

# **INDUSTRY INTERNSHIP SUMMARY REPORT**

**Full Stack Development**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

*Submitted by*

**Abhishek Jha (23SCSE1010836)**

**Vth Sem III Year**



**GALGOTIAS  
UNIVERSITY**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING  
GREATER NOIDA, UTTAR PRADESH  
2024 – 2025**



C-56/11, Ground floor, Near IT Stellar Park, Sector-62, Noida, U.P. - 201309  
Email: [support@upskillcampus.com](mailto:support@upskillcampus.com), Ph. No. 9354068856

## Internship Offer Letter

Date: **09-11-2025**

Name: **Abhishek Jha**

Email: **abhishekj4611989@gmail.com**

Location: **Remote**

Dear **Abhishek Jha**,

We are pleased to inform you that **Upskill Campus** alongwith its industry Partner **UniConverge Technologies Pvt. Ltd.** has agreed to provide you internship as **Full Stack Development Intern**. Your appointment of service is based on the following terms and conditions.

1. The internship shall start from **09-11-2025**
2. **Internship:** Internship period will be for 4-6 weeks starting from date of joining.
3. **Working Hours:** The internship runs from 10:00 AM to 6:00 PM.

On behalf of both the Company, we wish you every success in your position and trust that our relationship will be long and mutually rewarding.

Sincerely

**For Upskill Campus and UniConverge Technologies Pvt. Ltd.**



**Prabha Singh**  
**HR Manager**

## **Table of Content**

### **1. Chapter 1- Introduction**

Company Profile

Vision

History

Customers

### **2. Chapter 2- Introduction to Assigned Job**

### **3. Chapter 3- Modular Description of Job**

Introduction to Web Development

HTML

CSS

Javascript

Google Chrome Developer Console

React Js

Node Js

### **4. Chapter 4- Analysis of the tasks performed**

### **5. Chapter 5- Conclusion**

### **6. Chapter 6- Bibliography**

## **List of Figures**

- Fig 3.1- Skillset of a full stack web developer.
- Fig 3.2- HTML symbol.
- Fig 3.3- CSS symbol.
- Fig 3.4- Javascript Image.
- Fig 3.5- Elements tab of Developer Tools.
- Fig 3.6- Console tab to run Javascript.
- Fig 3.7- Sources tab to list source files.
- Fig 3.8- Network tab to show network activity.
- Fig 3.9- Performance tab.
- Fig 3.10- Memory Tab.
- Fig 3.11- Application Tab.
- Fig 3.12- Security Tab.
- Fig 3.13- Audit Tab.
- Fig 3.14- React Js Symbol.
- Fig 3.15- Node Js Symbol.
- Fig 4.1.1- Dialog Box to verify a transaction.
- Fig 4.2.1- Global Filters
- Fig 4.2.2- Growth Chart
- Fig 4.2.3- Goal Tracker
- Fig 4.2.4- Sales Chart
- Fig 4.2.5- Gaining and Losing Customers
- Fig 4.2.6- Repeat Frequency Chart
- Fig 4.2.7- Customer Preference
- Fig 4.2.8 – Customer Profile
- Fig 4.2.9- Time/Week Pattern Graph

- Fig 4.2.10- Average Order Value Chart
- Fig 4.2.11- Potential Chart Form
- Fig 4.2.12- Potential Chart Outcome
- Fig 4.2.13- Feed On Vouchers Page
- Fig 4.3.1- Sales Chart with Values Tab Selected
- Fig 4.3.2- Sales Chart with Orders Tab selected
- Fig 4.3.3- Vouchers Table and Conversion Funnel
- Fig 4.3.4 – Customer Preference Chart
- Fig 4.3.5- Customer Profile and City Distribution Chart

# **Chapter - 1**

## **INTRODUCTION**

### **CompanyProfile**

Magicpin is the platform where users and merchants in a locality discover, interact, and transact. For merchants, they are the go-to platform for spotlighting their uniqueness while boosting their revenues and utilization through real-time promotions, loyalty programs, and content updates. For users, they are the destination to find out the buzz in their locality and discover interesting people to connect with and events/merchants to go to. Magicpin's headquarters is in Gurgaon, Haryana. Magicpin's main competitors are Crownit, CouponDunia and Nearbuy.

### **Vision**

We are on a mission to make hyperlocal magical. Pin a location on the map to find the coolest people in the neighborhood and where they hangout - from the hippest cafes to the yummiest dimsums, the trendiest fashion boutique, luxurious spas, health clubs, best discounts on grocery stores and more. Find the buzz in your locality through our user generated pictures, videos, reviews to discover interesting people, places and get cashback / rewards for visiting those places. magicpin drives discovery that leads to business for local retailers across categories like restaurant, fashion, beauty, grocery, spa, yoga, gyms, and more. For their spending, users get rewarded in 'magicpin points' that can be used to buy more services and experiences.

### **History**

The company was founded in **2015** by Anshoo, former Venture Partner at Lightspeed India Partners, and Brij Bhushan, former Nexus Venture Partners, who were colleagues at Bain and Co. In 2017, Magicpin has raised \$7 million in Series B round from Lightspeed India Partners, Waterbridge Ventures, and two global family offices.

### **Customers**

Magicpin is by far the leader in offline marketing for retailers in India. They have grown 5x in 2018 and are on track to repeat that in the coming year. This is a massive \$1T market in India - 50x the size of all ecommerce in the country. magicpin has 5 Million+ users and is now live in Delhi, Gurgaon, Noida, Bangalore, Mumbai, Pune, Hyderabad, Chandigarh, Jaipur, Goa, Chennai, Ahmedabad and is quickly spreading to more cities and countries.

## **CHAPTER 2**

### **INTRODUCTION TO ASSIGNED JOB**

During the internship, various techniques were taught which are as follows:

Introduction to Web Development

HTML

CSS

JAVASCRIPT

GOOGLE CHROME DEVELOPER CONSOLE

REACT JS

NODE JS

## **CHAPTER 3**

### **MODULAR DESCRIPTION OF JOB**

#### **Introduction to Web Development**

**Goal-** To understand the basics of how the web technology works, both at the server and client-side, and how everything combines front-end to the back-end to give a complete web-app.

#### **Explanation-**

Full Stack Web Development alludes to the development of both front end(client side) and back end(server side) portion of web application.

Full stack web engineers can configuration complete web application and sites. They take a shot at the frontend, backend, database and maintenance of web application or sites.

Front end is the noticeable piece of site or web application which is liable for client experience. The client straightforwardly interfaces with the front end segment of the web application or site.

Back end alludes to the server-side portion of web application or site with an essential spotlight on how the site functions. It is answerable for dealing with the database through questions and APIs by customer side orders. This sort of site primarily comprises of three sections front end, back end, and database.

Database is the assortment of between related information which helps in productive recovery, addition and erasure of information from database and sorts out the information as tables, sees, patterns, reports and so on.

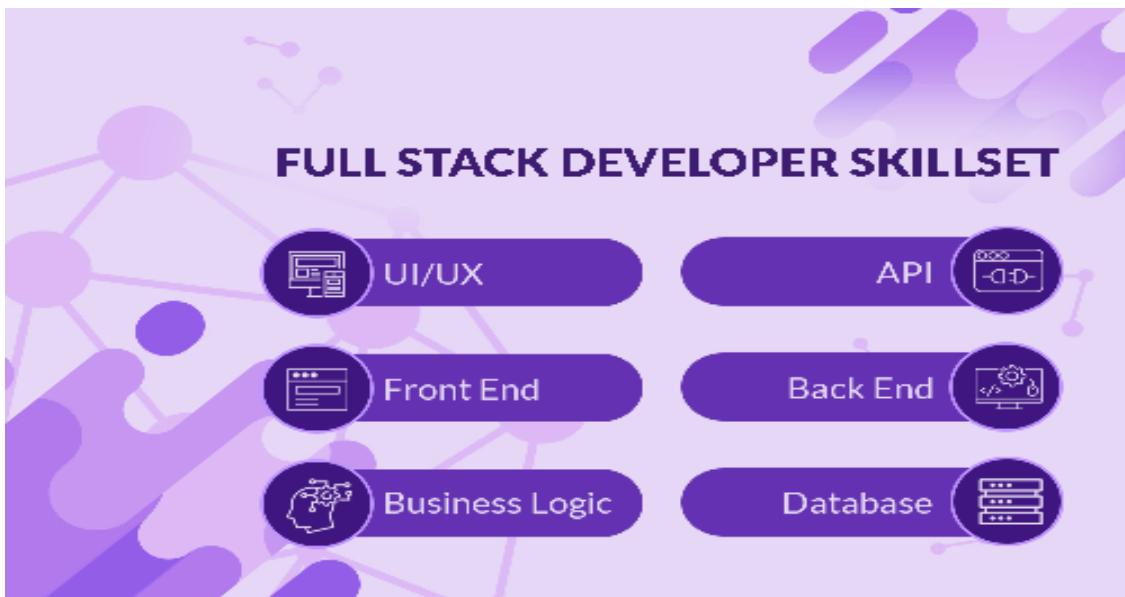


Fig 3.1- Skillset of a full stack web developer

## **HTML (Hypertext Markup Language)**

**Goal-** To utilize prior knowledge of HTML Fundamentals and become practically familiar with the most usable tags such as div, span, strong, forms, img, hyperlinks etc.

