

A Synopsis on

Krishisetu: Connecting Farmer and Consumer

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

in

Information Technology

by

Shreyas Chorge (17104022)
Vedangi Naigaonkar (16104046)
Abhijit Ambre (17104030)

Prof. Uttam Kolekar
Prof. Anagha Aher Prof. Neha Deshmukh



Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI
2020-2021

CERTIFICATE

This is to certify that the project Synopsis entitled "***Krishi Setu:Bridging gap between farmer and consumer***" Submitted by "***Shreyas Chorge (17104022), Vedangi Naigaonkar (16104046), Abhijit Ambre (17104030)***" for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology*** to the University of Mumbai, is a bonafide work carried out during academic year 2020-2021

(Prof. Anagha Aher) (Prof. Neha Deshmukh)
Co-Guide

(Dr. Uttam D.Kolekar)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane

Date:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Shreyas Chorge (17104022))

(Vedangi Naigaonkar (16104046))

(Abhijit Ambre(17104030))

Date:

Abstract

Farmers are not getting paid what they deserve because of all the middlemen like a distributor, retailers, vendors, etc. To overcome this situation, if a farmer wants to sell directly in the market the vendors and buyers, they tell them the margin rate which is not at all appropriate according to the current market rates. At this point Farmer cannot take back his goods and supplies since there is a risk of deterioration of goods, so the farmer has to sell his goods at margin rate determined by the vendors and buyers.

Introduction

Farmers these days do not get proper value for the farm produce. They strive so hard yet are compelled to sell their produce at a low price. The middleman forces the farmer to sell his produce at a low rate. These middlemen then sell the same farm produce in the cities or to distributors at a high cost. Thus, the middlemen are at an advantage while the farmers and vendors suffer a huge amount of loss. Sometimes, these middlemen also indulge in malpractices. To overcome this problem, we are making a cross-platform mobile application where these middlemen will almost be eliminated from the system and farmers can directly sell from farm to fork.

Objectives

We have the following objectives for our project which will help for the betterment of farmers.

- To build a distributed computing network with kubernetes
- To set up scalable image upload
- To set up Payments
- To let farmers to sell products
- To let consumers to buy products

Literature Review

[1] Leila Abdollahi Vayghan, Mohamed Aymen Saied, Maria Toeroe, Ferhat Khen-dek "Kubernetes as an Availability Manager for Micro service Applications"

The authors have presented a brief overview of Kubernetes architectural components and architectures for deploying micro services-based application with Kubernetes. The authors have investigated the impact of adding redundancy on the availability of micro service-based applications and performed experiments under Kubernetes default configuration and its most responsive one. They have also performed a comparative evaluation with the Availability Management Framework (AMF), a proven solution as a middleware service for managing high-availability. In Kubernetes, we have different master node components i.e Kube API-server, etcd, Kube-scheduler due to any failure if this single master node fails this cause a big impact on business. so to solve this issue we deploy multiple master nodes to provides high availability for a single cluster and improves performance.

[2] L Magnoni "Modern Messaging for Distributed Systems"

L Magnoni has discussed the Importance of loosely coupled communication, Connection-oriented communication, Messaging for loosely coupled communication, Messaging scenario, Messaging middleware. Author has also explained the basic Messaging Terminologies and compared different messaging services available and also their use cases.

[3] Pooja J Bhat, Priya D "Modern Messaging Queues - RabbitMQ, NATS and NATS Streaming"

Authors have explained in detail about the Messaging Technologies and Terminologies. Authors have also compared and explained the features of these technologies and provided the benchmark for RabbitMQ and NATS. They have also explained the architecture of RabbitMQ and NATS

Problem Definition

Farmers are not getting paid what they deserve because of all the middlemen like a distributor, retailers, vendors, etc. To overcome this situation, if a farmer wants to sell directly in the market the vendors and buyers they tell them the margin rate which is not at all appropriate according to the current market rates. At this point Farmer cannot take back his goods and supplies since there is a risk of deterioration of goods, so the farmer has to sell his goods at margin rate determined by the vendors and buyers.

