

Abstract

In Nepal, most of the things that we buy are not from the supermarkets but from local shops. A frequent problem that can occur while buying from these shops is price variation of same commodities. Bazzar Mulya solves this problem by creating a location specific database of some of the daily used goods by collecting price info from the users and applying algorithms to select the minimum price among them. With this the user atleast know the minimum price of goods they are buying before going shopping. Such database would also allow people to compare prices of goods at different places, making it easier for them to decide what to buy when visiting another place.

Contents

1	Introduction	1
1.1	Objectives	1
1.2	Features	1
1.3	Limitations	1
1.4	Case Study	1
2	Discussion	2
2.1	Tools Used	2
2.1.1	HTML & CSS	2
2.1.2	Programming Languages	2
2.1.3	Libraries	2
2.1.4	IDE	2
3	System Overview	3
3.1	Algorithm	3
4	Gantt Chart	4
5	Conclusion	5
6	References	6

1 Introduction

During the immediate aftermath of 2072 earthquake, there was a lot of fluctuation in prices of different daily used items. Many goods were sold at higher prices than they were worth because of the state of panic. At times like these, one of the ways to accurately know the prices of goods is by asking the manufacturer. But this is not always possible. The other way, that we have come up with, is to analyze price of goods at all the neighboring shops, and find the least price, excluding special discounts. This can be done on a location basis, so we end up with price database of various locations.

1.1 Objectives

- To create a price database of daily commodities like rice, sugar etc.
- To create a web site and an android app so that users can view the price database.
- To create a mechanism so that the prices can be updated by data sent by users.
- To use a machine learning algorithm so that the prices can be updated automatically.

1.2 Features

By the time of completion of this program, we hope to include the following features.

- The database will hold the tentative prices of the daily used goods.
- The project will include an android app and a website.
- There will be multiple access level. Users, Shop Keepers, Location Admin and Main Admins.
- The price update process will be automatic.
- The price database will be location specific. Different places will have different database. This will allow price comparison between different place.
- The program will display comparison of prices of goods in two different locations.

1.3 Limitations

Bazzar Mulya will have following limitations.

- The algorithm depends on the user input. Without sufficient data it can't work properly.
- If most of the users send false data, the calculated price will also be false.
- It will include prices of only few items.

1.4 Case Study

In the developed countries, most of the data about prices is available online. Since there are many online shops, the users have the luxury of comparing prices between them and selecting the best option. In Nepal, the online business have just started taking off. Sites like Kaymu, Sasto Deals and Sabaiko Mart offer online shopping feature to costumers. But most of these sites only sell electrical equipment like laptops, cell phones, etc. They do not sell daily used items like rice, sugar etc. And again, most of the things are still bought from the actual shops, not online shops.

Similarly, a project called KaliMatiMarket lists the prices of vegetables at Kalimati, which is the biggest vegetable market of Nepal [1]. But price information on Kalimati doesn't help most users, as prices vary when they are distributed to local shops. It is also limited to vegetable prices.

2 Discussion

2.1 Tools Used

We have decided to use the following tools for the project.

2.1.1 HTML & CSS

HTML(the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. HTML and CSS are the basis of building Web pages and Web Application[2].

2.1.2 Programming Languages

2.1.2.1 Python Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or java. The language provides constructs intended to enable clear programs on both a small and a large scale[3]. We are using python(version 3) because it has libraries that we need -django and scikit-learn.

2.1.2.2 Java Script JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applicatios[4]. We are using javascript to make the web site responsive.

2.1.2.3 Java Java is a programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model[5]. In the project, java is used to develop android app.

2.1.3 Libraries

2.1.3.1 Django Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development[6]. We are using django to build back end of the website.

2.1.3.2 Scikit Learn Scikit-Learn is a python based collection of simple and efficient tools for data mining and data analysis built on NumPy, SciPy and Matplotlib libraries[7]. We are using scikit-learn for implementation of machine learning algorithm.

2.1.4 IDE

2.1.4.1 Andrsoid Studio Android studio is an intelligent code editor capable of advanced code completion,refactoring, and code analysis. It automatically downloads the necessary tools for android app development[8]. So, to develop the android app, we are using Android Studio.

3 System Overview

3.1 Algorithm

Step 1: Start

Step 2: Collect data from the users, via website and mobile application.