### **Explanation –**

Hypertext Markup Language (HTML) is the standard markup language for archives intended to be shown in an internet browser. It very well may be helped by advancements, for example, Cascading Style Sheets (CSS) and scripting dialects, for example, JavaScript.

Internet browsers get HTML reports from a web server or from nearby capacity and render the records into mixed media site pages. HTML portrays the structure of a site page semantically and initially included signals for the presence of the archive.

HTML components are the structure squares of HTML pages. With HTML builds, pictures and different items, for example, intelligent structures might be implanted into the rendered page. HTML gives a way to make organized archives by indicating basic semantics for content, for example, headings, passages, records, connections, cites and different things. HTML components are depicted by labels, composed utilizing point sections. Labels, for example, <img/> and <input/> straightforwardly bring content into the page. Different labels, for example, <p> encompass and give data about archive message and may incorporate different labels as sub-components. Programs don't show the HTML labels, yet use them to decipher the substance of the page.

# **HTML**



Fig 3.2 – HTML symbol

## **CSS (Cascading Style Sheets)**

**Goal**-To get familiar with basics of styling a webpage using CSS and minifying CSS for more efficiency.

### **Explanation –**

Cascading Style Sheets (CSS) is a template language utilized for portraying the introduction of a record written in a markup language like HTML. CSS is a foundation innovation of the World Wide Web, close by HTML and JavaScript .

CSS is intended to empower the partition of introduction and substance, including design, hues, and fonts. This detachment can improve content availability, give greater adaptability and control in the detail of introduction qualities, empower different site pages to share arranging by determining the significant CSS in a different .css record, and lessen intricacy and redundancy in the auxiliary substance .

Partition of organizing and substance likewise makes it attainable to introduce a similar markup page in various styles for various rendering techniques, for example, on-screen, in print, by voice (by means of discourse based program or screen per user), and on Braille-based material gadgets. CSS additionally has rules for substitute designing if the substance is gotten to on a versatile device.

The name cascading originates from the predefined need plan to figure out which style rule applies if more than one principle coordinates a specific component. This falling need conspire is unsurprising.



Fig 3.3 – CSS symbol

## JAVASCRIPT

**Goal-** To understand the concepts of Javascript such as event-handling, OOPS etc. and use them to create interactive web-apps.

### **Explanation-**

JavaScript frequently shortened as JS, is a programming language that fits in with the ECMAScript determination. JavaScript is elevated level, regularly in the nick of time gathered, and multi-worldview. It has wavy section linguistic structure, dynamic composing, model based article direction, and top of the line capacities.

Nearby HTML and CSS, JavaScript is one of the center advances of the World Wide Web. JavaScript empowers intuitive site pages and is a

fundamental piece of web applications. Most by far of sites use it for customer side page conduct, and all significant internet browsers have a devoted JavaScript motor to execute it.

As a multi-worldview language, JavaScript underpins occasion driven, practical, and basic programming styles. It has application programming interfaces (APIs) for working with content, dates, ordinary articulations, standard information structures, and the Document Object Model (DOM). Notwithstanding, the language itself does exclude any info/yield (I/O, for example, systems administration, stockpiling, or illustrations offices, as the host condition (generally an internet browser) gives those APIs.

JavaScript motors were initially utilized distinctly in internet browsers, however they are presently implanted in certain servers, as a rule by means of Node.js. They are additionally installed in an assortment of uses made with systems, for example, Electron and Cordova.

In spite of the fact that there are similarities among JavaScript and Java, including language name, linguistic structure, and particular standard libraries, the two dialects are unmistakable and contrast enormously in plan.



Fig 3.4 – Javascript Image

## GOOGLE CHROME DEVELOPER CONSOLE

**Goal-** To understand how the data that flows through the app, can be checked in network tab, how console tab helps in logging and how element tab helps in DOM manipulation at runtime.

### Explanation-

Web improvement tools (regularly called devtools) permit web engineers to test and investigate their code. They are not the same as web designers and coordinated improvement situations (IDEs) in that they don't aid the immediate formation of a site page, rather they are instruments utilized for testing the UI of a site or web application.

Web improvement tools come as program additional items or inherent highlights in internet browsers. Most well known internet browsers, for example, Google Chrome, Firefox, Internet Explorer, Safari and Opera, have worked in apparatuses to help web designers, and numerous extra additional items can be found in their particular module download focuses.

Web advancement instruments permit designers to work with an assortment of web advances, including HTML, CSS, the DOM, JavaScript, and different segments that are taken care of by the internet browser. Because of expanding request from internet browsers to do more, famous internet browsers have included more highlights designed for engineers.

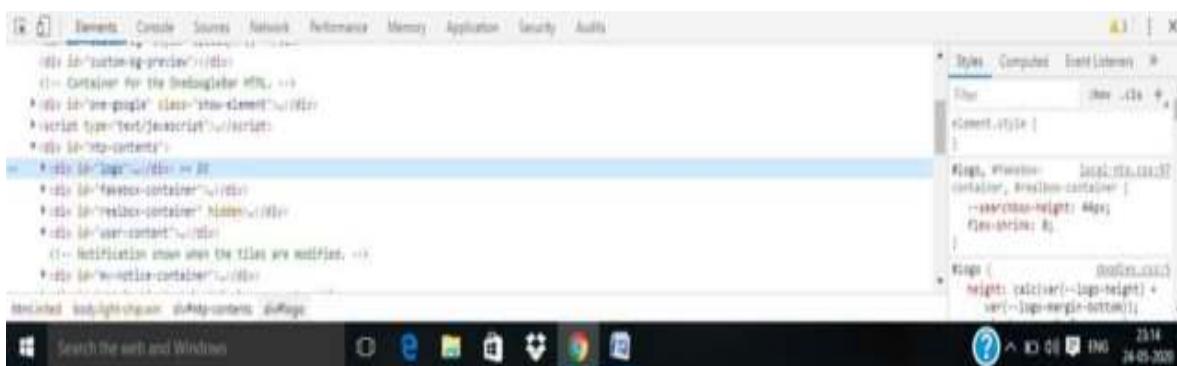


Fig 3.5 – Developer Tools Elements tab to view HTML and CSS.

