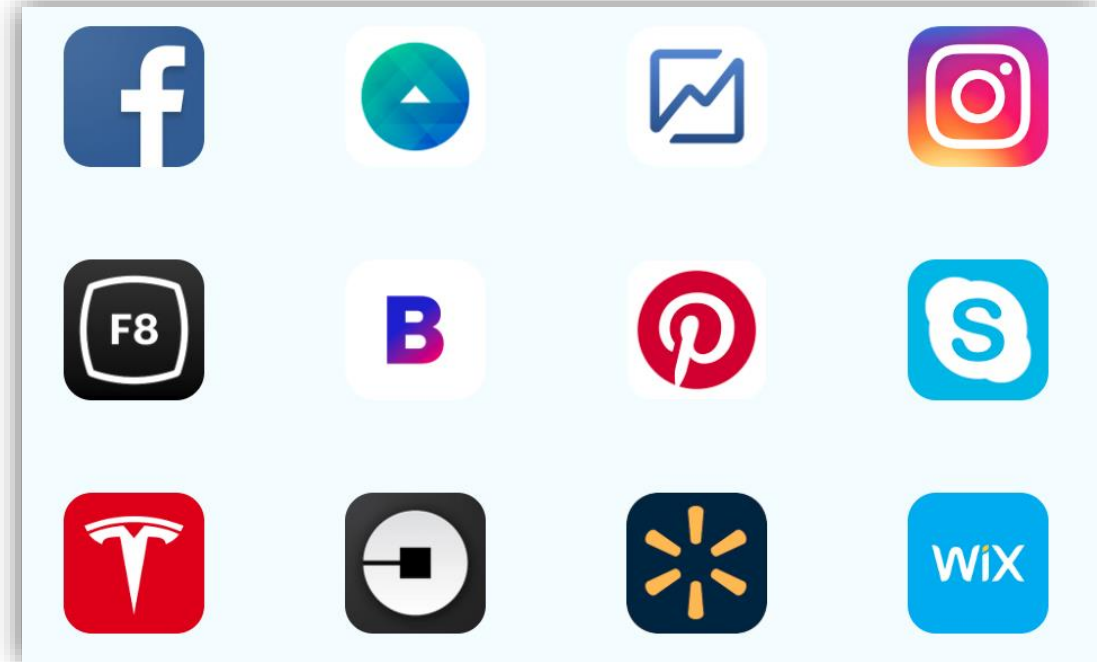
## What is React Native?