Step 3: Apply machine learning algorithm to select the best price from the entered data for each item.

Step 4: If the result of a particular location is unusual, notify the Location Admins.

Step 5: If there are many anomalies, notify the Root Admins.

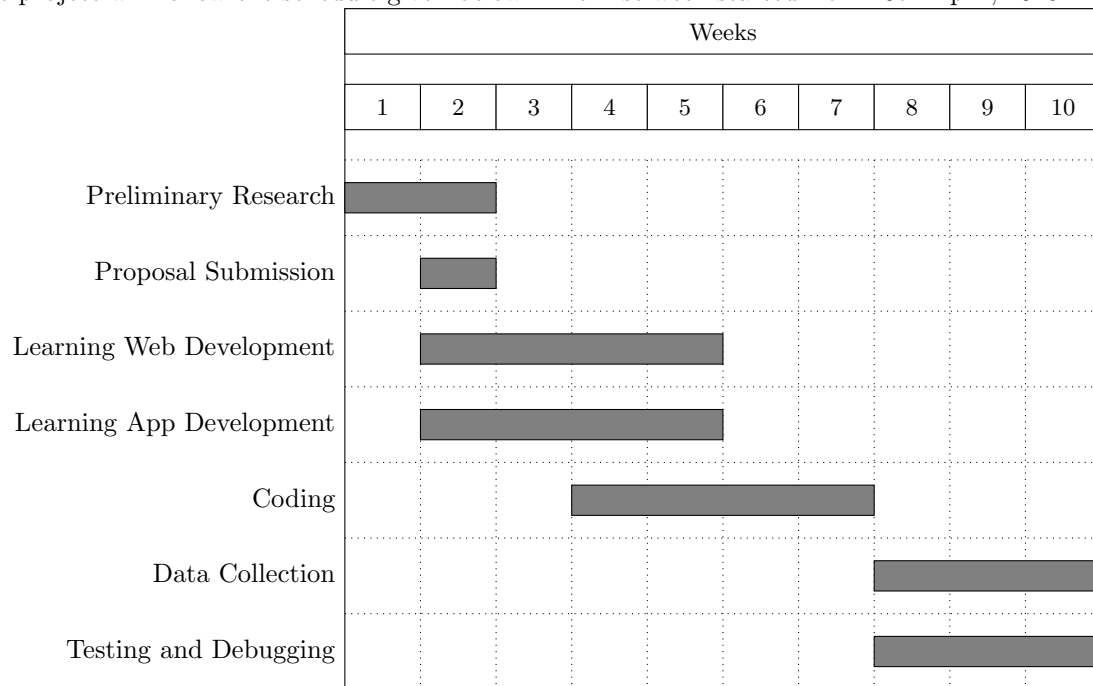
Step 6: Update the price database.

Step 7: Display the database in site and the mobile application.

Step 8: End

4 Gantt Chart

The project will follow the schedule given below. The first week started from 10th April, 2016.



5 Conclusion

The project aims to create a platform which will help people. The platform will be built based on the contributions of the people. With this project we hope to create a mechanism by which people helps others and themselves by contribution of information that is easily available to them.

6 References

- [1] *Leading Market For Vegetables and Fruits in Nepal*. URL: <http://kalimatimarket.com.np/home/language/EN> (visited on 05/15/2016).
- [2] *HTML & CSS - W3C*. URL: <https://www.w3.org/standards/webdesign/htmlcss.html> (visited on 05/15/2016).
- [3] *Python (programming language)*. URL: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) (visited on 05/15/2016).
- [4] *JavaScript Tutorial*. URL: <http://www.tutorialspoint.com/javascript/> (visited on 05/15/2016).
- [5] *Java Definition*. URL: <http://searchsoa.techtarget.com/definition/Java> (visited on 12/01/2015).
- [6] *Django: The Web framework for perfectionists with deadline*. URL: <https://www.djangoproject.com/> (visited on 05/15/2016).
- [7] *scikit-learn: machine learning in python*. URL: <http://scikit-learn.org/stable/> (visited on 05/15/2016).
- [8] *Android Studio Overview*. URL: <https://developer.android.com/tools/studio/index.html> (visited on 12/01/2015).