



PRODUCTZ HUNT - WEBAPP FOR PRICE COMPARISON

A MINI PROJECT REPORT

Submitted by

Darin Joshua D [REGISTER NO: 211418104043]

Gladwin Joseph Solomon B [REGISTER NO: 211418104064]

Joel Nithish Kumar M [REGISTER NO: 211418104102]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI - 600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

BONAFIDE CERTIFICATE

Certified that this project report “.....**Productz Hunt - WebApp for Price Comparison.....**” is the bonafide work of “.....**Darin Joshua D [REGISTER NO: 211418104043], Gladwin Joseph Solomon B [REGISTER NO: 211418104064], Joel Nithish Kumar M [REGISTER NO: 211418104102]....**” who carried out the project work under my supervision.

SIGNATURE

Dr.S.MURUGAVALLI,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

DR.S.BALAJI
ASSISTANT PROFESSOR
DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidates were examined in the Anna University Mini Project
Examination held on 10th August 2021

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C. VIJAYARAJESWARI, Thiru.C. SAKTHI KUMAR, M.E.** and **Tmt. SARANYASREE SAKTHI KUMAR B.E., M.B.A.**, for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of the CSE Department, **Dr. S. MURUGAVALLI , M.E.,Ph.D.** for the support extended throughout the project.

We would like to thank our Mini Project Supervisor **DR. S. BALAJI** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

Darin Joshua D
Gladwin Joseph Solomon B
Joel Nithish Kumar M

ABSTRACT

The large and growing industry of price comparison websites (PCWs) or ‘web aggregators’ is poised to benefit consumers by increasing competitive pricing pressure on firms by acquainting shoppers with more prices. However, these sites also charge firms for sales, which feeds back to raise prices. It's found that introducing a single PCW increases prices for all consumers, both shoppers and non-shoppers. More generally, in the most profitable equilibrium for competing PCWs, prices tend to rise with the number of PCWs. Productz Hunt gives ecommerce traders a good opportunity to boost their sales, attract new customers and go ahead against their competitors. Even Shoppers often look for best deals and offers for a particular product which they are looking for. It is difficult to visit each and every website for comparing the deals or the price of a particular product. Hence this comparison-shopping site is proposed which will help ecommerce merchants as well as shoppers for collecting product information, including price list, from participating retailers and then it displays collective information on a single result page in response to a shopper's or ecommerce merchant's search query. In this way, shoppers can compare prices, and service from multiple retailers on a single page and choose the merchant that offers the best overall value. Users need to open the website and search for a particular product, it will compare the prices from different websites and it will also auto correct the product spelling mistake. This will make shopping easy with best deals as well as ecommerce merchants to know well about their competitors.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	4
	LIST OF FIGURES	6
1.	INTRODUCTION	7
	1.1 Overview	7
	1.2 Problem Definition	7
2.	SYSTEM ANALYSIS	7
	2.1 Existing System	7
	2.2 Proposed System	8
	2.3 Development Environment	8
3	SYSTEM DESIGN	8
	3.1. UML Diagrams	8
	3.2 ER Diagram	11
	3.3 Data Flow Diagram	11
4.	SYSTEM ARCHITECTURE	12
	4.1 Architecture Overview	12
	4.2 Module Description	12
	4.3 Pseudo code / Algorithm Description	13
5.	SYSTEM IMPLEMENTATION	13
	5.1 Implementation Description	13
	5.2 Coding	13
6.	SYSTEM TESTING	17
	Test Cases & Reports	18
7.	CONCLUSION	19
	7.1 Sample Screenshots	19
	7.2 Conclusion and Future Enhancements	21
	REFERENCES	22

LIST OF FIGURES

FIG NO.	FIGURE DESCRIPTION	PAGE NO.
3.1	Use case diagram	8
3.2	Class diagram	9
3.3	Sequence diagram	9
3.4	Activity diagram	10
3.5	Communication diagram	10
3.6	ER diagram	11
3.7	Data flow diagram	11
4.1	Architecture diagram	12

INTRODUCTION

For someone who wants to save money, and receive the best value for money, they may meditate on comparison shopping. Though it sounds deadening, it can actually be challenging and rewarding. Analysing brands to find out the best value for the price will be helpful to everyone. It's not just to get the best products for their family, but will also help to save money long term. Even if they don't exactly exercise frugal living in all areas of their life, they can still practice comparison shopping to get the most money when certain purchases have been made.

1.1 OVERVIEW

Price Comparison websites are also known as "Aggregators" because they aggregate information on a product from many websites offering online shopping of the product. These sites render you easy and lucid comparisons of the prices on offer of a product under consideration, without any toiling involved, on a single screen in a matter of a few clicks of the mouse! These sites manage to save a substantial amount of effort, time and money of the prospective shopper, and above all, take away the headache of compiling information originally devised for computers and electronics in mind, the price comparison sites have caught on with almost all products and services on sale on web. The sites have rendered online shopping much more easily and quicker than it was earlier. Further, the price comparison sites display ratings against each site name so as to give the online customer a fair idea about the reliability and credibility of the site.

1.2 PROBLEM DEFINITION

The modern Internet savvy users prefer the Internet to make purchases rather than plod across offline stores. It is simply a matter of convenience and a feasible alternative to the utter lack of time faced by them. Given that these users prefer home delivery and instant service, going through web research is quite an ordeal indeed. As an age-old saying goes, 'Necessity is the mother of invention'. The Internet offers an overwhelming answer to all this turmoil of an online shopper in the form of "Price Comparison Sites". Price Comparison sites save online shoppers from undertaking a hectic web research, and basically cater the eventual product of the entire procedure of web research, otherwise undertaken by an avid web shopper before making a buying decision.