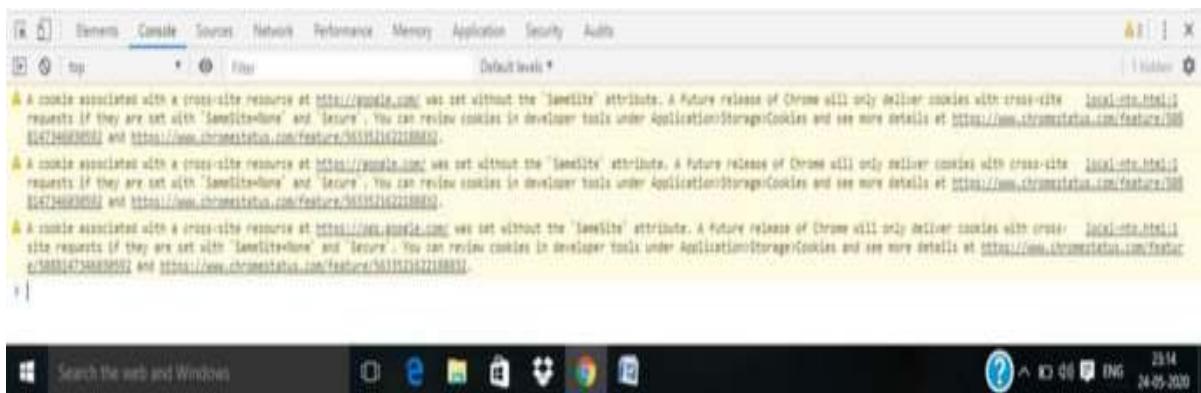


Fig 3.6 – Console tab to run Javascript

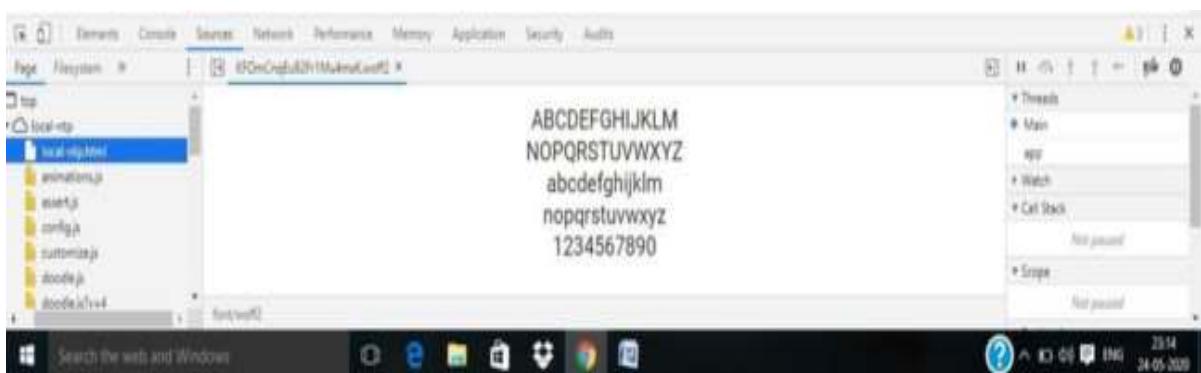


Fig 3.7 – Sources tab lists source files

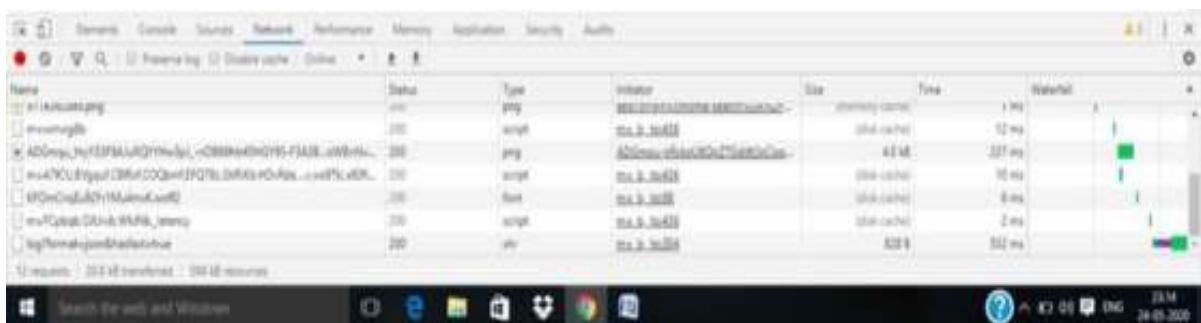


Fig 3.8 – Network tabs show the data fetched over network calls

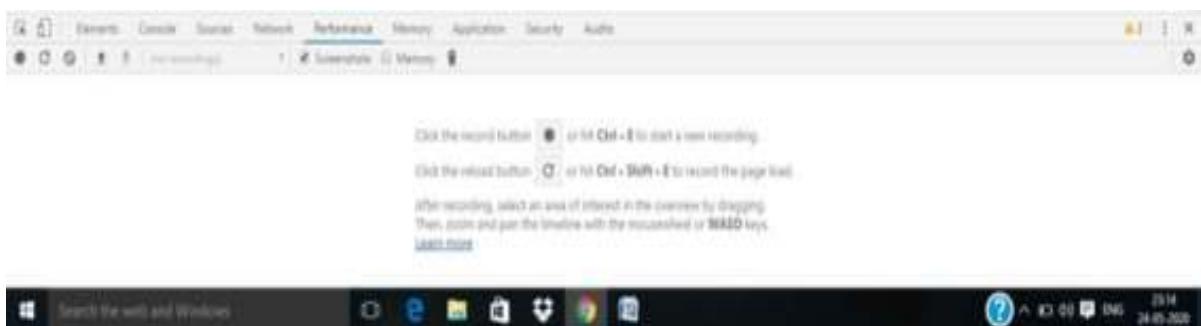


Fig 3.9 – Performance tab to show efficiency.