JavaScript + React



## Why React Native?

1. Only JavaScript
2. More powerful than HTML App
3. Many Companies choose it
4. Someone introduce it

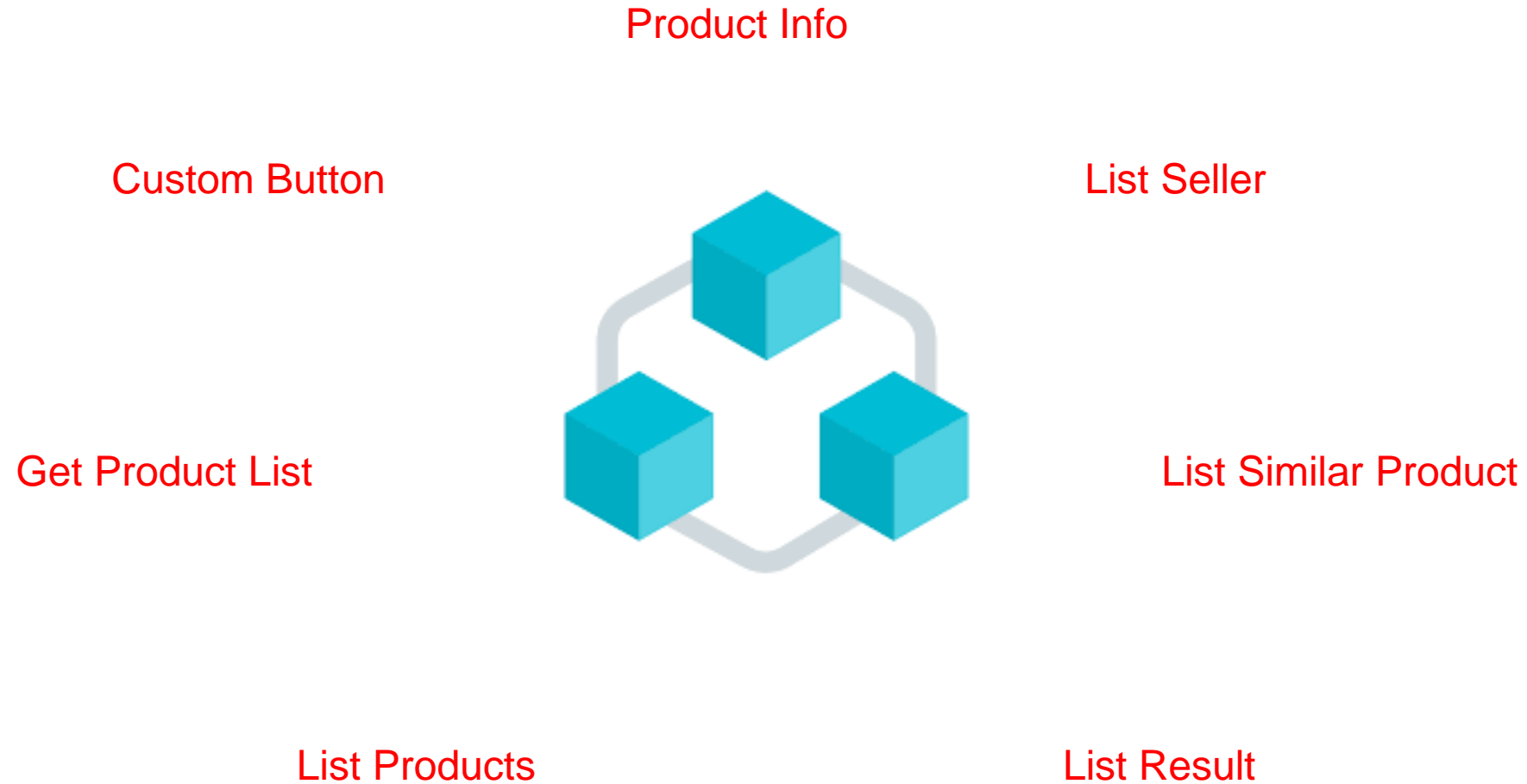




# App Progress - Module




# App Progress - Components



## App Progress - Demo



[Shopping AI](#) [Price](#) [Map](#) [Login](#)

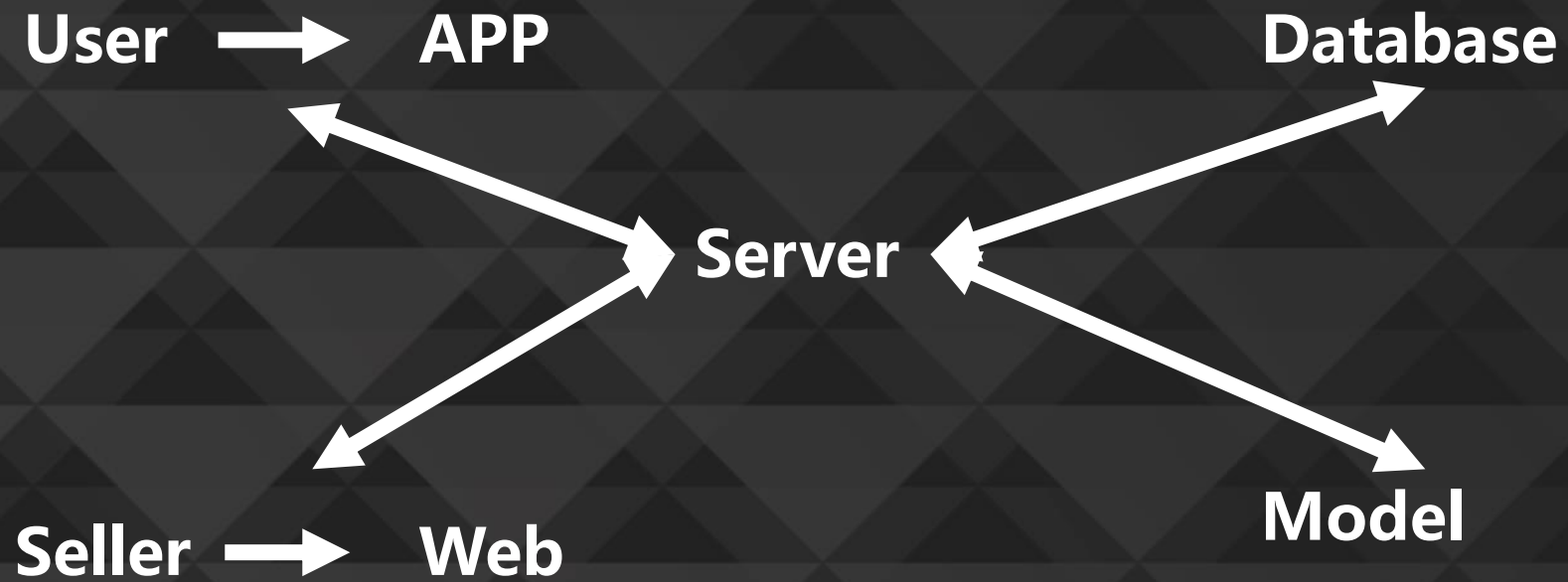
A photograph of a large, multi-story brick building with a prominent sign that reads "CENTRAL MARKET" in gold lettering. The building is set against a blue sky with scattered white clouds. The sign is mounted on a dark metal frame that extends over the entrance area.