2 SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system needs manual comparison of price and other details of a product available online. In India, there are two big competitors in online shopping namely Amazon and Flipkart. Both sites provide customers with a huge variety of products among which people buy certain products. Psychology experts suggest choosing between a large number of products online would be physically exhausting and causes frustration. Also 50% of potential sales are lost because site visitors can't find what they are looking for.

2.2 PROPOSED SYSTEM

Web application development which makes it possible to compare products and find deals that are built using simple and intuitive UI/UX design. Product graphs will be built for selective categories.

Provides various categories and sorts with algorithms. Product redirect securely to the parent online shopping site on buy product. Web applications are hosted and deployed on a cloud application platform.

2.3 DEVELOPMENT ENVIRONMENT

The Web application is developed using HTML5, CSS, JavaScript, its libraries and tools. The Online shopping site's data will be accessed using product listing APIs, Custom APIs, web scraping and hard coding. The design system used is Adobe XD to build the interactive and intuitive UI/UX.

3 SYSTEM DESIGN

3.1 UML DIAGRAMS

★ USE CASE DIAGRAM

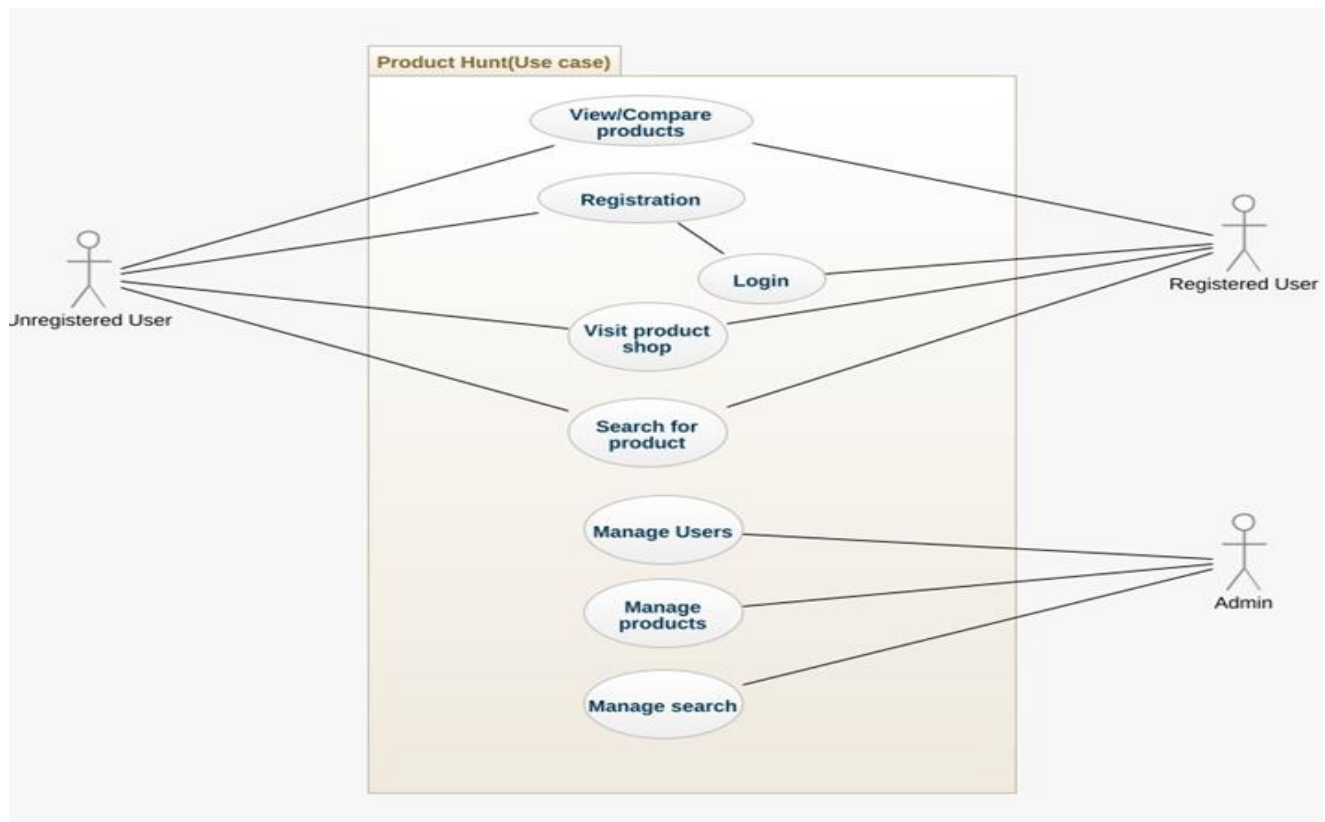


Fig 3.1

★ CLASS DIAGRAM

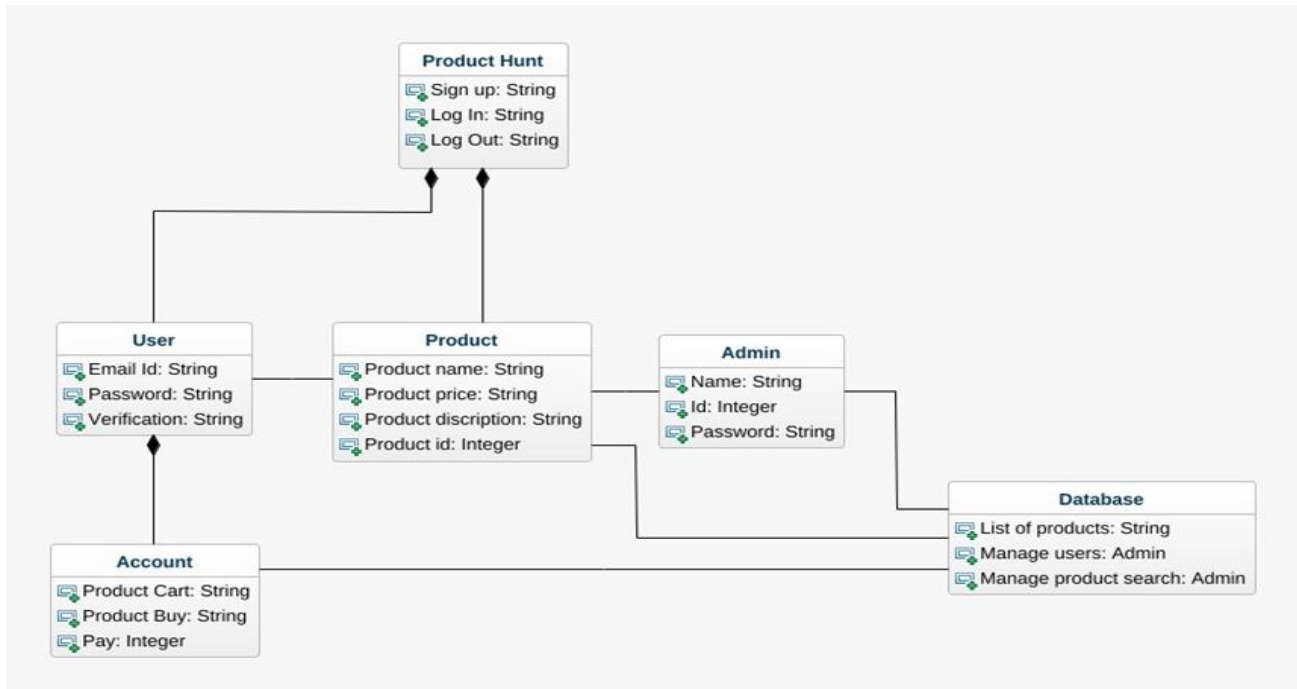


Fig 3.2

★ SEQUENCE DIAGRAM

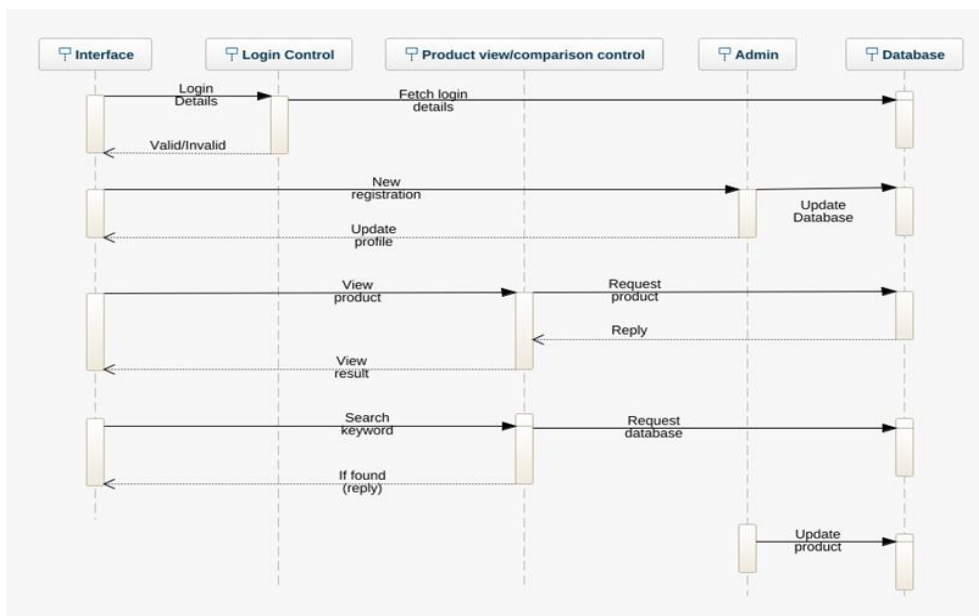


Fig 3.3

★ ACTIVITY DIAGRAM

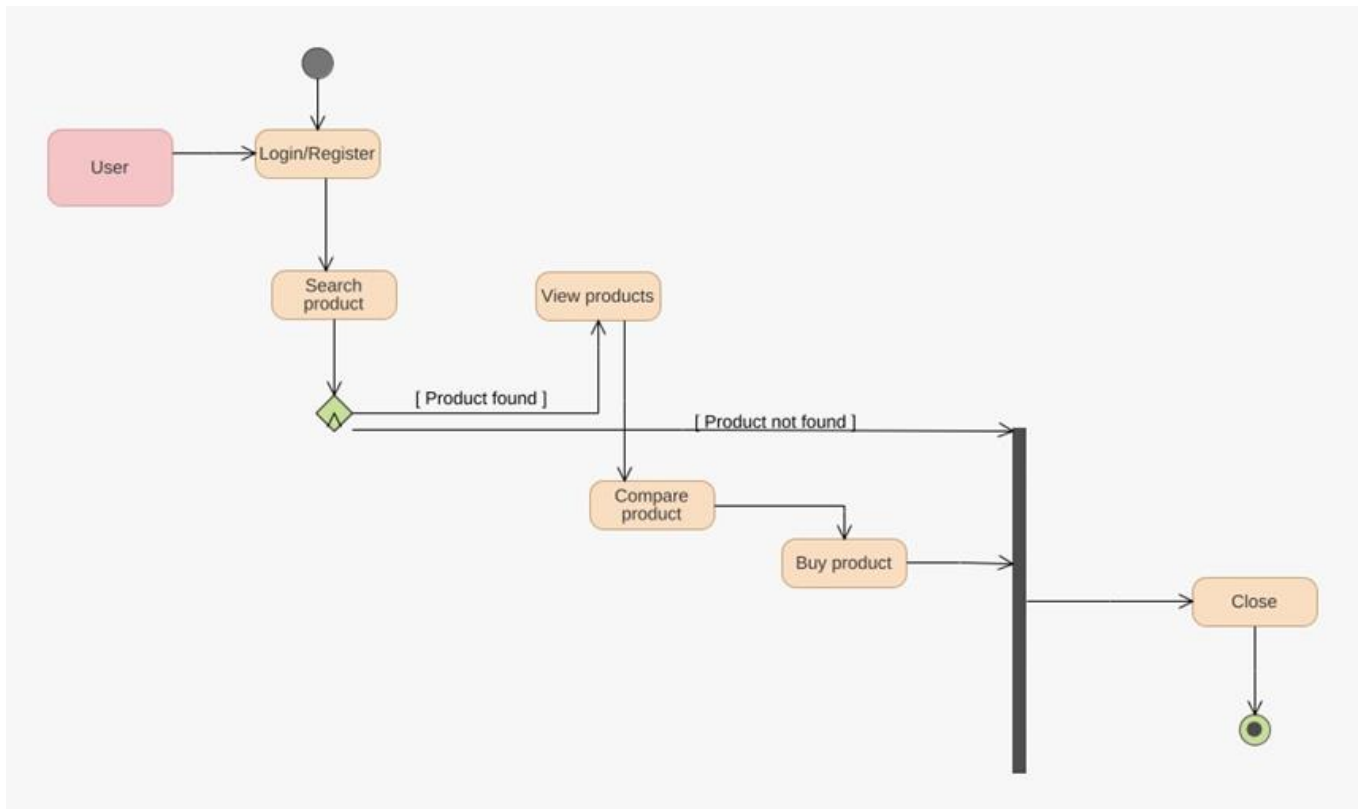


Fig 3.4

★ COMMUNICATION DIAGRAM

Communication Diagram: Product Hunt

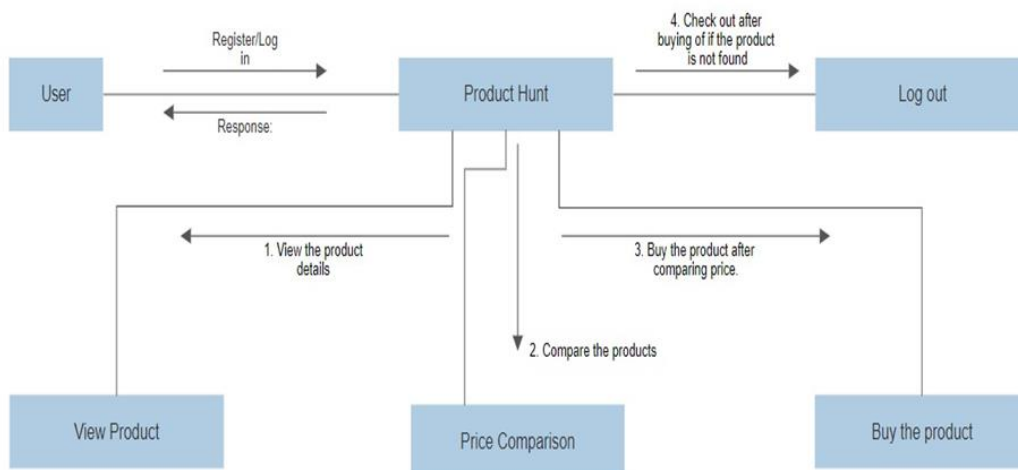


Fig 3.5

3.2 ER DIAGRAM

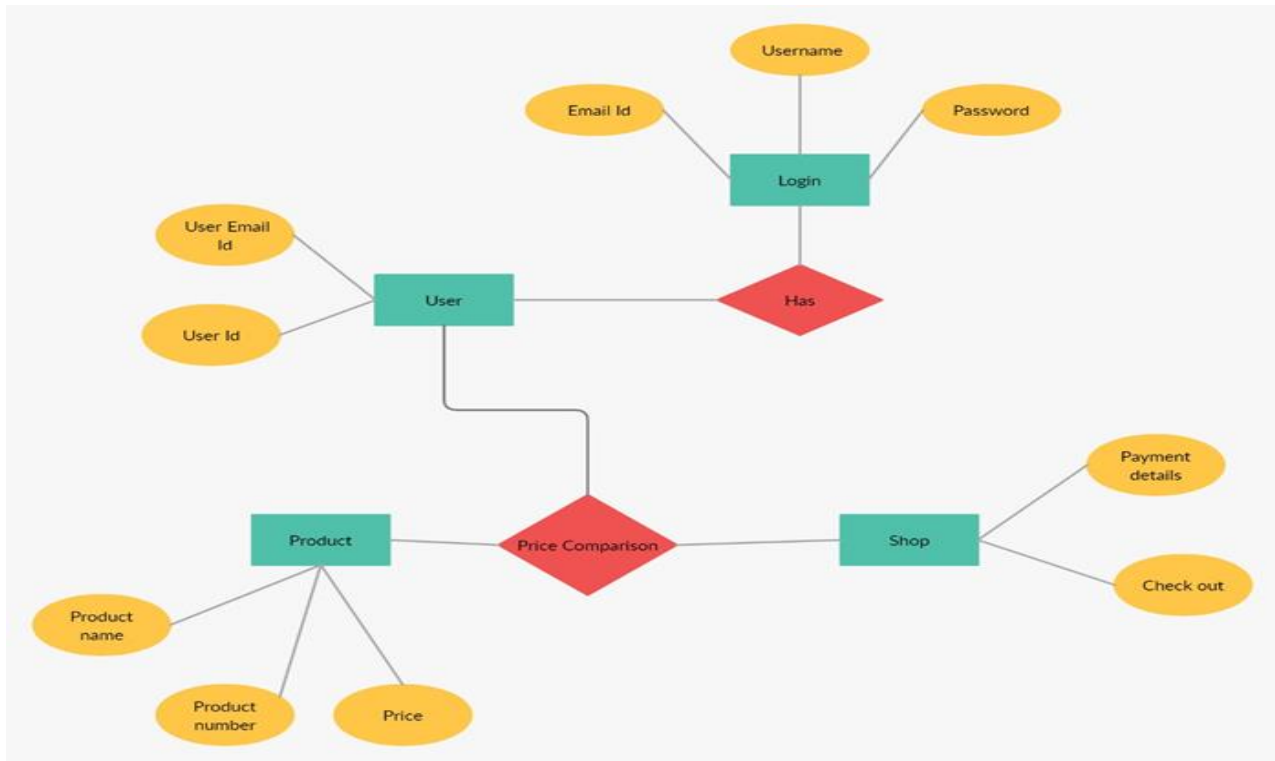


Fig 3.6

3.3 DATA FLOW DIAGRAM

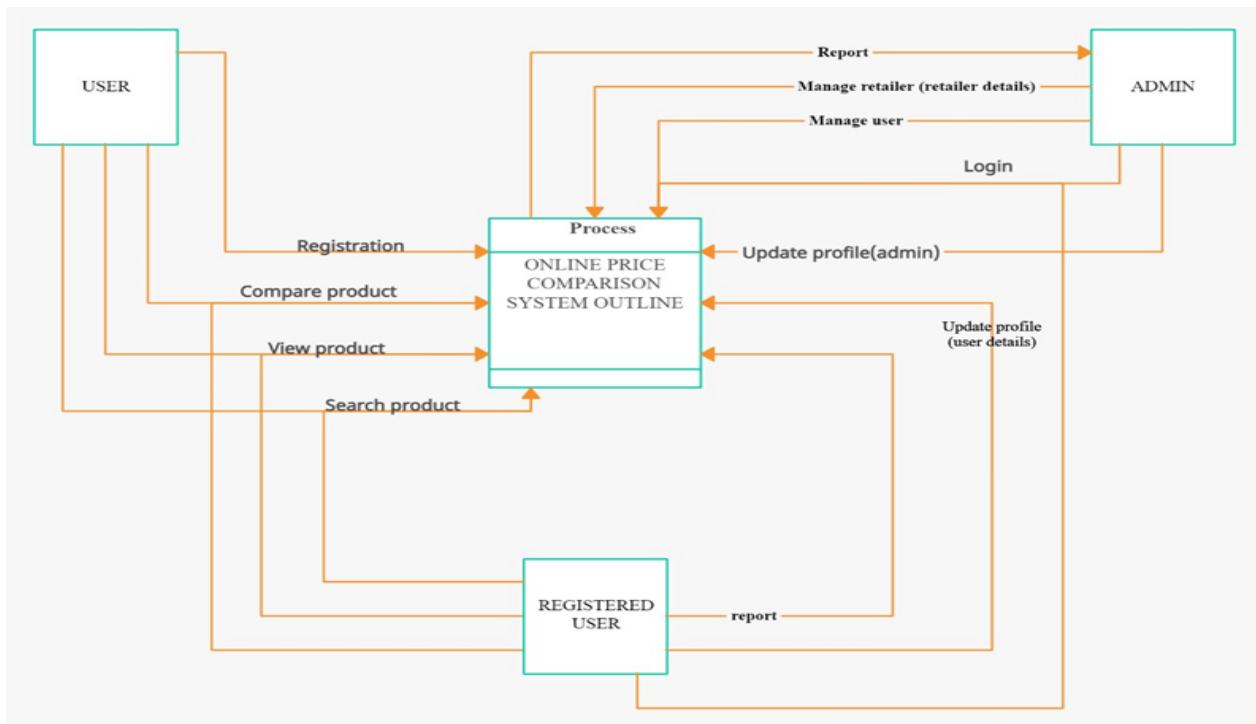


Fig 3.7

4. SYSTEM ARCHITECTURE

4.1 Architecture Overview

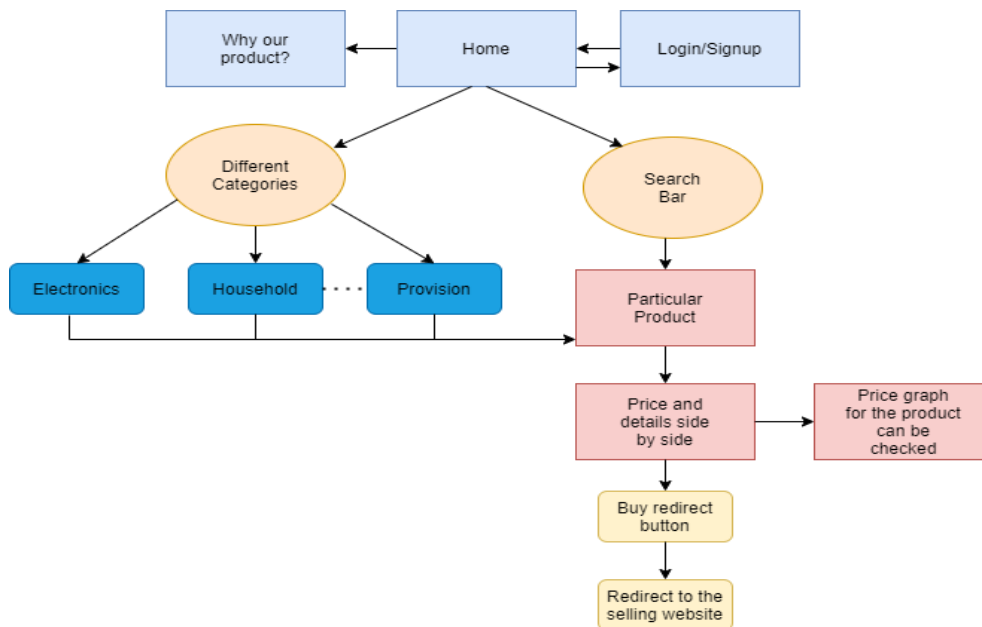


Fig 4.1

4.2 Module Description

Product Hunt has several interactive modules in the administration panel that are required for a dynamic and unique price comparison system.

Data Management

- Data APIs: Dynamically obtain all product and its details(Flipkart API, Custom Sheety API, Amazon API)
- Products : Manage data related to products.
- Web Scrapping: Extraction of data for Amazon and Flipkart.