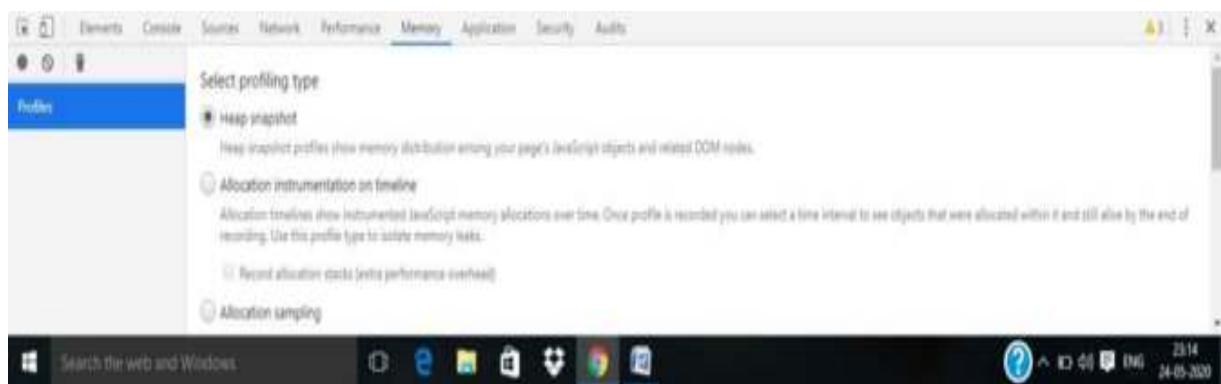


Fig 3.10 – Memory tab shows memory usage

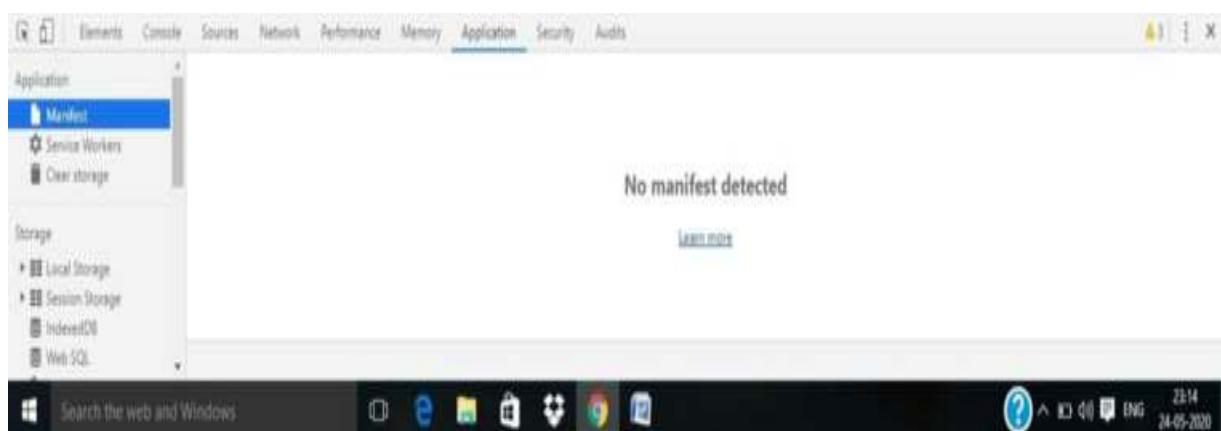


Fig 3.11 – Application tab show total application structure

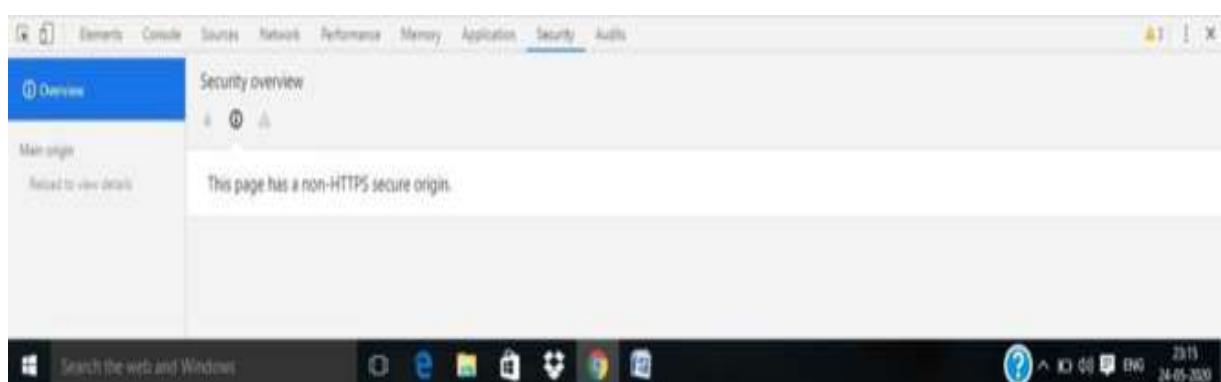


Fig 3.12 – Security tab reviews security

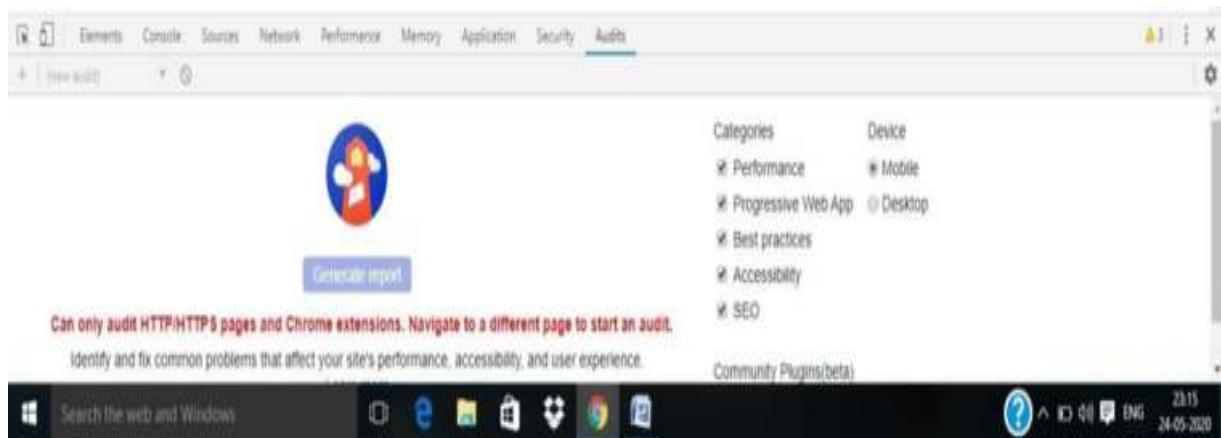


Fig 3.13 – Audit tab shows various miscellaneous features.

## REACT JS

**Goal-** To understand the architecture of front-end development using react framework developed by the Facebook.

### Explanation-

React (otherwise called React.js or ReactJS) is an open-source JavaScript library for building UIs. It is kept up by Facebook and a network of individual designers and organizations.

React can be utilized as a base in the advancement of single-page or portable applications. Be that as it may, React is just worried about rendering information to the DOM, thus making React applications for the most part requires the utilization of extra libraries for state the executives and directing. Redux and React Router are individual instances of such libraries.

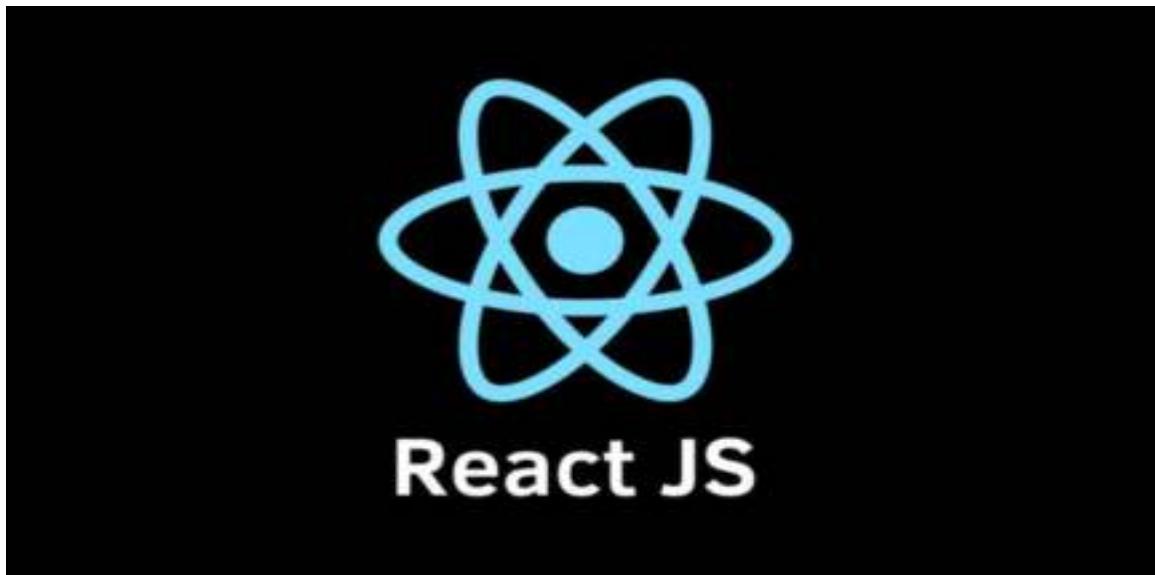


Fig 3.14 – React Symbol

## NODE JS

**Goal-** To understand the basics of backend development using node js and to host a server using express js.

### Explanation –

Node.js is an open-source, cross-stage, JavaScript runtime condition that executes JavaScript code outside of an internet browser. Node.js lets engineers use JavaScript to compose order line devices and for server-side scripting—running contents server-side to deliver dynamic website page content before the page is sent to the client's internet browser. Subsequently, Node.js speaks to a "JavaScript all over the place" worldview, bringing together web-application improvement around a solitary programming language, as opposed to various dialects for server-and customer side contents.

However .js is the standard filename expansion for JavaScript code, the name "Node.js" doesn't allude to a specific document in this unique circumstance and is only the name of the item. Node.js has an occasion driven design equipped for offbeat I/O. These plan decisions mean to upgrade throughput and adaptability in web applications with many info/yield tasks, just as for continuous Web applications (e.g., ongoing correspondence projects and program games).

The Node.js disseminated advancement venture was recently represented by the Node.js Foundation, and has now converged with the JS Foundation to frame the OpenJS Foundation, which is encouraged by the Linux Foundation's Collaborative Projects program



**Fig 3.15 – Node Js Symbol**

## CHAPTER 4

### ANALYSIS OF THE TASKS PERFORMED

#### FRONT-END

**1. Brand Panel:** In this panel, all the information related to various brands is shown. Data is displayed in the form of various graphs, representing the sales on the basis of various time periods such as on daily, weekly and monthly basis. Also, there are graphs representing the best city, locality, where the sales of that brand has occurred in the recent times. The competitor data is also shown in the form of various graphs.

Created a dialog box to be displayed, on verification of a transaction, and set buttons color using CSS. The screenshot shown below displays a webpage where the verification of billed transaction is done. An algorithm detects the items printed on the bill using machine learning and image processing techniques and then verification is performed on these listed items. If the verification is possible, the transaction is submitted, otherwise the transaction skipped and the dialog box gets displayed.