Proposed System Architecture/Working

Krishi Setu is a scalable application that will help to eliminate the middle-man from the supply chain. It has a large scale distribution architecture. Design and Architecture are done as follow: Travis CI - Travis CI is a Continuous Integration Server (CI/CD pipeline), It will run tests on the code after it is pushed on to Github's Master Branch. If any test fails it will notify the developer that the current version of code is broken. If all tests pass it will build and deploy it to AWS. AWS - Project will be hosted on EC2 instance on AWS. After receiving the code AWS will build the code and deploy it on EBS. EBS will be used for Load Balancing. Nginx server - It will act as a proxy server that will distribute the outgoing data to the respective client-side or server-side based on the incoming request. React-Native - It will be used to develop a cross-platform client-side mobile application. Node Server - NodeJs runs an event-loop thus tends to portray single-threaded behaviour. We can make it work as a multi-thread processing by using cluster modules in NodeJs. Hence it will use all the available cores on the server's ec2 instance so even if a process blocks an event-loop Node will use the remaining cores to execute the incoming request. Redis - Redis Server would be used to cache the data at the server-side of the application MongoDB - MongoDB has been used as a database. AWS S3 - AWS S3 is used to store Images. Stripe Client - Stripe server is used to handle payments. Once the order's been made Stripe will hold the payment. It will be returned to the consumer's account. If the product hasn't been delivered within 7 days. Thus farmers don't have to worry about refunds Mapbox Client - Mapbox Server is to handle maps.

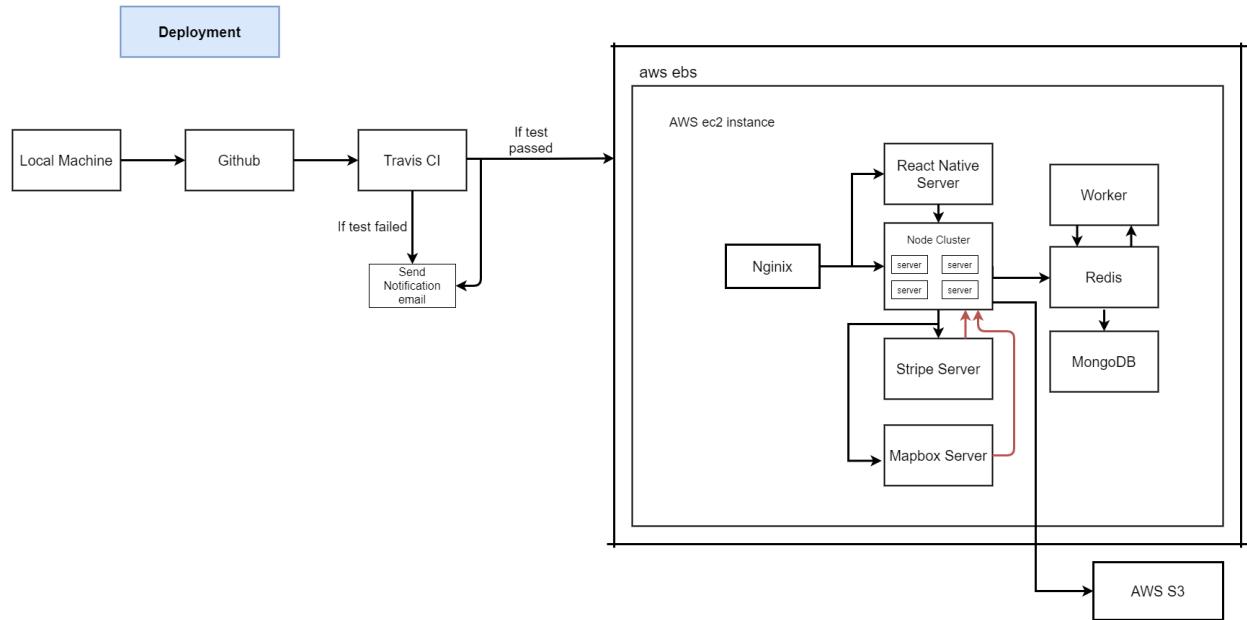


Figure 1: System Design Architecture

Design and Implementation

This is the Design and Implementation of Redux Store. Redux Store has been used to maintain the state of the application.