User Management

Login: Manages all the login and signup process.
Users: Manage all the user details and add admin users.

Reports

Users: View users report.
Sales: View the sales report.
Products: View the product details report.

4.3 Pseudo code / Algorithm Description

- The system design is mainly focused on the web structure of product comparison.
- Login, Signup & Logout page for authentication through Google Authentication.
- Product catalog list using Amazon/Flipkart/Custom API calls
- The search and sort mechanism for better handling of products.
- Redirect to the corresponding Amazon and Flipkart Product page.

5. SYSTEM IMPLEMENTATION

5.2 Implementation Description

The project implementation was done using the design system and principles for the layouts using the Adobe XD. The technologies of HTML, CSS, JavaScript were used to create the different modules of the project and api integration using the RESTFUL JSON API of product list, custom APIs. The login and signup page were created for authentication and products categories and listing integration using the API calls and different types of system functionalities were executed for the web application and ready to host in the cloud server.

5.2 Coding (sample)

Project Repository: <https://github.com/DarinJoshua-dev/Productz-Hunt>

```
<!DOCTYPE html>
<html>
<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>signup</title>
<link rel="stylesheet" type="text/css" id="applicationStylesheet"
href="styles/signup.css"/>
<script id="applicationScript" type="text/javascript"
src="scripts/signup.js"></script>
</head>
<body>
<div id="signup">
  <div id="Group_1430">
    <div id="Social_Button" class="Social_Button">
      <div id="Button_Background">
        <svg class="Button_Background_background">
          <rect id="Button_Background_background" rx="0"
ry="0" x="0" y="0" width="427" height="63">
            </rect>
          </svg>
          <svg class="Button_Shape" viewBox="0 0 426.563 62.615">
            </svg>
```

```

</div>
<div id="_Sign_in_with_Apple">
  <span>Sign up with Google</span>
</div>
<div id="Group_1429">
  <svg class="Rectangle_1478">
    <rect id="Rectangle_1478" rx="0" ry="0" x="0" y="0"
width="31" height="31">
    </rect>
  </svg>
  <div id="logo_googleg_48dp">
    <svg class="Shape" viewBox="0 0 11.271 11.031">
    </svg>