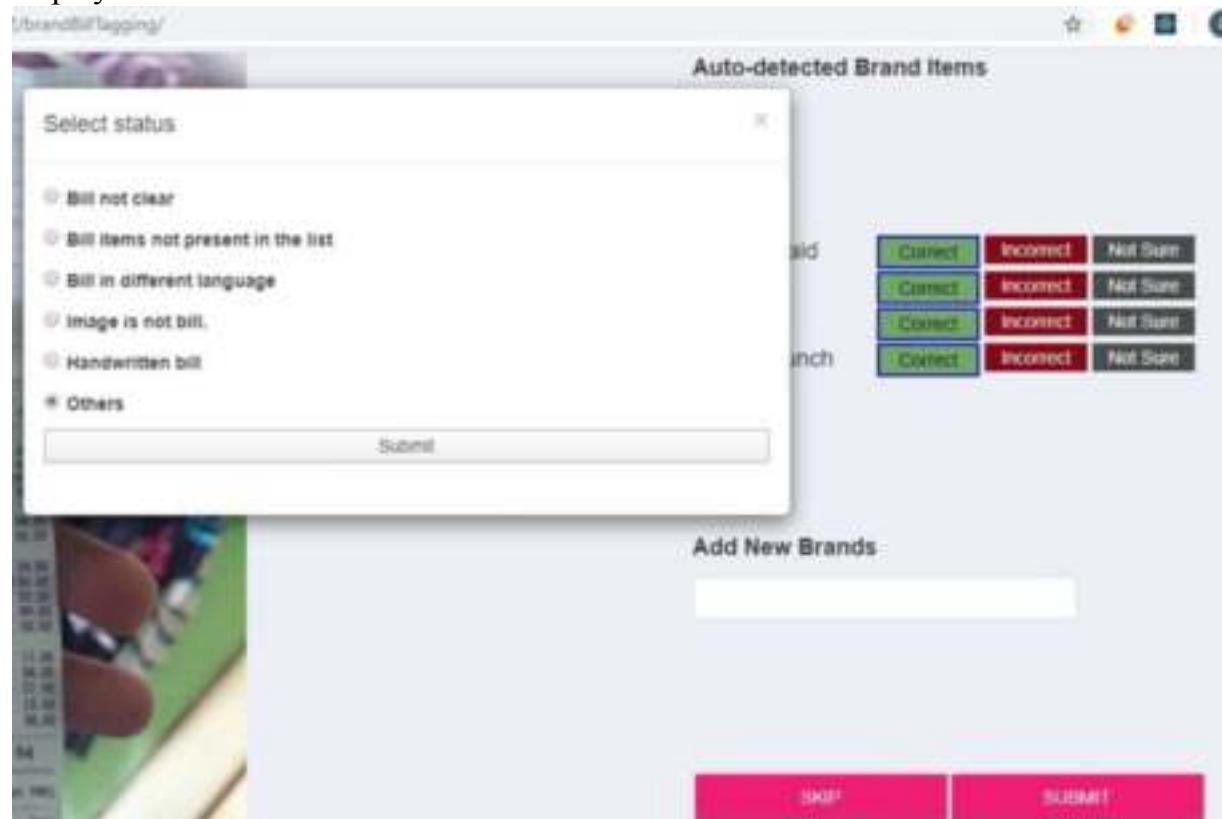


Fig 4.1.1 - Dialog Box to verify a transaction

**2. Retail Panel:** This panel lists all the information related to the retail merchants. These merchants are categorized as retail-merchants, online merchants etc. On retail panel, we have four various web pages namely, Home, Voucher, Customers, and Feedback. The home page displays the data related to the merchant campaigns. The voucher page shows the vouchers offered on the magicpin platform by the merchants, campaign wise. The customer page shows customers related data and feedback page presents user-feedback.

Created global filters to change the information displayed on the webpage as the dates are changed or campaign is changed. There are two global-filters, one that allows switching between the campaigns and another, that changes the dates for which the data is displayed. These dates can be set manually by the user, along with some default options.



Fig 4.2.1 – Global Filters

Created and wired the data for the Growth Graph, using react chart-js library. The Growth Graph displays percentage wise growth of a merchant in recent times. It's view can be changed on daily, monthly and weekly basis. This graph provides comparison data for a merchant for the various competitors in its categories.

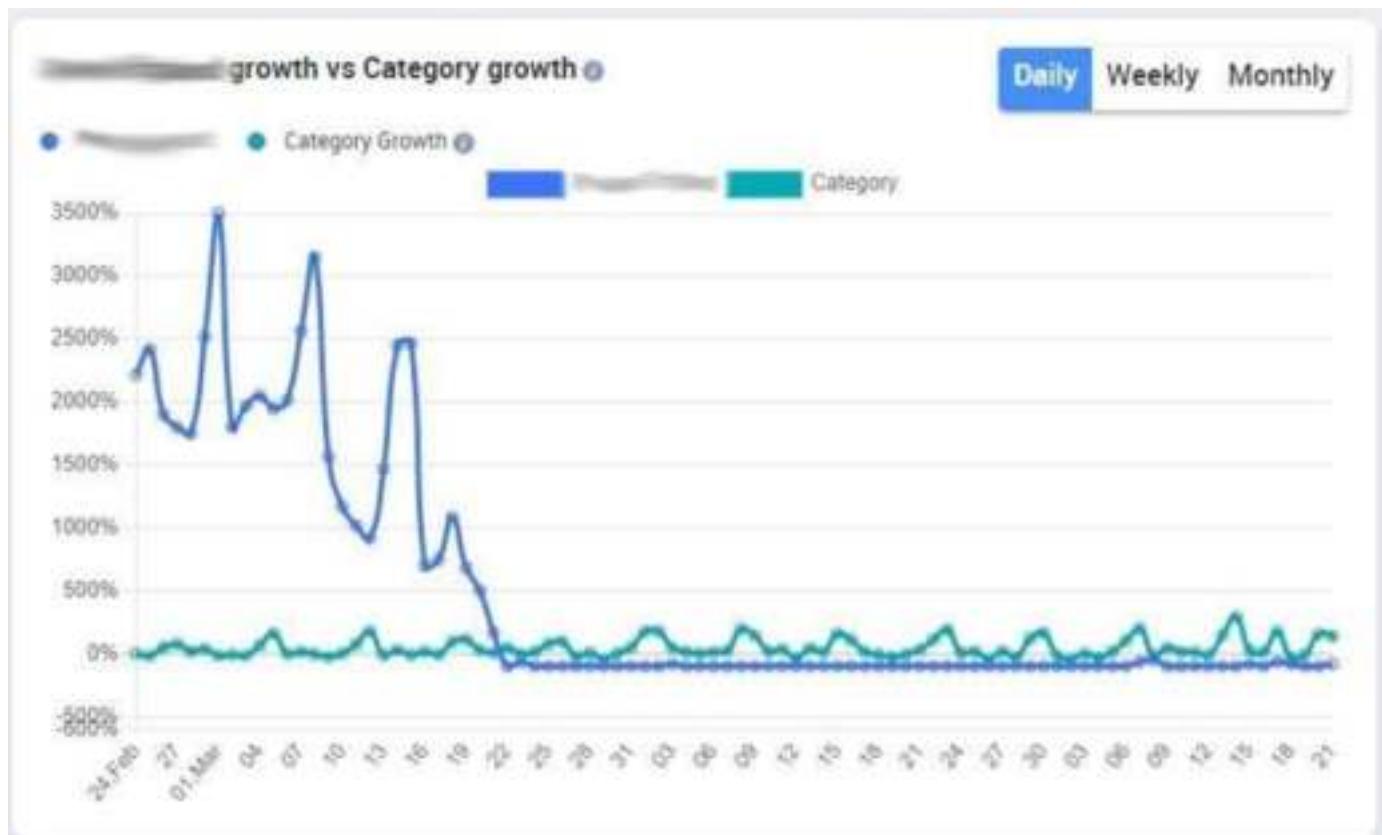


Fig 4.2.2 – Growth Chart

Updated the design of the Goal Tracker. The goal tracker represents the current campaign details, the mission and how its progressing over the period of time.

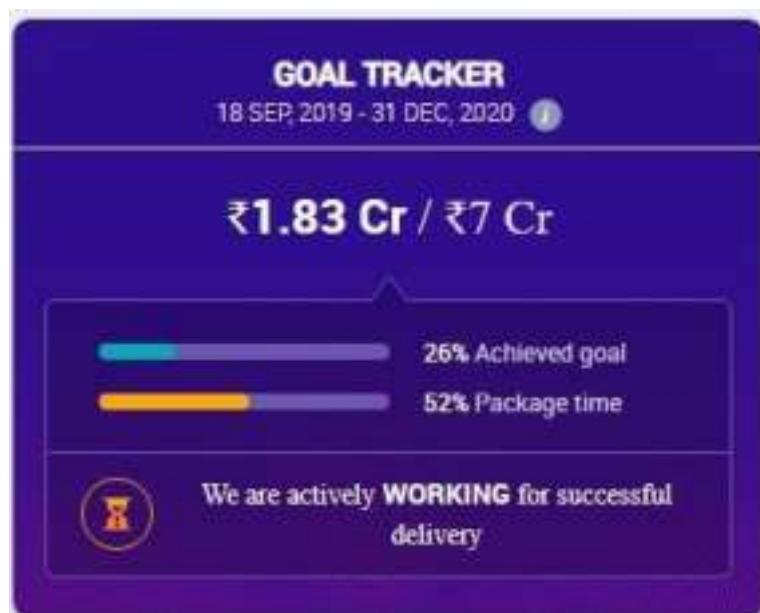


Fig 4.2.3 – Goal Tracker

Updated the design of Sales Chart to use react chart-js library. This chart displays the sales and orders data for the merchant in the current campaign. It is a stacked bar chart.