Why

Motivation

# Why & Motivation

Necessary





## Current function

**Seller sign up/login**

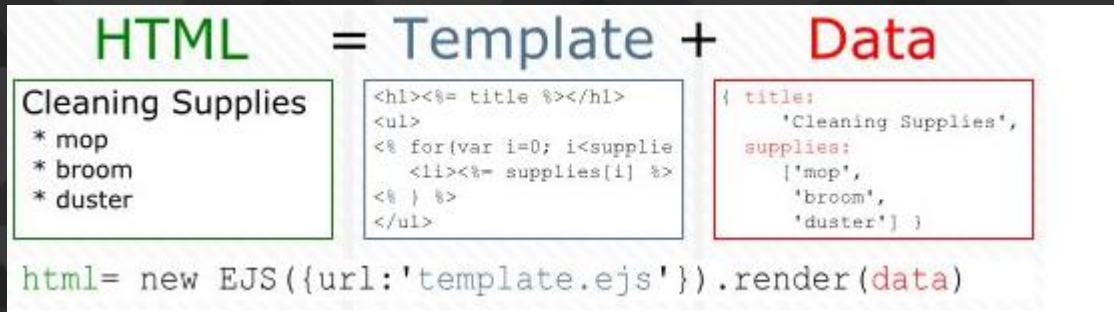
**Products Upload**

**Search products info**

**Update info(in progress)**



## 1. EJS is a JavaScript template library for generating HTML strings from JSON data.



## 2. Comparing with HTML

## 3. EJS can receive JSON format data sent asynchronously by WebService

# Logic: router in express

1. `var ljw = require('./routes/ljw')`

2. `app.use('/submittest',ljw);`


3. 

```
<form action="/submittest/ljw" method="get">
  <div class="divprice">
    <div class="divprice11">
      <%if(result[i].price2 == "") {%>
        ONLY AVAILABLE IN STORE
      <%}else{%>
        $<%=result[i].price2%>
      <%} %>
    </div>
  </div>
  <input type="text" name="newprice" value="">
  <input type="submit" value="submit">
</form>
```

4. 


```
1 var express = require('express');
2 var router = express.Router();
3
4 router.get('/ljw',function(req,res,next){
5   //console.log(req.query.newprice)
6   value = req.query.newprice;
7   //console.log(value);
8   res.render('ljw',{'newvalue' : value});
9 });
10
11 module.exports = router;
```

banana



banana2  
location:Coles

\$2for5



banana1  
location:Coles

\$5.0

```
GET /submittest/ljw?newprice=4.98 304 1.421 ms - -
GET /submittest/ljw?newprice=3.8 200 1.584 ms - 320
```

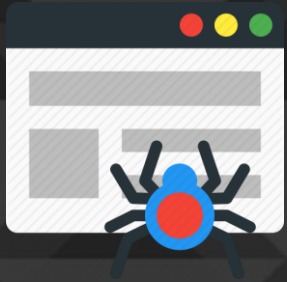
× Headers Preview Response Cookies Timing

**Request URL:** http://localhost:3000/submittest/ljw?newprice=3.8  
**Request Method:** GET  
**Status Code:** 304 Not Modified  
**Remote Address:** [::1]:3000  
**Referrer Policy:** no-referrer-when-downgrade