    <svg class="Shape_cun" viewBox="0 0 18.261 9.51">
    </svg>
    <svg class="Shape_cuo" viewBox="0 0 5.171 10.545">
    </svg>
    <svg class="Shape_cup" viewBox="0 0 18.347 9.51">
    </svg>
    <svg class="Shape_cuq" viewBox="0 0 23.481 23.481">
      <path id="Shape_cuq" d="M 0 0 L
23.48050308227539 0 L 23.48050308227539 23.48050308227539 L 0
23.48050308227539 L 0 0 Z">
      </path>
    </svg>
  </div>
</div>
</div>
<div id="Social_Button_cur" class="Social_Button">
  <div id="Button_Background_cus">
    <svg class="Button_Background_background_cut">
      <rect id="Button_Background_background_cut" rx="0"
ry="0" x="0" y="0" width="427" height="63">
      </rect>
    </svg>
    <svg class="Button_Shape_cuu">
      <rect id="Button_Shape_cuu" rx="6" ry="6" x="0"
y="0" width="427" height="63">
      </rect>
    </svg>
  </div>
  <div id="_Sign_in_with_Apple_cuv">
    <span>Sign up with Facebook</span>
  </div>
  <div id="Group_1428">
    <svg class="Rectangle_1478_cux">
      <rect id="Rectangle_1478_cux" rx="0" ry="0" x="0"
y="0" width="31" height="31">
      </rect>
    </svg>
    <svg class="facebook-square" viewBox="0 32 23.942
23.942">
    </svg>
  </div>