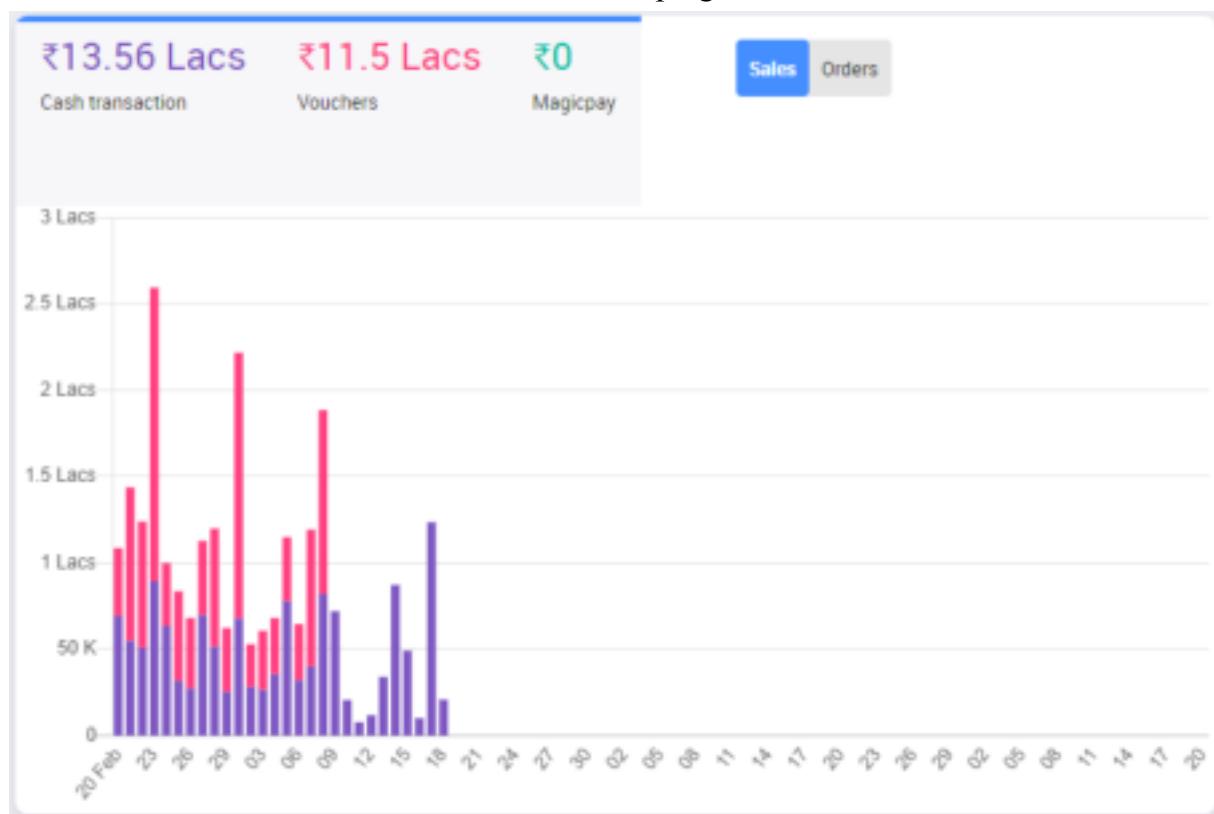


Fig 4.2.4 - Sales Chart

The New Customer Distribution and the losing customers graph are pie-charts. The new customer distribution graphs represents the distribution of customers along the various competitor brands in recent times. And the losing customer distribution shows how the customers of the current brands are shifting towards the other competitors.



Fig 4.2.5 – Gaining and Losing Customers

Created the design and wired the data for repeat-frequency graph. This graph shows the percentage of customers who have transacted more than 2, 3, 4, 5 times.



Fig 4.2.6 – Repeat Frequency Chart

Created the design customer preferences chart, with horizontal slider using the react slick library. This graph shows customer preferences locality-wise and city-wise.

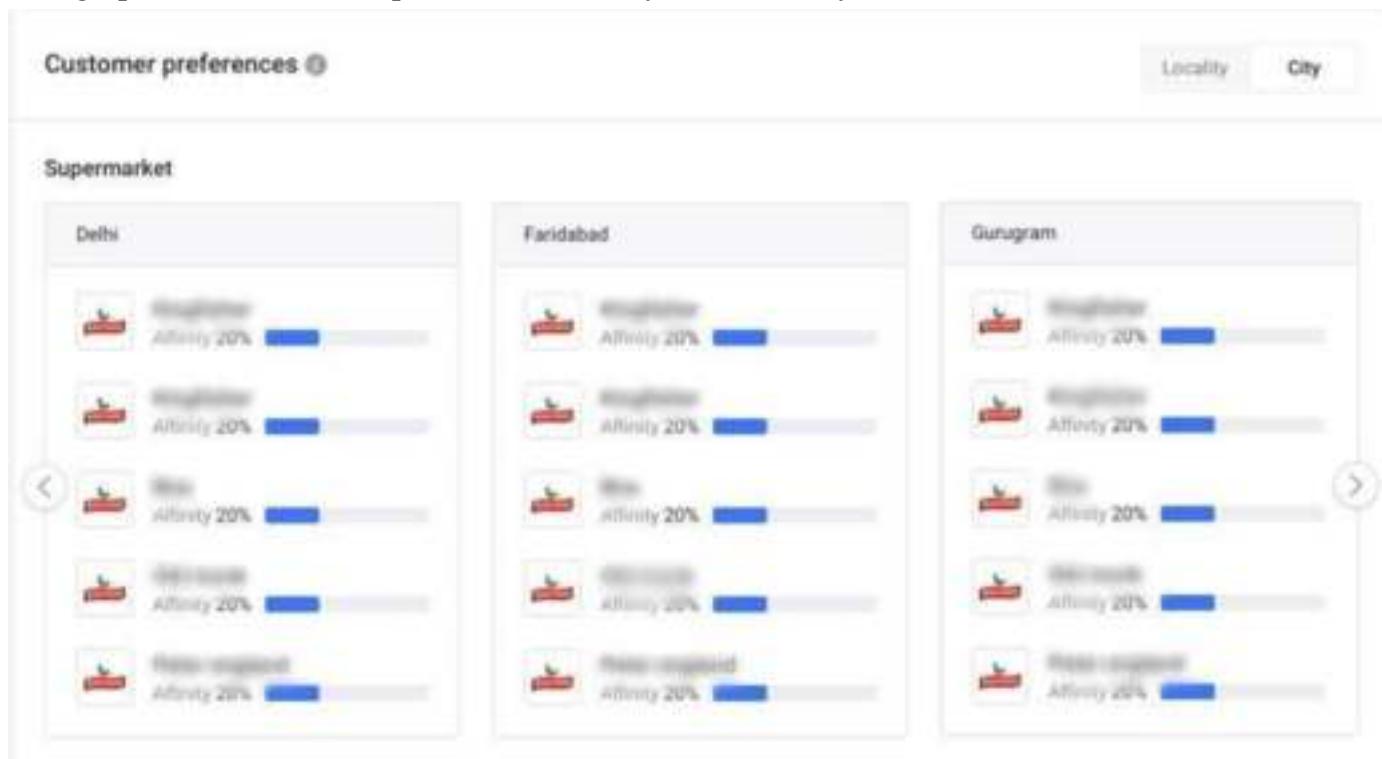


Fig 4.2.7 – Customer Preference

Developed the component for customer profile chart and wired its data. This component displays the distribution of customers for the merchant across genders and the average age of the customers. Same data is shown for the competitors.



Fig 4.2.8 – Customer Profile

Updated the design of Time/Week Chart to use react chart-js. This graph shows how the transactions of the customer varies according to various different times in the day. There is also another option to check how data varies according to various days in the week.

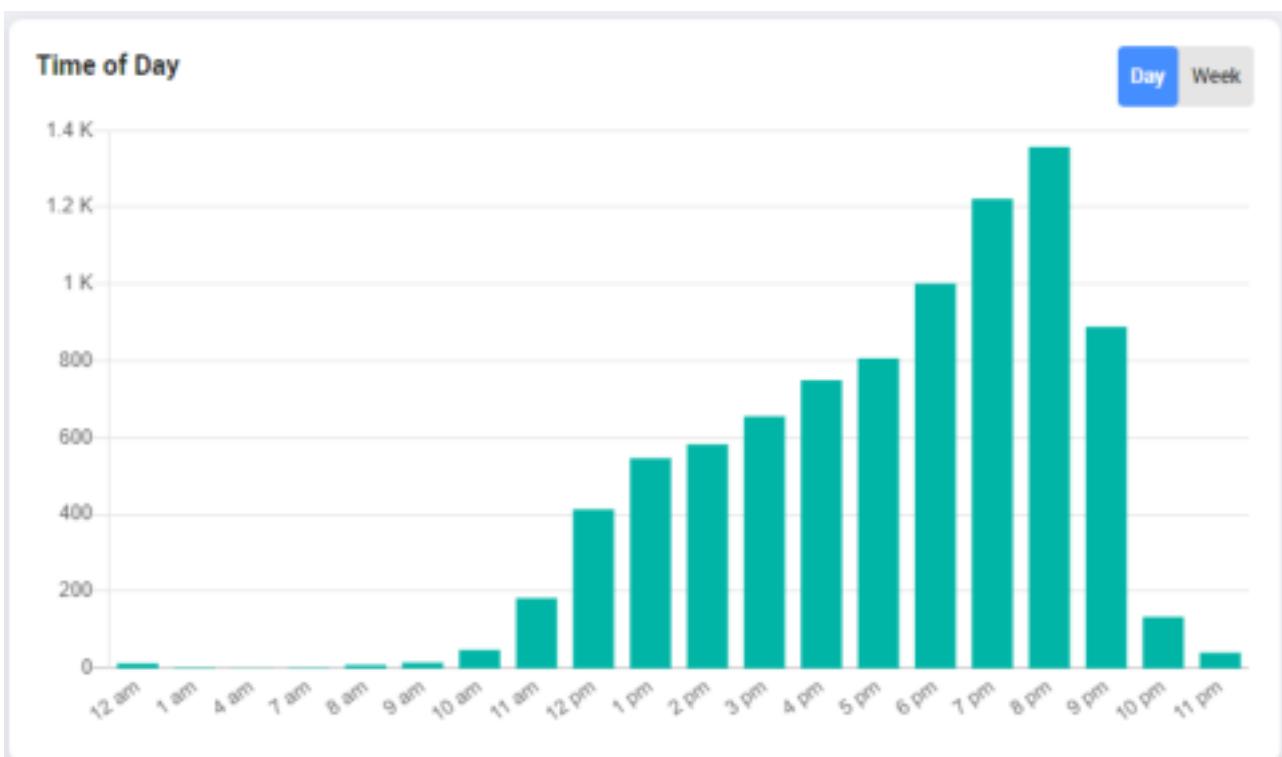


Fig 4.2.9 – Time / Week Pattern Graph

Created the design of Average order value Chart and wired the data for it. This chart shows what is the average value of the customers for the current merchant and its various competitors on the magicpin platform.