Redux Store Setup

```
import { createStore, applyMiddleware, compose } from 'redux';
import thunk from 'redux-thunk';
import rootReducer from 'src/reducers';

const initialState = {};
const middleware = [thunk];

let composeEnhancers = compose;

declare global {
    interface Window {
        __REDUX_DEVTOOLS_EXTENSION_COMPOSE__: typeof compose;
    }
}

if (__DEV__) {
    composeEnhancers = window.__REDUX_DEVTOOLS_EXTENSION_COMPOSE__ || compose
}

const store = createStore(
    rootReducer,
    initialState,
    composeEnhancers(applyMiddleware(thunk))
);

export default store;
```

Redux Reducer Setup:

```
import { combineReducers } from 'redux';
import { test } from "src/reducers/testReducer";

export default combineReducers({
    test: test,
});
```

Redux Reducer;

```

import { TEST_ACTION } from "src/actions/constants";

import { RootAction } from 'typesafe-actions';

interface testAction {
    type: string;
    payload: testBlock[];
}

interface testBlock {
    id: number;
    title: string;
    content: string;
}

const initialState = {
    data: []
};

export const test = (state = initialState, action: testAction) => {

    switch (action.type) {

        case TEST_ACTION:
            return {
                ...state,
                data: action.payload
            };

        default:
            return state;
    }
};

```

Redux Action Setup:

```

import { StateType, ActionType } from 'typesafe-actions';

declare module 'typesafe-actions' {
    export type Store = StateType<typeof import('src/store').default>;
}

export type RootState = StateType<typeof import('src/reducers').default>;

```

```
export type RootAction = ActionType<typeof import('src/actions').default>;  
  
interface Types {  
    RootAction: RootAction;  
}  
}  
  
}
```

Redux Action:

```
import axios from "axios";  
import { Dispatch } from "redux";  
  
import { TEST_ACTION } from "src/actions/constants";  
  
const getTest = () => async (dispatch: Dispatch) => {  
    const res = await axios.get('https://radiant-stream-39786.herokuapp.com/notes/');  
    console.log("In action", res);  
    dispatch({  
        type: TEST_ACTION,  
        payload: res.data,  
    });  
};  
  
export default {  
    getTest  
}
```