```

```

</div>
<div id="Form_Divider" class="Form_Divider">
  <svg class="Line_192" viewBox="0 0 426.563 1">
    <path id="Line_192" d="M 0 0 L 426.5625 0">
    </path>
  </svg>
  <div id="Group_1423">
    <svg class="Path_1077" viewBox="0 0 44 30">
      <path id="Path_1077" d="M 0 0 L 44 0 L 44
30.00000190734863 L 0 30.00000190734863 L 0 0 Z">
      </path>
    </svg>
    <div id="Or">
      <span>Or</span>
    </div>
  </div>
</div>
<div id="Underline_Input" class="Underline_Input">
  <svg class="Rectangle_1607">
    <rect id="Rectangle_1607" rx="0" ry="0" x="0" y="0"
width="427" height="72">
    </rect>
  </svg>
  <svg class="Line_184" viewBox="0 0 427 1">
    <path id="Line_184" d="M 0 0 L 427 0">
    </path>
  </svg>
  <div id="Password">
    <span>Password</span>
  </div>
</div>
<div id="Underline_Input_cu" class="Underline_Input">
  <svg class="Rectangle_1607_cu">
    <rect id="Rectangle_1607_cu" rx="0" ry="0" x="0" y="0"
width="427" height="72">
    </rect>
  </svg>
  <svg class="Line_184_cva" viewBox="0 0 427 1">
    <path id="Line_184_cva" d="M 0 0 L 427 0">
    </path>
  </svg>
  <div id="Email">
    <span>Email</span>
  </div>
</div>
<div id="Underline_Input_cvc" class="Underline_Input">
  <svg class="Rectangle_1607_cvd">
    <rect id="Rectangle_1607_cvd" rx="0" ry="0" x="0" y="0"
width="427" height="72">
    </rect>
  </svg>
  <svg class="Line_184_cve" viewBox="0 0 427 1">
    <path id="Line_184_cve" d="M 0 0 L 427 0">
    </path>

```

```

        </svg>
        <div id="Full_Name">
            <span>Full Name</span>
        </div>
    </div>
    <div onclick="application.goToTargetView(event)"
id="Already_have_an_account_Log_I">
        <span>Already have an account? </span><br><span style="font-
style:normal;font-weight:bold;color:rgba(68,120,236,1);">Log In</span>
    </div>
    <div id="Default_Button">
        <svg class="Rectangle_1469_cvj">
            <linearGradient id="Rectangle_1469_cvj"
spreadMethod="pad" x1="0" x2="1" y1="0.156" y2="0.868">
                <stop offset="0" stop-color="#0068c7" stop-
opacity="1"></stop>
                <stop offset="0.4684" stop-color="#6780ff" stop-
opacity="1"></stop>
                <stop offset="1" stop-color="#d29bb6" stop-
opacity="1"></stop>
            </linearGradient>
            <rect onclick="application.goToTargetView(event)"
id="Rectangle_1469_cvj" rx="8" ry="8" x="0" y="0" width="427"
height="62">
                </rect>
            </svg>
            <div id="Join_us">
                <span>Join us</span>
            </div>
        </div>
    </div>
    <div onclick="application.goToTargetView(event)" id="Group_15_cvl">
        <svg class="Union_1_cvm" viewBox="0 0 116.025 178.8">
            </svg>
            
        </div>
        
        <div onclick="application.goToTargetView(event)"
id="Or_continue_without_an_account">
            <span>Or continue without an account</span><br>
        </div>
    </div>
</body>
</html>

```


6. SYSTEM TESTING

Testing refers to testing software so it is also called software testing. Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- is sufficiently usable,
- can be installed and run in its intended environments, and
- achieves the general result its stakeholders desire.

Unit Testing:

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Parameterized unit tests (PUTs) are tests that take parameters. Unlike traditional unit tests, which are usually closed methods, PUTs take any set of parameters. PUTs have been supported by Testing, JUnit and various .NET test frameworks. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

Integration Testing:

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

System Testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

Acceptance Testing:

In engineering and its various sub disciplines, acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests.

In systems engineering it may involve black-box testing performed on a system (for example: a piece of software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery.