Fig 4.2.10 – Average Order Value Chart

Created Potential Chart which calculates potential audience for a merchant. This is a form which allows a merchant to fill details such as city, age, gender, category and the average amount of spend. Based on this data, the target audience for the merchant.

We've got the right customers for you. Tell us about your target customers

Where do your target customers live?

Select City

How old is your target group?

15-25    25-40    >40

Select gender

Male    Female    Both

In which category does your target segment shop?

Select Category

How much do the customers spend per transaction?

Select Spend

Submit

Fig 4.2.11 – Potential Chart Form

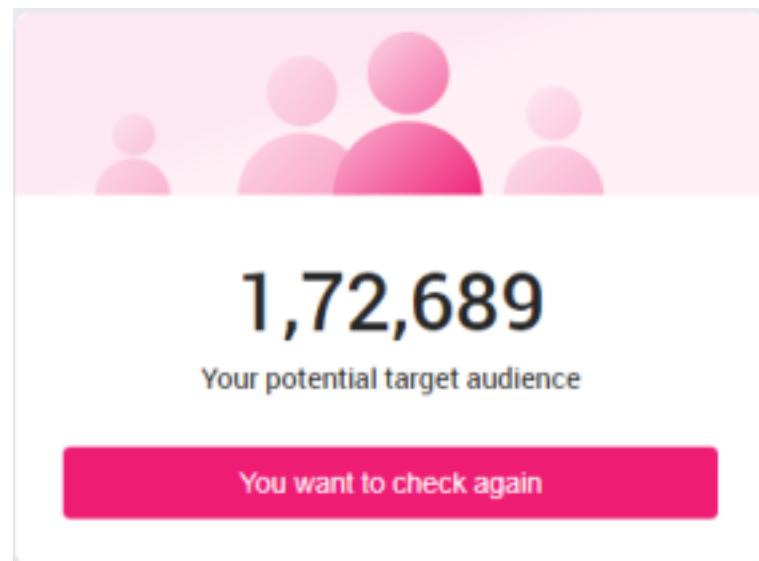


Fig 4.2.12 – Potential Chart Outcome

Wired the feed data on vouchers page for retail-panel. This feed represents the offers provided by the merchants throughout their campaigns on the magicpin platform.



Fig 4.2.13 – Feed on Vouchers Page

**3. Online – Panel:** This panel lists the data of the online merchants. It has various components such as Sales graph, conversion funnel, customer preferences, voucher details, goal tracker, customer profile and city distribution and global filters.

Created the design and wired the data for the sales chart. This chart represents the sales and orders values for the current merchant. Both of these data varies on daily, weekly and monthly basis. In the value tab, average order values is displayed as a line graph with respect to the secondary y-axis and the voucher data is displayed as a bar chart with respect to the primary y-axis. The order tab represents the orders on a bar chart.



Fig 4.3.1 – Sales Chart with values tab selected



Fig 4.3.2 – Sales Chart with orders tab selected.

Created the conversion funnel component and the voucher - details table. The voucher-details table shows the voucher variant displaying the information about the voucher, total number of vouchers sold, total number of vouchers redeemed and the amount of user spend. The conversion funnel graphs shows how many times the merchants portal has been viewed, how many views have been converted into clicks and how many clicks have changed into transactions.

## Vouchers detail

| Voucher variant                                | Voucher Sold | Voucher Redeemed | User Spend       |
|--|--------------|------------------|------------------|
| Get Rs.500 off on minimum purchase of Rs.999   | 383          | 155              | ₹1,65,395        |
| Get Rs.700 off on minimum purchase of Rs.1399  | 438          | 140              | ₹2,34,660        |
| Flat Rs 1000 off on Leaf Pods Wireless Earbuds | 725          | 19               | ₹75,981          |
| Flat Rs 250 Off                                | 76           | 10               | ₹6,292           |
| <b>Total</b>                                   | <b>1622</b>  | <b>324</b>       | <b>₹4,82,328</b> |

## Conversion Funnel

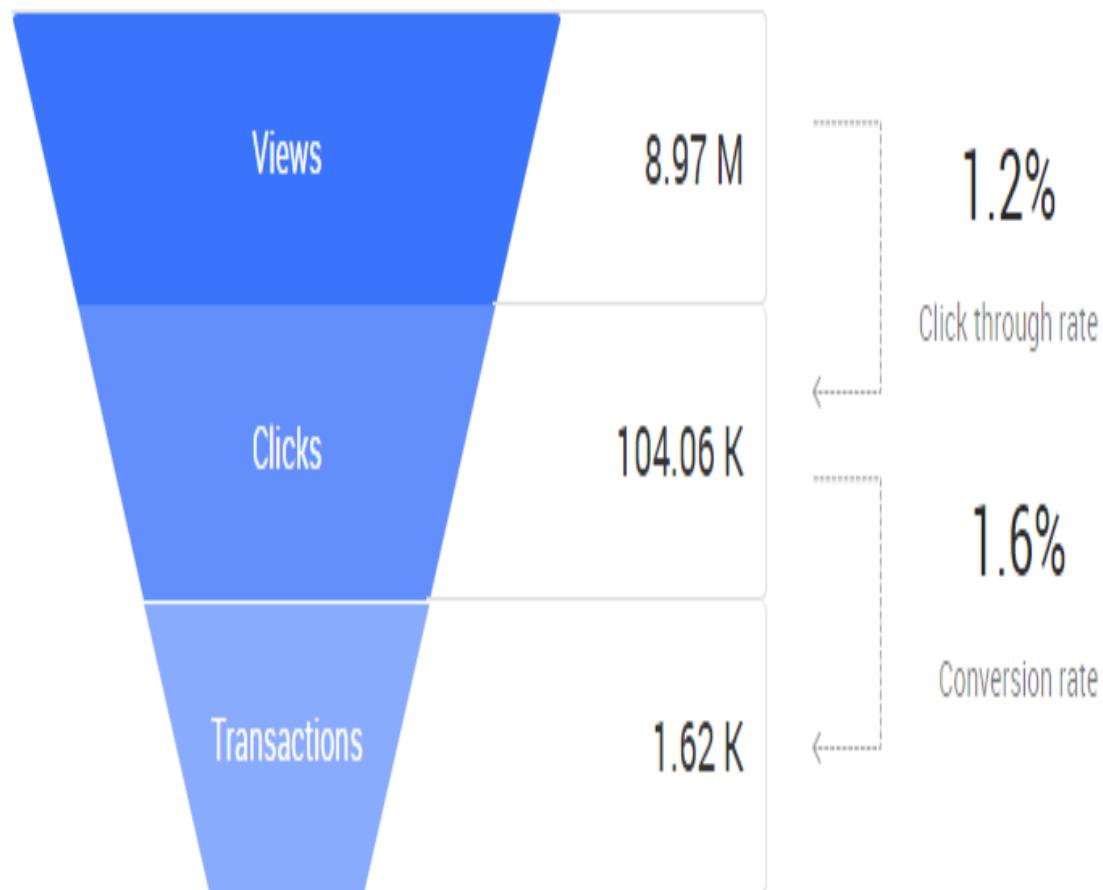


Fig 4.3.3– Vouchers Details Table and Conversion Funnel

Developed the Customer Preference Chart and wired its data. This chart represents the customer preference in various categories such as Food and Beverages, Fashion , Grocery etc.