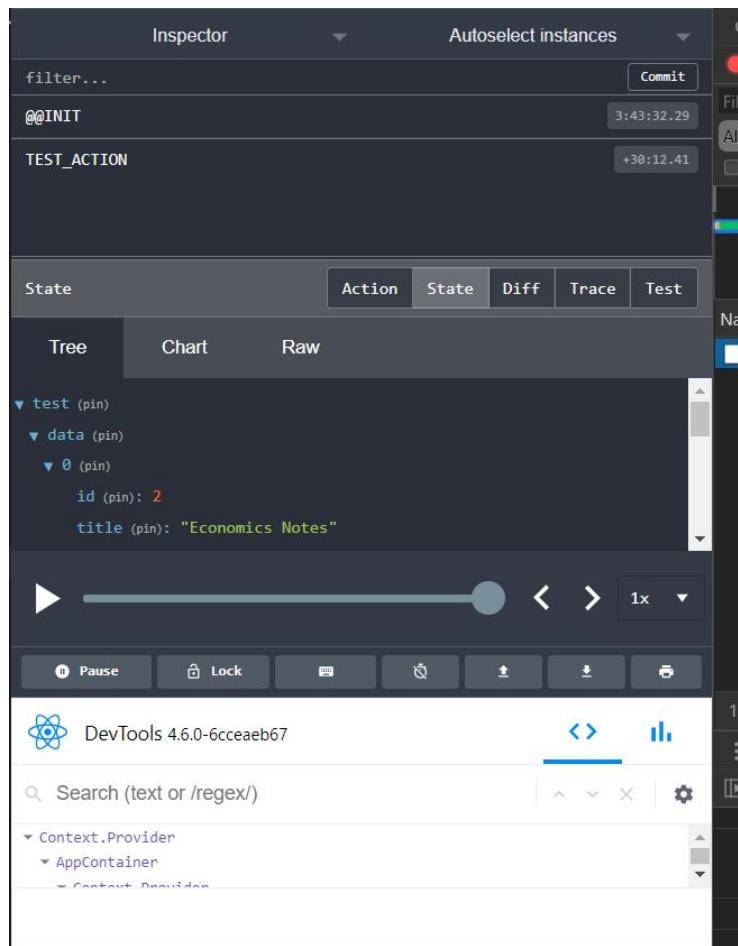


Figure 2: Redux Store

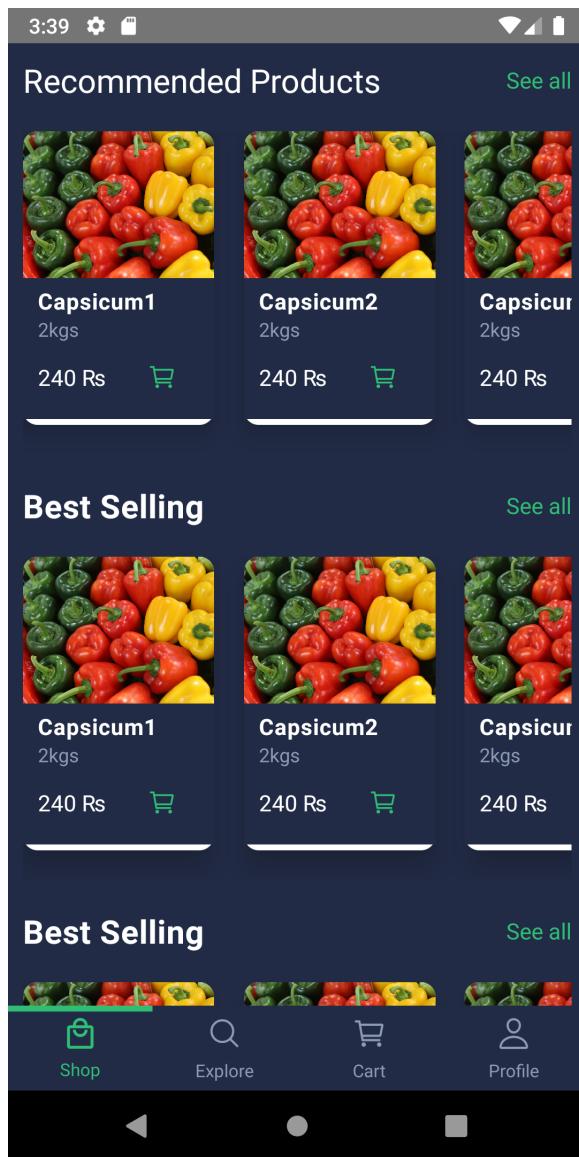


Figure 3: Home Page

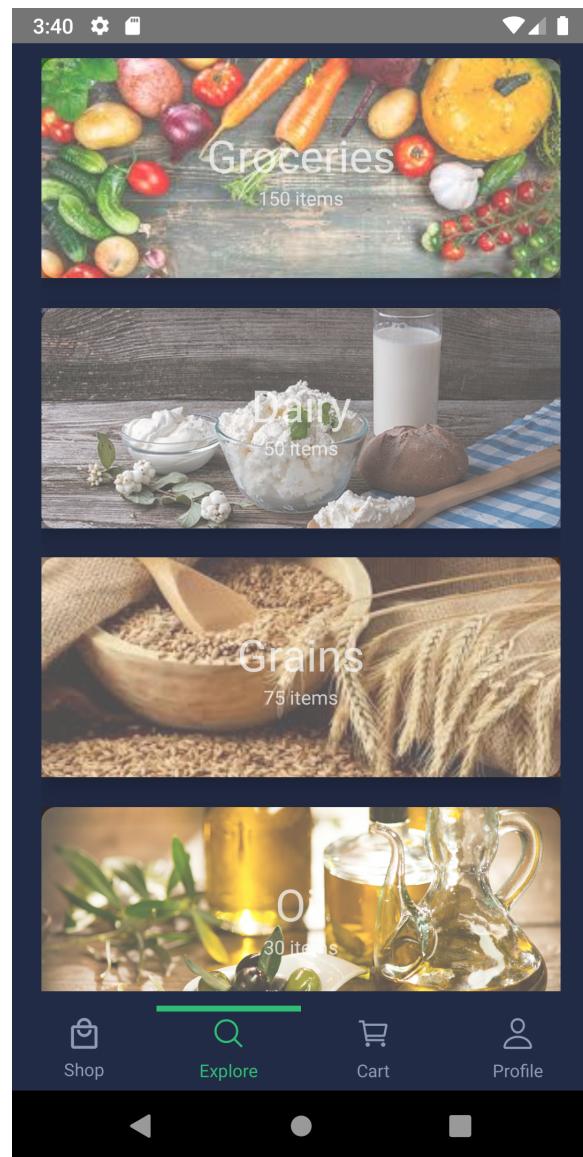


Figure 4: Categories

Summary

Krishi Setu is a scalable application that will help to eliminate the middle-man from the supply chain. It will have two users: farmer and consumer. The farmer will provide the products and consumers will pay for the product. The payment would be held by the server until and unless the product is been delivered to the consumer. If the product has not been delivered within seven days the payment will be reverted back to the consumer's bank account. Thus farmers do not have to worry about the refunds. Krishi Setu will have multi-linguistic support for the regional languages so farmer from any region will be able to use this application.

References

- [1] Leila Abdollahi Vayghan, Mohamed Aymen Saied, Maria Toeroe, Ferhat Khende, "Kubernetes as an Availability Manager for Microservice Applications", unpublished.
- [2] L Magnon "Modern Messaging for Distributed Systems", Journal of Physics Conference Series 608(1):012038
- [3] Poojya J Bhat, Priya D, "Modern Messaging Queues - RabbitMQ, NATS and NATS Streaming", International Journal of Recent Technology and Engineering (IJRTE)