▼ Response Headers view source

**Connection:** keep-alive  
**Date:** Wed, 29 Aug 2018 01:40:44 GMT  
**ETag:** W/"140-1j5DIktCbKqHyiZkNl9dYZU53Qg"  
**X-Powered-By:** Express

# Data Collection



Crawler



Enormous products



Image retrieval



Scrapy

Screen capture ->  
Web crawling  
framework



Redis

Memory cache  
database ->  
Url queue

# Database



**MongoDB**



**Non-relational  
database**



**Distributed  
Computing**



**High scalability**

## products collection

```
{
  "_id" : ObjectId("5b7d263de9d2fd3e779f8d7e"),
  "name" : "Nestle CERELAC Muesli with Pear 8m + 200g ",
  "image" : "https://shop.coles.com.au/wcsstore/Coles-CAS/images/8/6/7/8674529-th.jpg",
  "location" : "Coles",
  "price1" : "4.58",
  "price2" : "4.58"
}
```

## photo collection

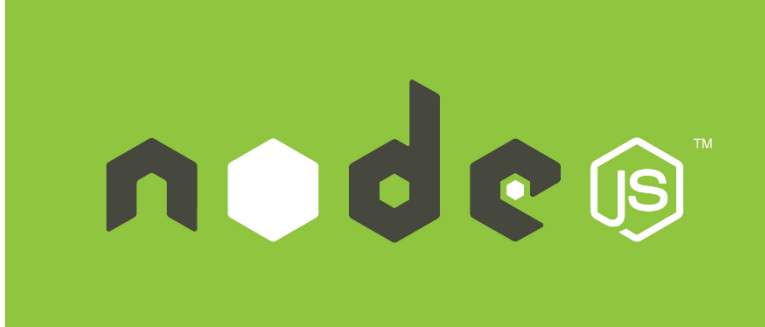
```
{
  "_id" : ObjectId("5b7e0591f01301339d26fc3b"),
  "path" : "files/1534985617051-photo.jpg",
  "caption" : "",
  "__v" : 0
}
```

## user collection

```
{
  "_id" : ObjectId("5b7e02573ccef53314ff3231"),
  "email" : "admin",
  "password" : "123",
  "dateCreated" : ISODate("2018-08-23T00:39:51.461Z")
}
```