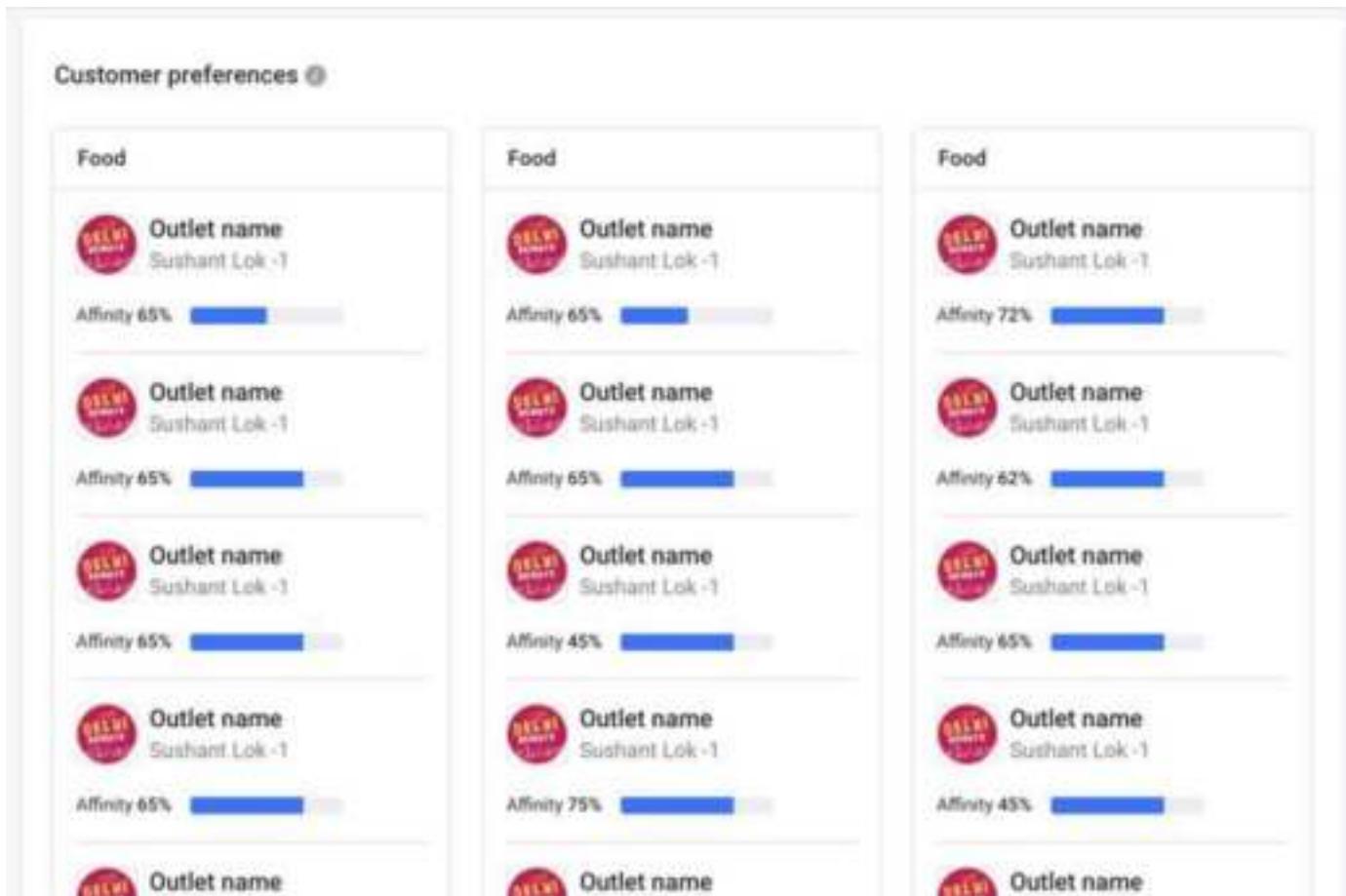


Fig 4.3.4 – Customer Preference Chart

Developed the Customer Profile and City Distribution component. The customer profile represents the percentage distribution of customer gender-wise and in various age groups. The city distribution chart is a pie-chart which shows the distribution of orders for the merchant in various cities.

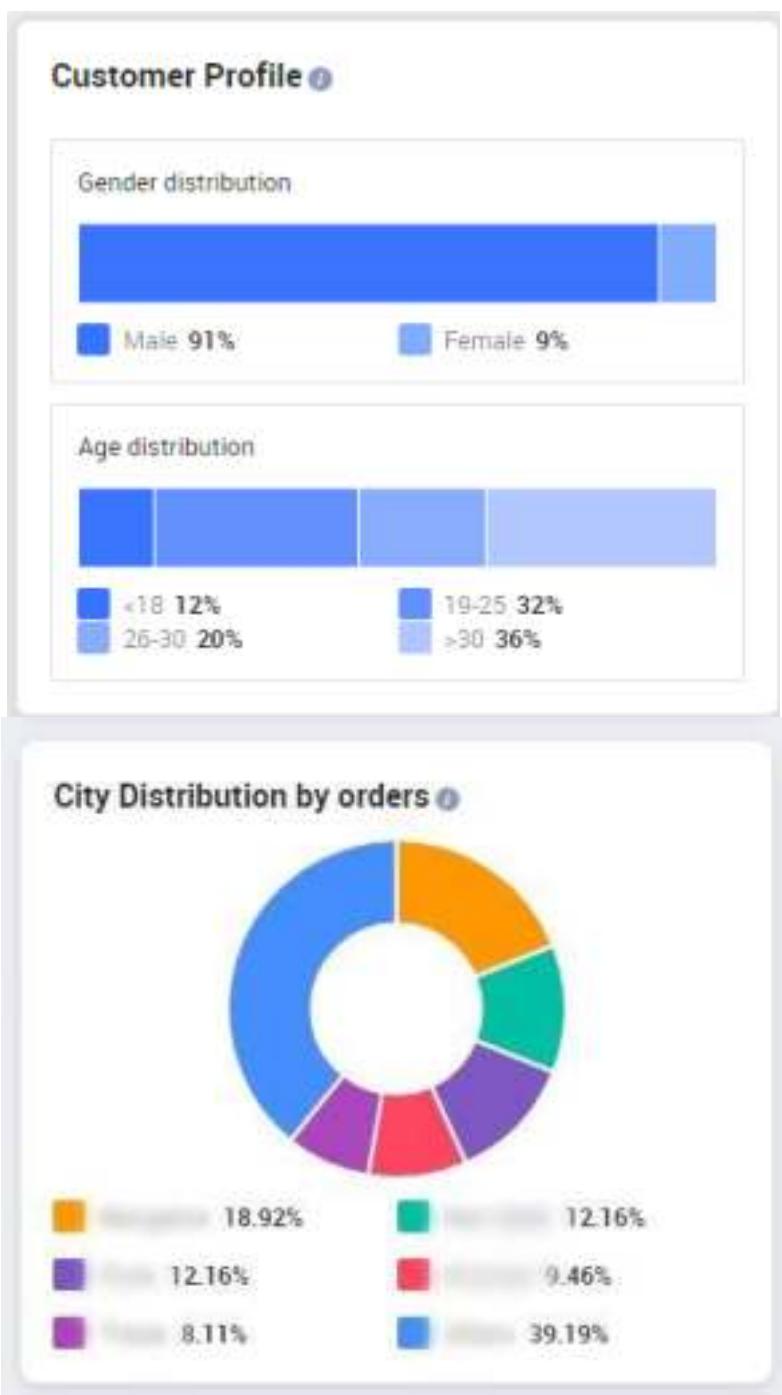


Fig 4.3.5 – Customer Profile and City Distribution Chart

## **BACK-END**

- Written restful API's (Application Programming Interface) for the city distribution chart and customer profile for the online panel and for the potential chart on the retail-panel.
- Performed caching of data using various data structures, so that database isn't referenced on each API call.

## Conclusion

The Upskill Campus Full Stack Development Internship has been an enriching and transformative learning journey. Throughout this program, I gained practical experience in both frontend and backend development, which strengthened my overall technical foundation.

On the frontend, I worked with HTML, CSS, JavaScript, and React, learning how user interfaces are designed and implemented. Understanding state, components, and event handling helped me build interactive and responsive web pages.

On the backend, I developed knowledge of Node.js, Express, API development, and database connectivity. I learned how data flows through applications, how servers handle requests, and how real-world systems maintain security and efficiency.

# **Bibliography**

1. MDN Web Docs – <https://developer.mozilla.org>
2. W3Schools – <https://www.w3schools.com>
3. GeeksforGeeks – <https://www.geeksforgeeks.org>
4. React Official Documentation – <https://react.dev>
5. Node.js Documentation – <https://nodejs.org>
6. Express.js Guide – <https://expressjs.com>
7. MongoDB Manual – <https://www.mongodb.com/docs>
8. Stack Overflow – <https://stackoverflow.com>
9. Python Official Docs – <https://docs.python.org>
10. Upskill Campus Learning Material