- [4] Node JS Documentation: <https://nodejs.org/en/docs/>
- [5] PM2 Documentation: <https://pm2.keymetrics.io/docs/usage/cluster-mode/>
- [6] Redis Documentation: <https://redis.io/documentation>
- [7] React Native Documentation: <https://reactnative.dev/docs/getting-started>
- [8] UI-Kitten Documentation: <https://akveo.github.io/react-native-ui-kitten/docs/getting-started/what-is-ui-kittenwhat-is-ui-kitten>
- [9] TypeScript Documentation: <https://www.typescriptlang.org/docs/>
- [10] Axios Documentation: <https://github.com/axios/axios>
- [11] Docker Documentation: <https://docs.docker.com/get-started/>
- [12] Kubernetes Documentation: <https://kubernetes.io/docs/home/>
- [13] Expressjs Documentation: <https://expressjs.com/en/guide/routing.html>
- [14] AWS EC2 Documentation: <https://docs.aws.amazon.com/ec2/index.html>
- [15] AWS S3 Documentation: <https://docs.aws.amazon.com/s3/index.html>

1 Publication

- Paper entitled “Krishi Setu: Connecting Farmers and Consumers” is presented at “ICIRCA : 3rd IEEE International Conference on Inventive Research in Computing Applications” by “Shreyas Chorge”, “Vedangi Naigaonkar”, “Abhijit Ambre”.