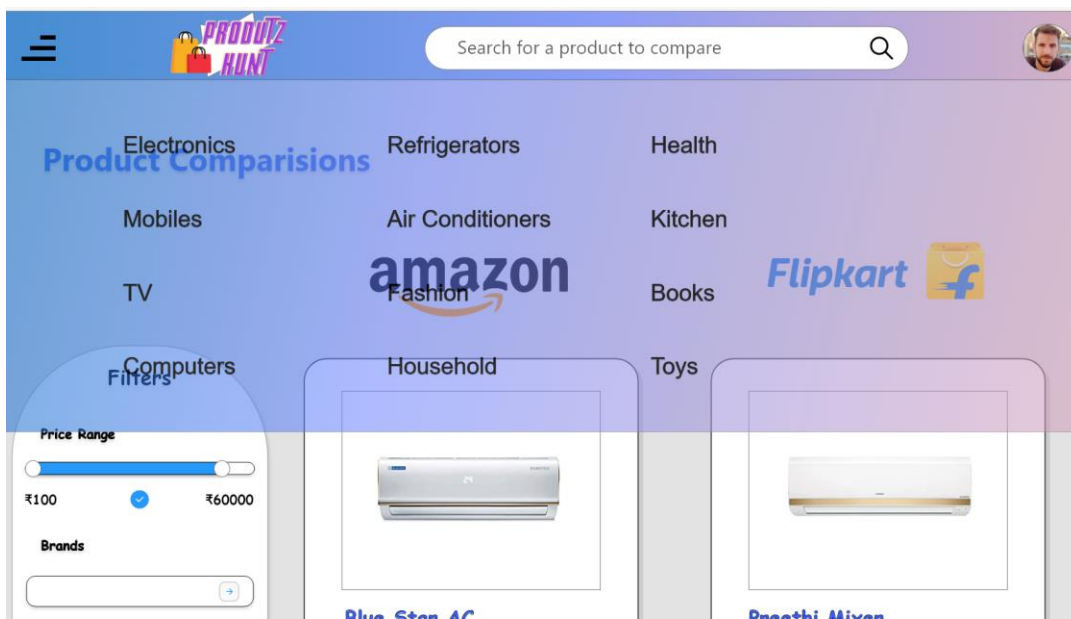
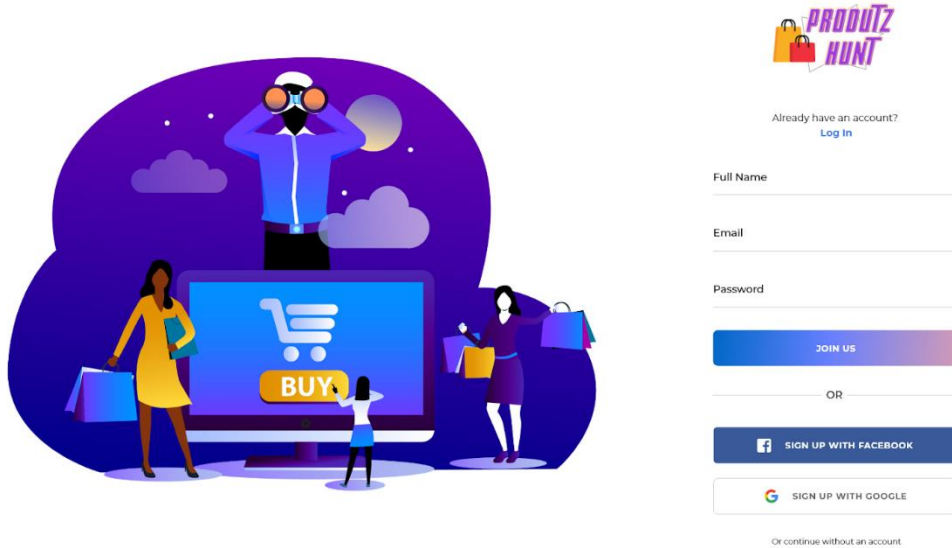
Test Cases & Reports:



Test Case ID	Description	Test steps	Test data	Expected Result	Actual Result	Test pass/fail
1	Login page	1. Enter email 2. Enter password	Darin D@rin	Login successful	Login successful	Pass
2	Selecting a product	Click on the product	N/A	Show comparison	Show comparison	Pass
3	Navbar	1. Click on navbar 2. Select a category	N/A	Show products under that category	Show products under that category	Pass

4	Why our product	Click why our product	N/A	Show why our product page	Show why our product page	Pass
5	Logo	Click on logo	N/A	Show why our product page	Show why our product page	Pass


7. CONCLUSION

7.2 Sample Screenshots (Project Working Prototype)





Search for a product to compare



Product Comparisons



Filters

Price Range

₹100 ✓ ₹60000



Brands




Boult Audio BassBuds



Preethi Mixer



Search for a product to compare



Product Comparisons



Filters

Price Range

₹100 ✓ ₹60000

Brands



Blue Star AC



Preethi Mixer



7.2 Conclusion and Future Enhancements

Currently the system is designed to compare the desired products of Amazon and Flipkart with an intuitive UI. In future, we would be able to add other popular companies in the **Productz Hunt** after modifying the existing code and algorithms. Also we would be adding more products, feature integrations and cloud server facilities, with more data and API sets.

With proper feedback and critics, we would be able to fix the issues and make it useful for every user and the business requirements can be fully satisfied. There will also be continuous maintenance and upgrades to give add-ons and bug fixes to the feature sets.

In future, We are planning to add various improvements, product API integrations and best practices will be made possible using various user data and will be unified using AI and a control system to give advanced suggestions and referrals.

REFERENCES

<https://snipcart.com/blog/integrating-apis-introduction>

<https://www.budibase.com/blog/how-to-make-a-web-app/>

<https://www.agconsult.com/en/usability-blog/4-product-comparison-best-practices/>

<https://www.feelingpeaky.com/9-principles-of-good-web-design/>

<https://realpython.com/beautiful-soup-web-scraper-python>

<https://www.w3schools.com/js/>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<https://www.w3schools.com/css/default.asp>

<https://www.ibm.com/cloud/learn/api>

<https://martinfowler.com/articles/serverless.html>

<https://css-tricks.com/>