## Back-end Server



## Node.JS



**Asynchronous**



**Event cycle (callback function)**



**Communication with APP**



**I/O function**



**Middleware**

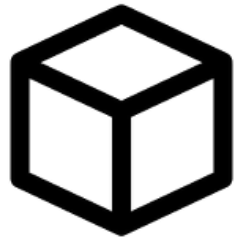


**Routing table**



**Dynamically render**

# Structure



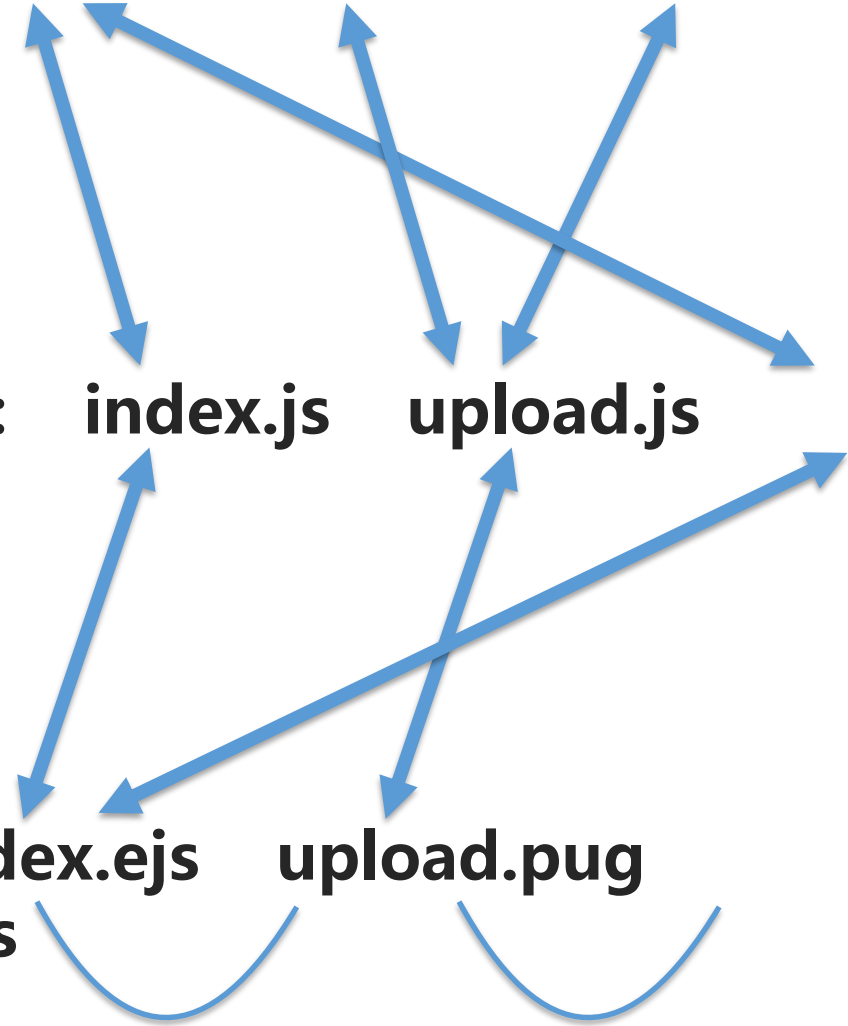
Model: db.js product.js photo.js



Controller: index.js upload.js  
user.js



View: index.ejs upload.pug  
search1.ejs



# Model migration

## Hybrid Programming



```
POST /login 302 121.911 ms - 58
GET /index2 200 1836.087 ms - 1506
GET /stylesheets/style.css 304 1.908 ms - -
watermelon
POST /upload 200 5202.099 ms - 21111
GET /css/bootstrap.min.css 200 2.908 ms - 121200
GET /js/bootstrap.min.js 304 5.115 ms - -
GET /js/smartsearch.js 200 7.958 ms - 6136
GET /js/jquery.min.js 200 9.260 ms - 86351
GET /js/in_search.js 200 3.465 ms - 2882
```

## Upload images

FILE

b.jpg

Caption

UPLOAD ➤

banana



Coles Kids Pack **Bananas** 750g

location:Coles

ONLY AVAILABLE IN  
STORE



Coles Deli Frozen Raw **Banana** Prawns 5kg Carton on special

location:Coles

ONLY AVAILABLE IN  
STORE



Coles **Banana** Yoghurt Pouch 70g

location:Coles

\$0.75



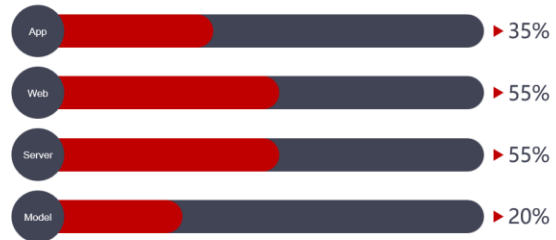
Coles **Bananas** approx. 180g

location:Coles

\$0.81

# Conclusion

## Progress



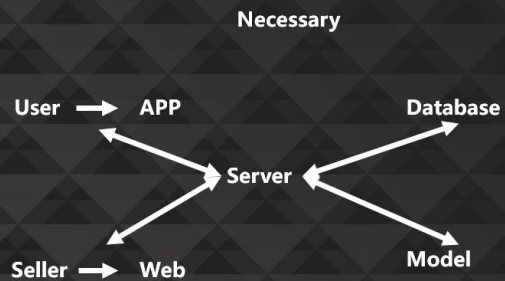
a1728768 6

## App Progress - Module



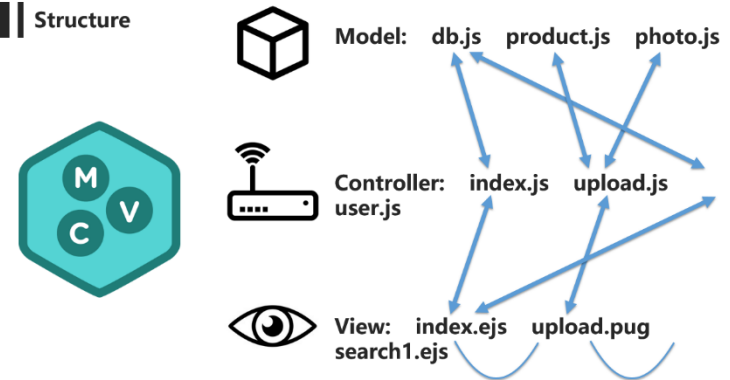
a1728768 9

## Why & Motivation



a1710640 13

## Structure



a1699631 11